**Azure Synapse Analytics (Data Warehouse)**

Over years I worked on Azure data platform and I saw data services grows exponentially. The reasons are new business requirement, limitation, reduce time to market and challenges. There are many data services from storing csv files, tabular data, nosql data, graph data, column store, key-value etc. which helps different business to choose which is best for them. On Azure there are many other services for data orchestration, processing, AI and presenting the data. If dig further on any of the terminology one can find too many options. Sometime it confuses user but i feel you choose what's best for the problem and skill set organization have. Azure provide all and covers different persona in any organization.

Last year, Microsoft bought more capabilities to Azure SQL Data warehouse and named it as Azure synapse analytics. As of today (day I am writing this blog) it's under private preview. It's a limitless analytic services which bring data warehousing and big data analytic capabilities together. It's beyond just seperating storage and compute.

However there are still some concerns from organizations before moving to production.

**Why**

Any organization who is ready to spin Azure service always curious of performance, load, security etc. The story is no different with Data. With Azure Synapse DWH organization want to make sure they choose right skus and design to benefit customer and themselves. Any organization can come up with following questions

1) Are we choosing right sku for current and future workload?

2) Are we utilizing the resources well?

3) What about the latency?

4) Based on upcoming request, what will be the impact on allocated resources?

5) How to make sure quries are performing well on DWH design?

6) How to build baseline matrix?

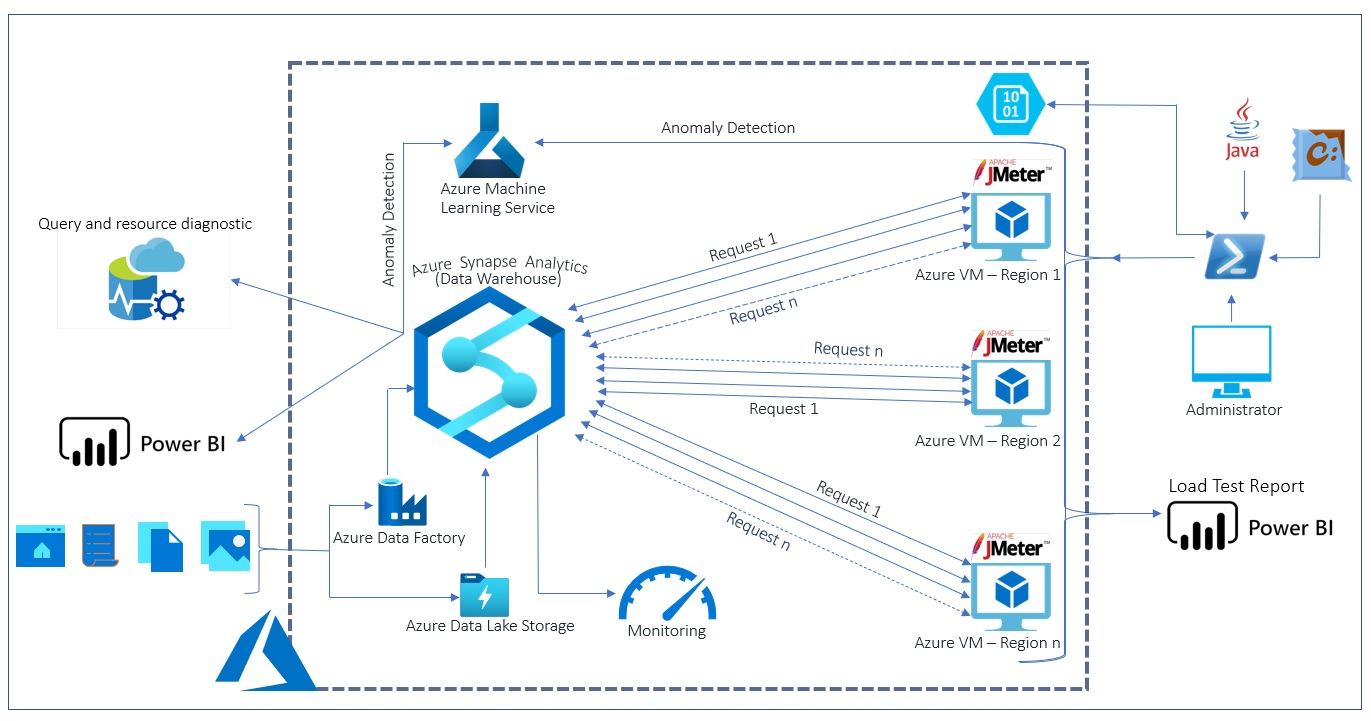
and may be more...

