COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Covid19DataVault

INITIAL WORK 2020-04-14

Created local on-premise SQL Server 2019 data vault, named it:

Covid19DataVault

Started gathering staging data from sources – only using REST-type WEB data that can be access raw JSON, XML and CSV using a Powershell call.

Current Sources

https://covidtracking.com/api/states/daily

https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties.csv

https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-states.csv

https://covid.ourworldindata.org/data/ecdc/full_data.csv

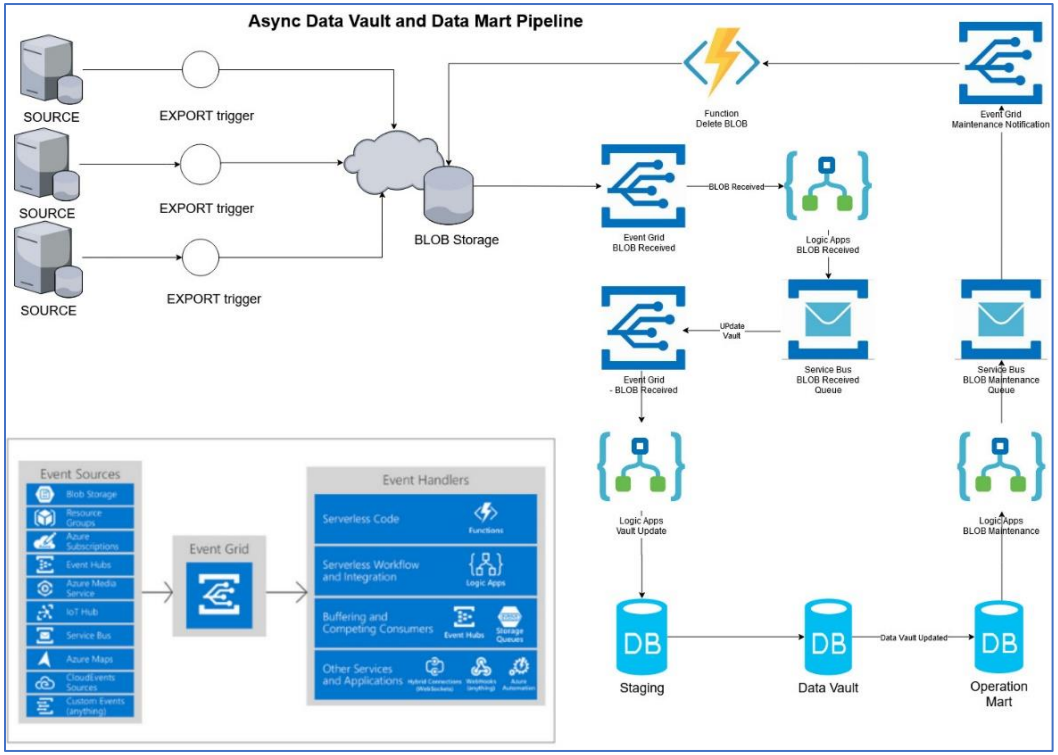
Also loaded **reference** data for World and US State population and population density data. Working on income per capita per state data.

Built Powershell scripts to load JSON, XML and CSV data. Currently running local version that loads daily country and US State and county totals.

Created github site for storing dataVault scripts and documents see https://github.com/dkeeshin/covid19datavault

Modeled an Azure pipeline as an eventual home for process/data vault

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14



Started modelling Data Vault. Right now, it looks like the data vault is something like this

Core Tables

Continent
Region (i.e.Middle East, Oceania)
Country
ProvinceState
County
City

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Detail Tables

CountryDetailPerSource
- MonitorDate
- CountryName
- NewCase
- Deaths

ProvinceStateDetailPerSource01
- MonitorDate
- StateName
- NewCase
- Deaths
- Positive
- Negative

CountyDetailPerSource
- MonitorDate
- CountyName
- Positive
- Negative
- NewCase
- Deaths

Reference Tables

- StatePopulation, Density and Income Per Capita
- CountryPopulation, Density and Income Per Capita
- StatePopulation Density and Income Per Capita
- CountyPopulationDensity, Density and Income Per Capita

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Reviewed a good SQL Server Graph Database video

https://www.youtube.com/watch?v=_KEmK_4ZBYQ

UPDATE 2020-04-19

Scheduled via Powershell **16ScheduleDailyStageLoad.ps1** script Windows Task Scheduler to run Stage table loads for Country and US States and County data from my on premise SQL Server 2019 server. Runs nightly at 7:30pm. Loads all collected data to date – could be optimized to most recent days' worth of data.