Well, one of the ways is to monitor (and setup alerts) the service on production environment and adjust the sku. However this may impact the user experience and become hard practice to follow.

This blog post try to overcome this problem in advance to avoid any surprises on production environment. The focus will be on building a framework to simulate the load with multiple queries and get answers for above questions.

**Reference Architecture**

In this end to end architecture, we'll simulate the load from different regions with different queries. The outcome will help us understand if we choose right sku and setup good DWH design like table distribution, index, cache etc .

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/RefArch.jpg)

The key elements of the architecture :

**Azure Synapse Analytics (Data Warehouse)** : - One interface to design end to end advance analytics needs for an organization. The platform provides various options to reduce cost (SQL pool, SQL on-demand), low/no learning curve, easier to build, deploy and manage (Synapse studio). Microsoft provided great content on Azure Synapse Analytics [here](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/).

**Azure Virtual Machine** : - Azure virtual machine will be used to generate actors to send request to Azure Synapse Data Warehouse.

**Power BI** : - Power BI is used as reporting tool to provide deep analysis on query execution and load metrics. Look out [here](https://powerbi.microsoft.com/en-us/) for more information about Power BI capabilities.

**Azure Data Studio** : - A cross platform tool to query data warehouse. I like two of it's feature very much 1) source control integration 2) built-in charting capabilities of query result sets. There are more which can be found [here](https://docs.microsoft.com/en-us/sql/azure-data-studio/what-is?view=sql-server-ver15).

**Azure Machine Learning** : - Azure Machine learning make machine learning model developement, deployment, retain, tracking and manage make easier. The service provide a workspace where irrespective of your skill you can build machine learning model. You can build machine learning model using machine learning studio, notebook or auto ml. Check out more [here](https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-ml)

**Azure Data Factory** : - A cloud based data integration service make data movement across different location with ease, fast and securely. You can build data pipeline within no time and connect to more than 90+ sources. Please check [here](https://docs.microsoft.com/en-us/azure/data-factory/introduction) for more information.

**Azure Blob Storage** : - A low cost, high performant storage solution on cloud. Use different tools like azcopy, Azure blob storage explorer, ADF to store unstruture data. Please click [here](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-overview) for more information.

**Apache Jmeter** : - [Apache Jemter](https://jmeter.apache.org/) is a open source solution to perform load test and measure performance. Apache Jemeter is a java based 100% open source technology. Thi tool can help in generating the expected load, perform action and provide various reports to evaluate the performance of the targeted application.

**Chocolatey** : - A software management tool for Windows. Just bundle the software packages in PowerShell and deploy in any Windows environment. It's easy and fast. More more information visit [here](https://chocolatey.org/).

**Java Runtime Environment** : - Java is a programing language to build software. This technology provide libraries to build software and JRE (java runtime engine).Using JRE in this blog because Apache jemter is written on Java.

**Setting up environment**

**Azure Synapse Analytics (Data warehouse)**

Use [Microsoft Azure Portal](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/create-data-warehouse-portal) or [PowerShell](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/create-data-warehouse-powershell) to create Data warehouse. Select appropriate perfromance level (start with minimum if not sure).

**Setup Data warehouse**

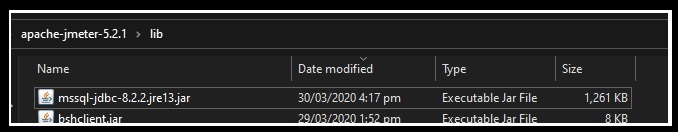
Once Azure Synapse DWH is setup, next step is to setup Data warehouse design and load data. One of the ways to consider is first move metadata like table, SP, index etc creation then leverage [Azure Data Factory](https://docs.microsoft.com/en-us/azure/data-factory/introduction) for data movement. I am using Adventureworks dwh which is provided as a sample while creating Azure Synapse DWH.

**Download and configure Apache Jmeter**

Download Apache Jmeter (version 5.2.1) from [here](http://jmeter.apache.org/download_jmeter.cgi). Unzip the downloaded folder.

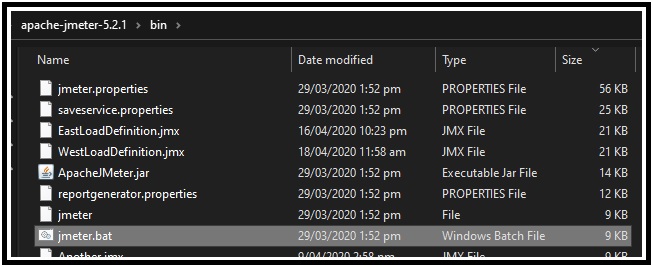
By deafult Apache Jmeter don't include JDBC driver to connect to Azure Synapse Data warehouse. Let's download the JDBC driver.

1. Download [JDBC driver](https://docs.microsoft.com/en-us/sql/connect/jdbc/system-requirements-for-the-jdbc-driver?view=sql-server-ver15)
2. Unzip and copy **smssql-jdbc-8.2.2.jre13.jar** file
3. Paste mssql-jdbc-8.2.2.jre13.jar file under \apache-jmeter-5.2.1\lib\

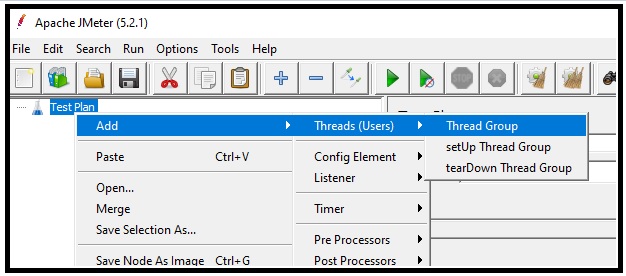
[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/SQLJDBCDriver.jpg)

**Build Java Management Extension (aka jmx) file**

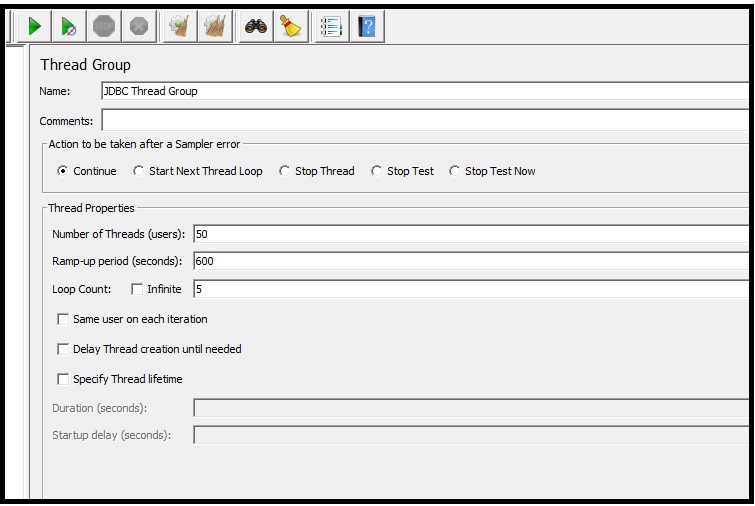
1. Open Apache Jmeter UI by clicking jmeter.bat under \apache-jmeter-5.2.1\bin\

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/JmeterBatch.jpg)