Created Powershell US Census county data load script – breaks counties down by US Census Bureau Region and Sub-Region definitions. Loads Stage.USCensusCountyPopulation table in SQL Server.

All Stage tables now export data to JSON files using stored procedures that contain the "JSON" at the end of each stored procedure name i.e. **Stage.upUSCensusCountyPopulationJSON.** The idea is to save JSON files from the Staging tables to BLOB storage in Azure to trigger EVENT GRID and manage flow thru SERVICE BUS to DataVault in Azure.

Generated Core data vault tables locally using Powershell script **18GenerateCoreDataVaultTable.ps1** and CSV definitions table **CoreTableCreate.csv.** Script generates a CoreTableCreate.sql script.

```
DatabaseName,TableName
Covid19DataVault, WorldRegion
Covid19DataVault, WorldSubRegion
Covid19DataVault, Country
Covid19DataVault, CountryRegion
Covid19DataVault, CountrySubRegion
Covid19DataVault, ProvinceState
Covid19DataVault, CountyRegion
Covid19DataVault, County
Covid19DataVault, City
```

Generated a WorldRegion table from https://sciencetrends.com/the-geographic-regions-of-the-world/ for grouping country data by WorldRegion – SubRegion – Country. Cleaned-up a few spelling and naming issues.

Created a MESSAGE schema in the Covid19DataVault. Built and got working three stored procedures for receiving a JSON message and inserting data into DataVault CORE tables via Azure pipeline.

Message.upInsertWorldRegionJSON

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Message.upInsertWorldSubRegionJSON
Message.upInsertCountryJSON

Need to optimize stored procedures with transaction management. Also need to add simple error tracking and sending "received" message and status back to pipeline.

Next, finish JSON message processing to update DataVault.

DEPLOYMENT IDEA--when ready empty staging tables, migrate local database to Azure via BACPAC. Schedule daily updates to data vault via on-premise to Azure pipeline.

UPDATE 2020-04-25

Got stored procedures for updating data vault ITEM tables working
Item.CounTYDetail
Item.CountryDetail

Link.CounTYDetailToCountyCore in place and updating from Message.upItemCounTYDetail

Created Reference schema added table Reference.WorldRegionCountryCode to lookup WorldRegion, WorldSubRegion from CountryCode in Item.CountryDetail Table

UPDATE 2020-04-28

Created **99CSVToSQLForeignKey.ps1**. Returns a list of foreign key candidates from existing data vault tables that are in Core, Item and Link schemas.

UPDATE 2020-04-29

Fixed left outer join #add to Item.CountryDetail find missing dates from #add

Ran backfill scripts to load item and link tables thru yesterday 04-28
23, 27 and 30

Testing completed. 2020-04-29

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Now running local version of Staging to DataVault load. Scheduled via Powershell/Windows Task scheduler. See **16ScheduleDailyStageLoad.ps1**

UPDATE 2020-04-30

Removed redundant key from LINK tables that reference the primary key in ITEM table by DetailID like - **done**

PrimaryStateID in LINK.ProavinceStateDetail to ProvinceStateCore -- **fixed 2020-04-30**

- Insert data to temp tables
- DROP FKs
- DROP Tables
- ReCREATE TABLES
- Insert temp data back into new tables
- RECreate Fks
- Fix procs

Views for reporting - **done**
Created 3 views

- Report.vwCountryDetail
- Report.vwCounTYDetail
- Report.vwProvinceState

UPDATE 2020-05-01

Fixed Reference,WorldRegion changed 'European' to 'European Union' **- done**

Moving Average on Item.CounTYDetail not showing

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Reworked Staging scripts to create separate JSON files per state, county and country otherwise JSON message would exceed Azure Storage QUEUE (64KB) or Service BUS Queue message size (256KB). See stored procedure **[Message].[upGenerateItemCountryDetailJSON]** Stored procedure calls ***Stage.upDailyChangeByCountryJSON*** see **StageupDailyChangeByCountryJSON.sql** . Notice the select statement at the end of the file with the *FOR JSON AUTO*. This is how I prep the data to be sent as a JSON message to the Azure Service Bus Queue.