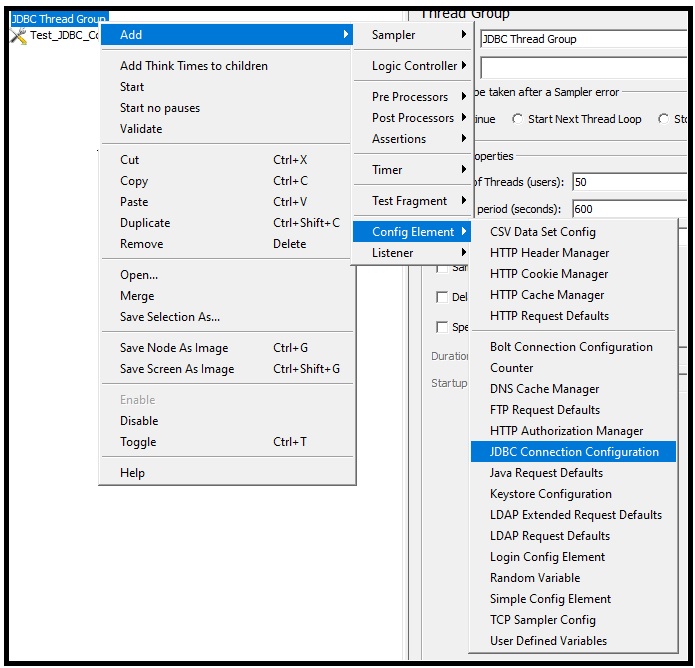
1. Click **Add** -> **Threads (Users)** -> **Thread Group**

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/AMeterAddThread.jpg)

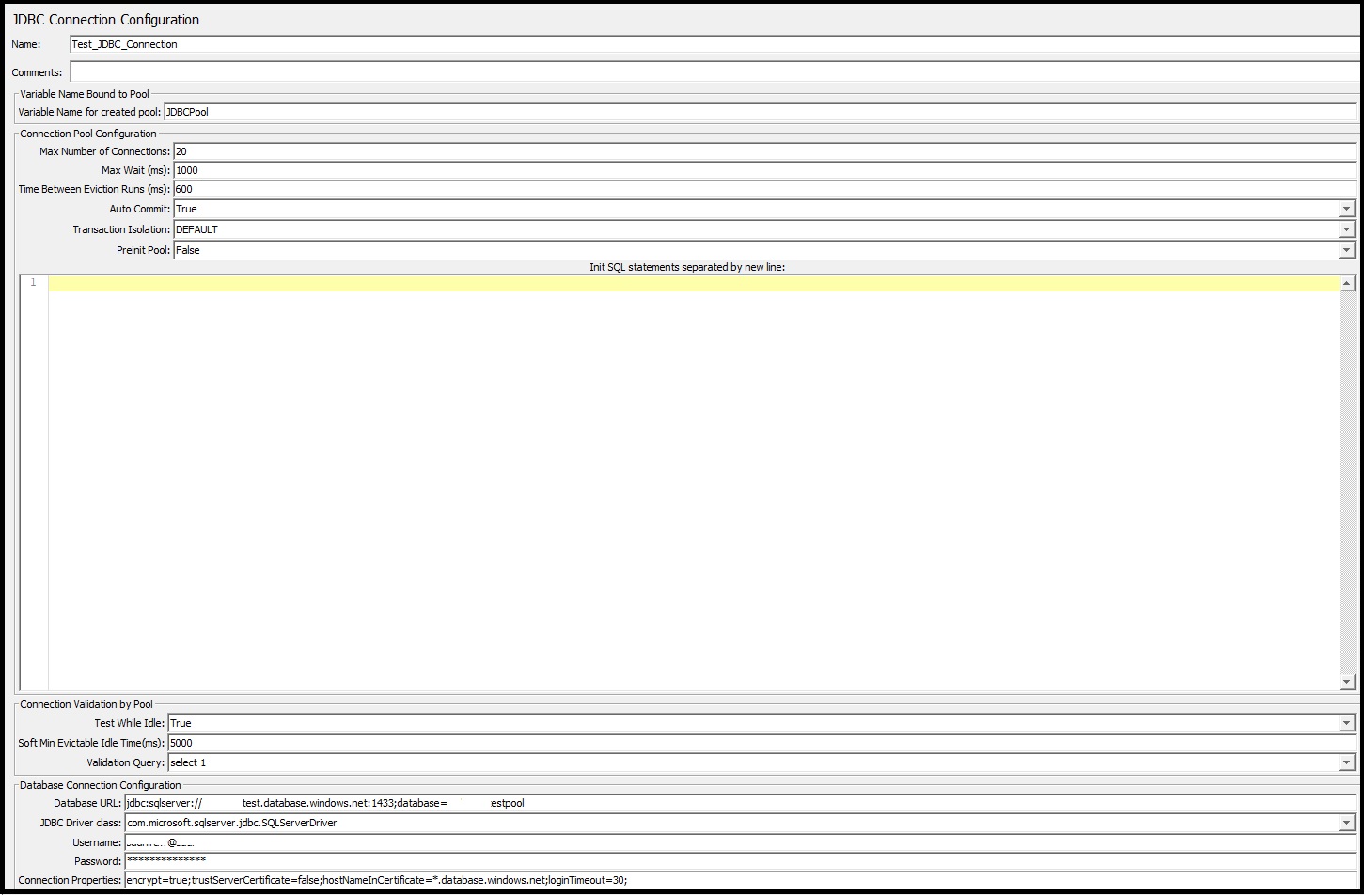
1. Provide **Number of Threads (users)** . Add number of users you are expecting in future. Provide **Ramp-up period (seconds)** menaing how much time Apache jmeter takes to spin up number of threads(user). Also, provide **Loop Count** define number of times test repeat.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/JDBCThreadConfigured.jpg)