UPDATE 2020-05-02

Fixed country names that were not matching between Source and world population tables. Need to fix region and sub region data.

```
UPDATE Stage.OurWorldDataDailyCountry
       SET
           Source = 'https://covid.ourworldindata.org/data/ecdc/full_data.csv'
     WHERE Source IS NULL;
     UPDATE Stage.OurWorldDataDailyCountry
       SET
           LoadTime = GETDATE()
     WHERE LoadTime IS NULL;
     UPDATE Stage.OurWorldDataDailyCountry
       SET
           Hash = HASHBYTES('SHA2_256', CAST([date] AS NVARCHAR(27)) + CAST([location] AS NVARCHAR(64)))
     FROM Stage.OurWorldDataDailyCountry
     WHERE Hash IS NULL;
         --country name fixes 20200417
         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Brunei Darussalam', Source = 'Covid19DataVault'  where Location = 'Brunei'
         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Congo, Rep.', Source = 'Covid19DataVault'  where Location = 'Congo'
         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Congo, Dem. Rep.', Source = 'Covid19DataVault'  where Location = 'Democratic
Republic of Congo'
         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Timor-Leste', Source = 'Covid19DataVault'  where Location = 'Timor'
         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Virgin Islands (U.S.)', Source = 'Covid19DataVault'  where Location = 'United
States Virgin Islands'
         --country name fixes 2020-05-01

         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Russian Federation', Source = 'Covid19DataVault'  where Location = 'Russia'

         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Egypt, Arab Rep.',  Source = 'Covid19DataVault'  where Location = 'Egypt'
         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Iran, Islamic Rep.',  Source = 'Covid19DataVault'  where Location = 'Iran'

         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Kyrgyz Republic',  Source = 'Covid19DataVault'  where Location = 'Kyrgyzstan'

         update [Stage].[OurWorldDataDailyCountry]  set Location = 'Lao PDR', Source = 'Covid19DataVault'  where Location = 'Laos'
         update [Stage].[OurWorldDataDailyCountry]  set Location = 'North Macedonia',Source = 'Covid19DataVault'  where Location = 'Macedonia'
```

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

```
update [Stage].[OurWorldDataDailyCountry]  set Location = 'Slovak Republic', Source = 'Covid19DataVault'  where Location = 'Slovakia'

update [Stage].[OurWorldDataDailyCountry]  set Location = 'Korea, Rep.', Source = 'Covid19DataVault'  where Location = 'South Korea'
update [Stage].[OurWorldDataDailyCountry]  set Location = 'Syrian Arab Republic', Source = 'Covid19DataVault'  where Location = 'Syria'
update [Stage].[OurWorldDataDailyCountry]  set Location = 'Venezuela, RB' , Source = 'Covid19DataVault'  where Location = 'Venezuela'
```

Similarly, needed to match US County name data. Used US Census county names;

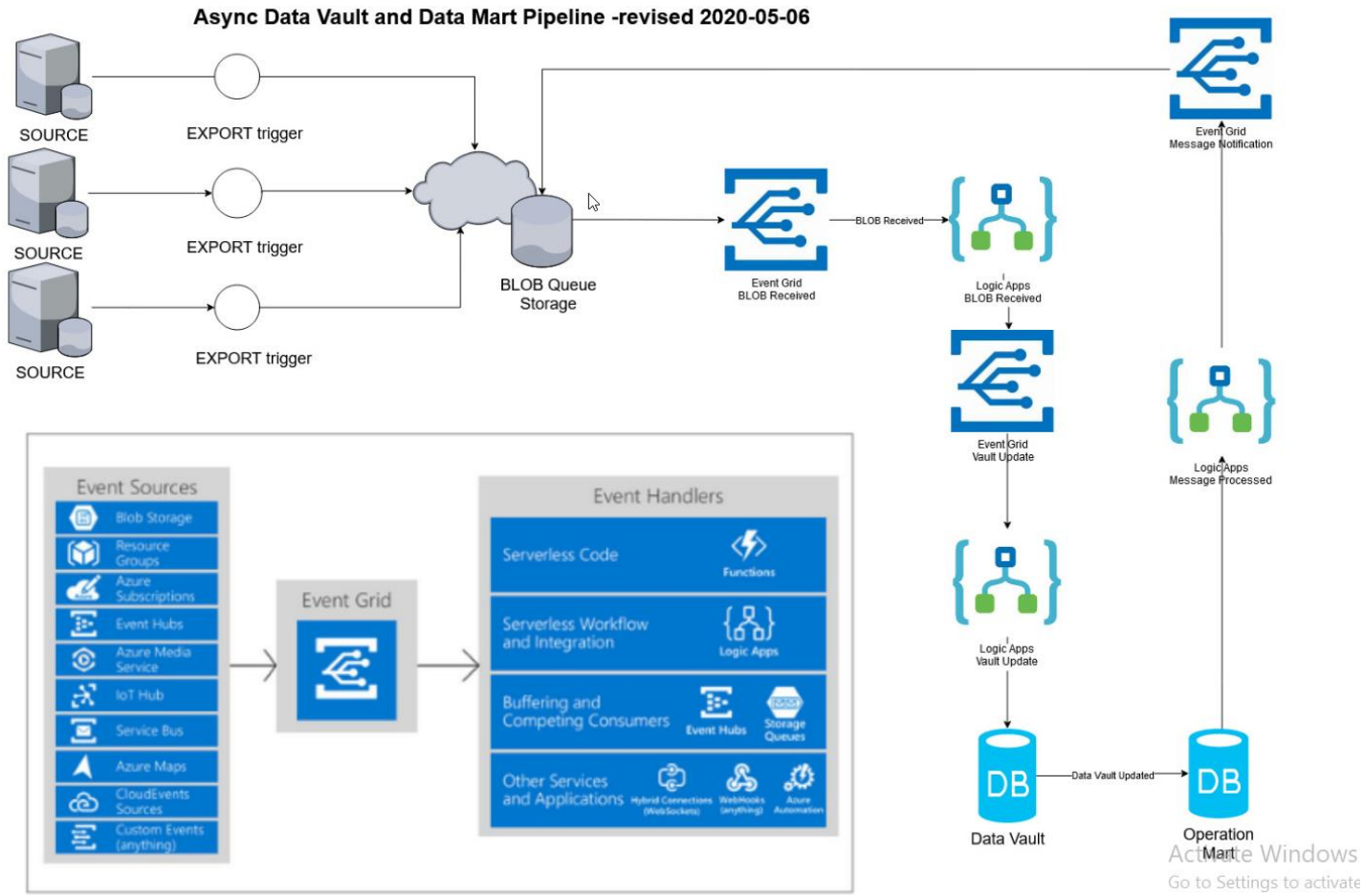
```
--NEED TO RUN the following after initial load. Matches US Census county name to NYTimes Source,
--that drops 'County' and 'Parish' from the county name
IF OBJECT_ID('tempdb.dbo.#1') IS NOT NULL
    DROP TABLE #1;
SELECT CTYNAME,
       STNAME
INTO #1
FROM [Stage].[USCensusCountyPopulation];
ALTER TABLE #1
ADD NYTimesCounty NVARCHAR(64);

UPDATE #1
  SET
      NYTimesCounty = SUBSTRING(CTYNAME, 1, (PATINDEX('%County', CTYNAME) - 1))
WHERE CTYNAME LIKE '%County';
UPDATE #1
  SET
      NYTimesCounty = SUBSTRING(CTYNAME, 1, (PATINDEX('%Parish', CTYNAME) - 1))
WHERE CTYNAME LIKE '%Parish';
UPDATE #1
  SET
      NYTimesCounty = CTYNAME
WHERE NYTimesCounty IS NULL;
UPDATE [Stage].[NYTimesCovid19USCounty]
  SET
      USCensusCounty = CTYNAME
FROM #1 AS a
     INNER JOIN [Stage].[NYTimesCovid19USCounty] AS b ON a.NYTimesCounty = b.county
         AND a.STNAME = b.[state] ;
UPDATE [Stage].[USCensusCountyPopulation]
  SET
      PopulationYear = 2019;
```

UPDATE 2020-05-06

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Revised stored procedures to generate one json file per country row per day. Decided to use BLOB – Queue storage vs. Service Bus. Queue storage max message type is 64KB -- prior to this I was planning on sending all countries (~200) per day as a JSON file. Its' around 100KB. Could of used Service Bus to send entire file since it has a 256KB limit. But decided to use simpler Blob Queue storage. Would have had a similar problem with CounTY data – one days worth of json data equals 1.5MB—so would needed to break that up as well even for Service Bus. Revised ASYNC Pipeline design to reflect this.

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14



"Sharpened" my C# chops. Created C# console utility to read JSON from processed SQL Server staging table and generate json message that will get sent to AZURE Queue Storage. Powershell would have required firing off stored procedure – exporting data to CSV – reading it back – and outputting it to json file. TOO many steps. Will use Powershell to excute C# utility. Anyways, "sharpening" C# skills will help with EVENT TRIGGER and LOGIC APPS coding. I'm using Visual Studio 2019. Started using it to handle SQL Server Management Studio needs as well. Makes

application development easier with database development and C# development IN ONE PLACE. Besides Microsoft removed SQL Server Debugger from SQL Server 2017 Management Studio and put it in Visual Data Tools making it really complicated trying to figure out where to debug. The community version (companies with less than five employees) Visual Studio 2019 therefor seems to be the official landing zone for the SQL Server debugger. Also, I'm using SQL Server 2019 for database development. I just noticed, it looks like I still need to use SQL Server Management Studio in my case 18.x, to generate an entire T-SQL version of a database.

UPDATE 2020-05-07

Got C# code working for generating daily Country, CounTY (notice spelling) and ProvinceState JSON messages. See **GenerateJSONMessage.cs**

Need to check if Staging to Data Vault stored procedures need changes. Should be capable now of receiving individual JSON messages.

Need to configure QUEUE storage via Azure CLI and review EVENT GRID for triggering. Revised design to use Queue Storage instead of Service Bus.

UPDATE 2020-05-08

Checked data vault stored procedures to make sure they are JSON message ready.

```
EXEC Message.upItemCountryDetail @m
EXEC Message.upItemCountyDetail @m
EXEC Message.upItemProvinceStateDetail @m
```

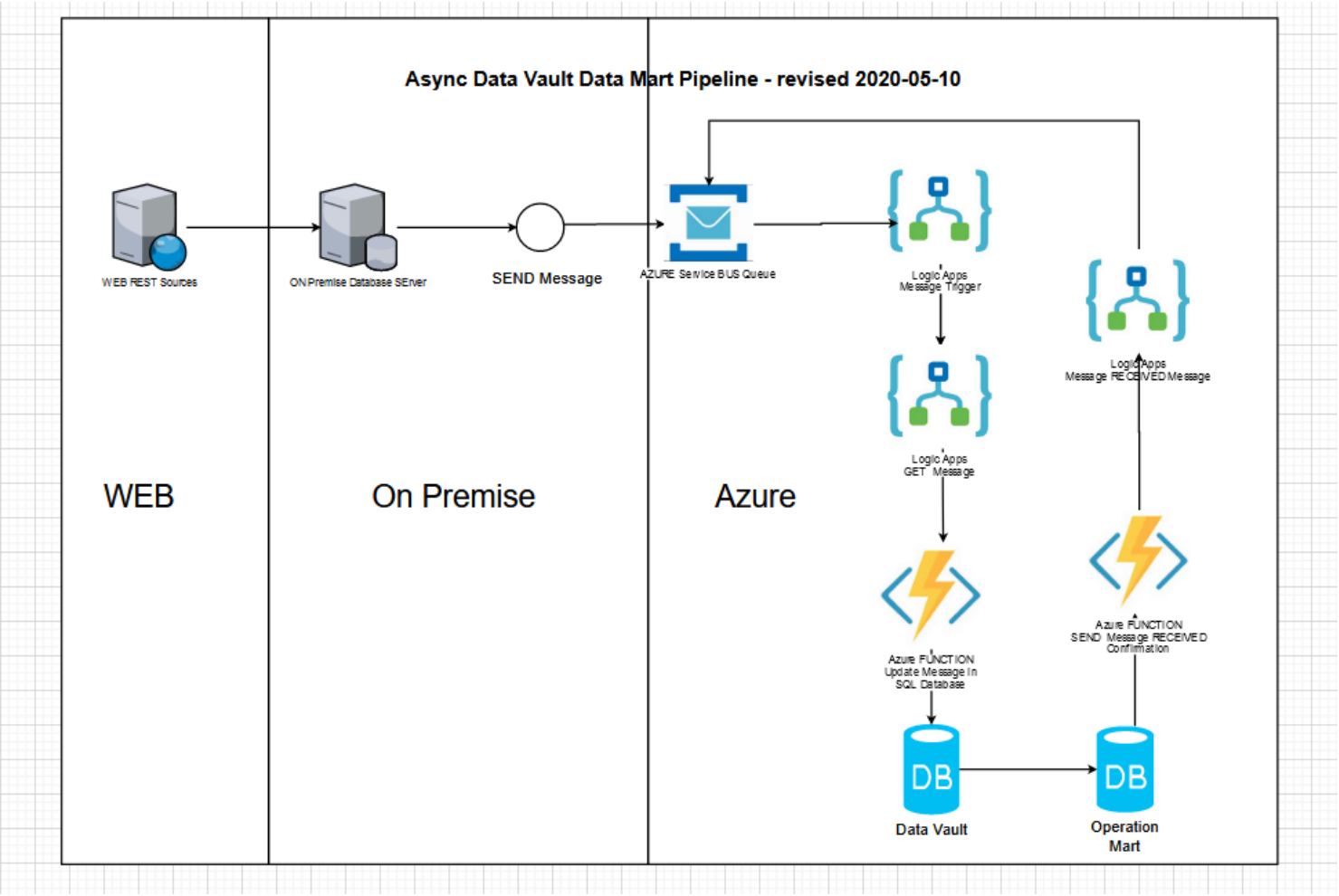
UPDATE 2020-05-10

Fixed data views and loading process [Stage].[upDailyChangeByCountry] to use TOTAL cases Per Density vs New Cases.

Decided to go with Service Bus vs Queue storage. I decided Service Bus is AMQP compliant and has more features that may come in handy.

Therefor, the most practical design right now is:

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14



UPDATE 2020-05-11

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

From the Azure Portal created a ServiceBus resource

Created a queue

Could do this with an Azure Powershell script like this

```
Install-Module Az.ServiceBus   ##installs CLI module
Login-AzAccount
Select-AzSubscription -SubscriptionName " Azure Subscription 1"
Get-AzContext


##basic script for creating resource, namespace, queue and retrieve connection string
# Create a resource group
New-AzResourceGroup –Name my-resourcegroup –Location eastus

# Create a Messaging namespace
New-AzServiceBusNamespace -ResourceGroupName my-resourcegroup -NamespaceName namespace-name -Location
eastus

# Create a queue
New-AzServiceBusQueue -ResourceGroupName my-resourcegroup -NamespaceName namespace-name -Name queue-name -
EnablePartitioning $False

# Get primary connection string (required in next step)
Get-AzServiceBusKey -ResourceGroupName my-resourcegroup -Namespace namespace-name -Name
RootManageSharedAccessKey
```

Installed **Service Bus Explorer**

Next, send a message.

UPDATE 2020-05-11

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Got Azure service bus set up. C# code **GenerateJSONMessage.cs** to SEND message to QUEUE directly from on-premise SQL Server table containing JSON message(s). Got it working.

UPDATE 2020-05-12

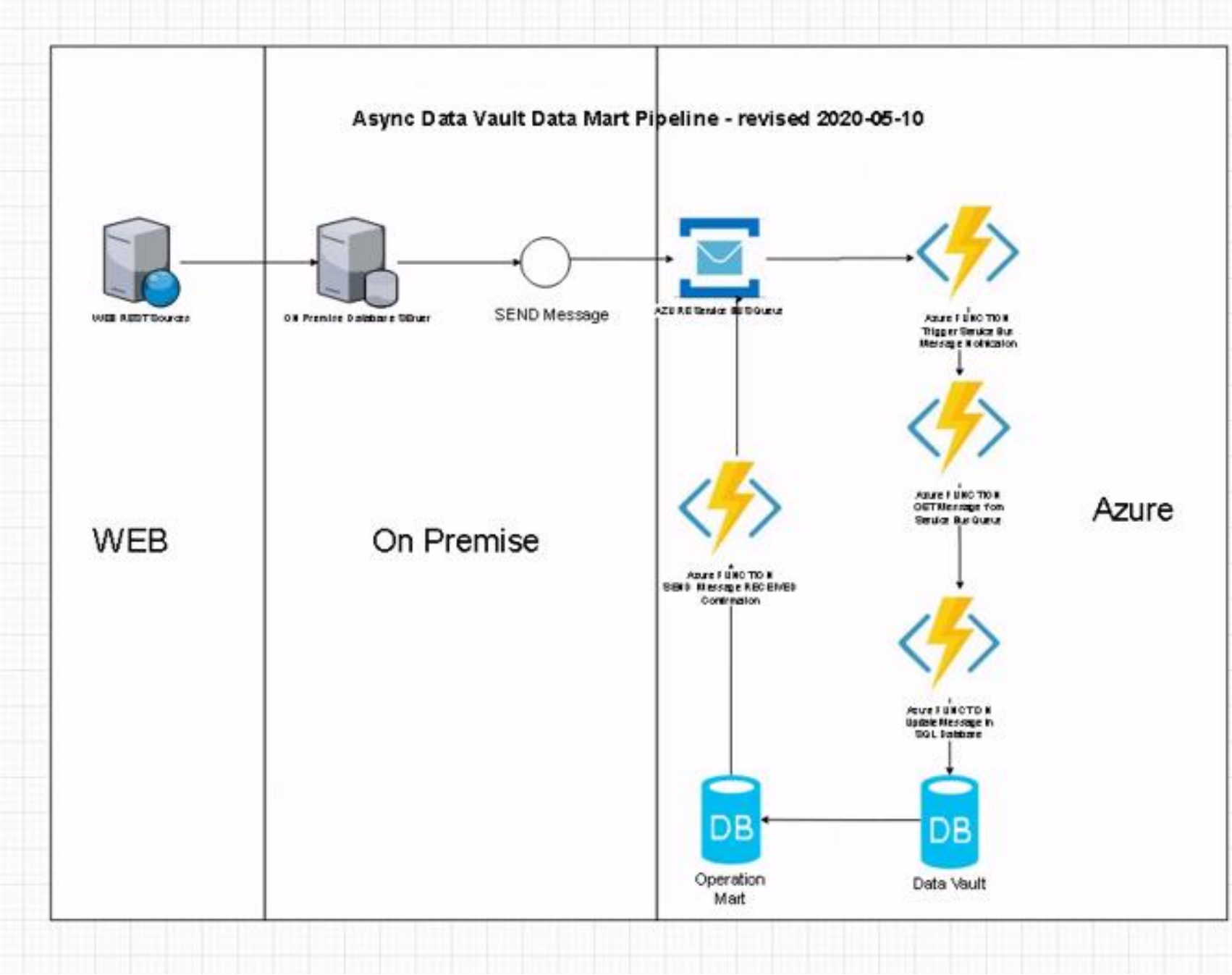
Investigating using Azure logic Apps vs Azure function to trigger. Both have triggers. Event Grid does too. Event Grid was ruled out temporarily since it requires Premium Service Bus subscription – potentially making it prohibitively expensive. And not necessary since the volume of messages is low.

Logic Apps are "set it and forget" cloud only components with lots of connectors to micro services. When not it in use you would want to disable them, otherwise they are charging your subscription. I realized they look a lot like BizTalk and noticed:

https://azure.microsoft.com/en-us/updates/azure-biztalk-services-simplifying-our-azure-offerings/

They are forever polling Service Bus – meaning charges can add up. Looks like functions are the way to go at this point.

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14



COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

UPDATE 2020-05-14

Its' complicated and confusing getting a local version of Azure Functions working. You can see what I'm talking about here:

https://docs.microsoft.com/en-us/azure/azure-functions/functions-run-local?tabs=windows%2Ccsharp%2Cbash#v2

Need some training?

https://docs.microsoft.com/en-us/learn/modules/develop-test-deploy-azure-functions-with-core-tools/1-introduction

I finally, was able to get it installed and working. At times, Microsoft Azures' release early and often efforts make using a newish tool like Azure Functions frustrating. Problem was the format and location of the Service Queue connection string needs to source from the json file. See **local.settings.json.** "Connection = " statement is required. The function that I got working, triggers when a message is sent to a Service Bus Queue. Now, I need to figure out how to get the function to fire off a stored procedure that will load the message.

UPDATE 2020-05-15

Got the local version of Azure Functions working. See **Function3.cs**

Reviewing ways to migrate on premise SQL Server database to Azure SQL Database.

UPDATE 2020-05-16

Decided to restore a current backup of Covid19DataVault to a on-premise database named Covid19DatVaultForAzure. I then scripted removing staging tables and procedures . Finally I generated a BACPAC file for deploying to Azure.

This is the database deployment procedure.

For database schema see **01Covid19DatabaseForAzure.sql**

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Was able to get Azure function working. It will read the data vault queue and fire off a stored procedure to update the data vault in Azure. Got the local version running. See sample C# code:

Function3.cs

UPDATE 2020-05-17

Testing of pipeline process between staging source database and local destination data vault through Azure service bus queue

Investigated options for C# configuration file for connection strings, server names, etc. for **GenerateJSONMessage.cs** Determined that
`ConfigurationManager.AppSettings.Get("DatabaseServer");`
required using System.Configuration. Added that, still didn't work. Kept telling me it was out of context. Finally found a stackoverflow note to NUGet the lastest version. That resolved the issue.

Got Service Bus function to finally trigger a retrieve and fire the stored procedure to insert ProvinceState data in a local version of the data vault. For testing purposes, I need to compile the code the sends the messages to the queue – since I can't start the service bus trigger in debug mode and send messages to the queue in debug mode at the same time. Maybe able to do it in one project file – will test.

Will create a total of three ServiceBus queues. One for each data vault source. Will reduce ServiceBus function trigger complexity.
Will need to figure out a way to stop trigger from polling(?). Maybe disabling or dropping queue after processing is done might be easiest approach.
Daily process will likely be:

Start the ServiceBus triggers
Load on-premise staging data from sources,
Transmit data as messages through Service Bus queue to Azure database
Remove or disable queues

UPDATE 2020-05-18

COVID19DataVault Work Log Notes
Prepared by dkeeshin@keeshinds.com
2020-04-14

Had issues running functions multiple times. Was fine sending one message to the database, after that it stopped. I was surprised to find the function was ok and noticed the stored procedure **[Message].[upItemCounTYDetail]** was not finding missing data after the first row that contained the data for the date I was inserting. Needed a tie breaker, since everything previously was loaded in a batch by date.. Updated stored procedure to include day plus state or day plus country. Simplified state-county load by adding Statename and CountyName to Item.CounTYDetail table.

Updated Message.upCountryDetail, Message.upCounTYDetail and Message.upProvinceStateDetail updated. Added columns to Item.CountyDetail.

Got Function2.cs running and created Function3.cs. Having an issue with Message.upCounTYDetail not updating – message gets read from the queue, just not updating the table.

UPDATE 2020-05-19

Changed stored procedure **[Message].[upItemCounTYDetail]** mentioned yesterday to use **MonitorDate** and **FIPS** number to break the tie for finding new data. The NYtimes data uses the FIPS number to uniquely identify County-State combinations.

Meanwhile I had another issue, with Message.upCounTYDetail table not updating from the servicebus trigger. Discovered the stored procedure [Stage].[upDailyChangeByCounTYJSON] that generates JSON CounTY string for message queue was using a nested select statement. FOR JSON AUTO statement was nesting the JSON string as well. I just wanted one short, straight message about 600KB in length to send to the data vault. Decided to insert nested select query into a memory table table first prior to generating JSON. Worked.

All three functions now functional. Ready to prep for testing deployment to Azure.