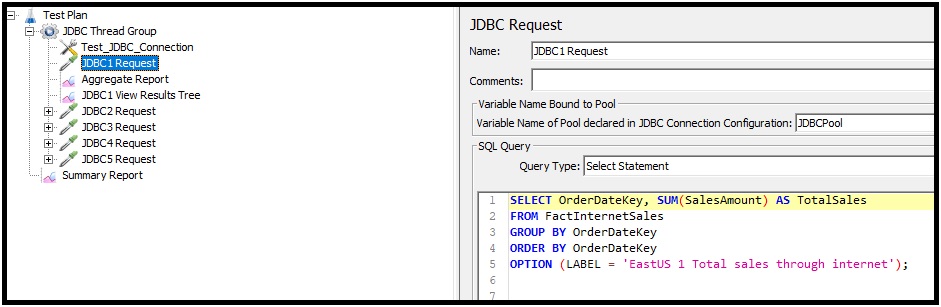
1. Add JDBC connection. Click **Add** -> **Config Element** -> \*\*JDBC Connection Configuration.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/AmeterConnection.jpg)

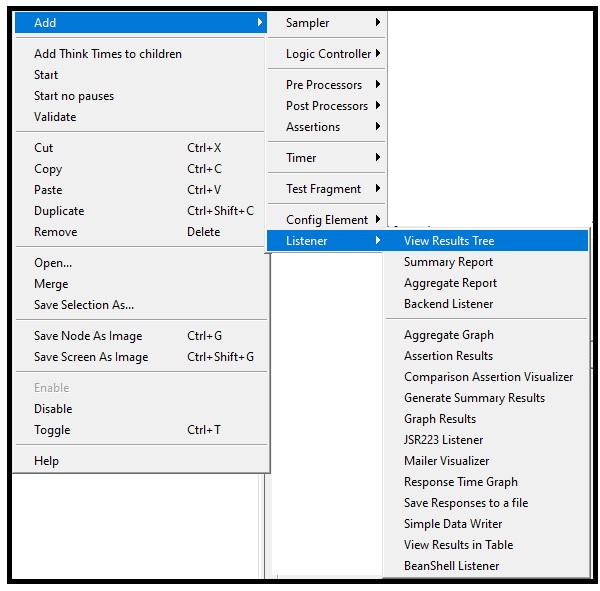
1. Add Azure Synapse Data warehouse connection value. Provide **Variable Name for created pool** which we created in earlier step. Provide Data warehouse connection value in **Database Connection Configuration**

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/JDBCConnection.jpg)

1. Create JDBC request. Righ click on thread group. Click **Add** -> **Sampler** -> **JDBC Request**. Provide **Variable Name of Pool declared...** name created in earlier step. Provide **SQL Query** to be execute against data warehouse.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/JDBCRequest.jpg)

1. Create report for each request execution. Let's add **View Results Tree**

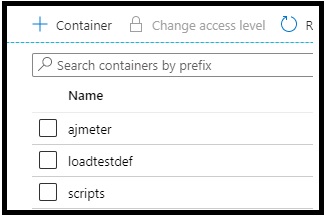
[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/AddResult.jpg)

1. Create multiple JDBC request (with different queries). A sample jmx file is located at \Scripts\SampleLoadDef.jmx. Also sql query are located at \Scripts\SampleQuery.txt

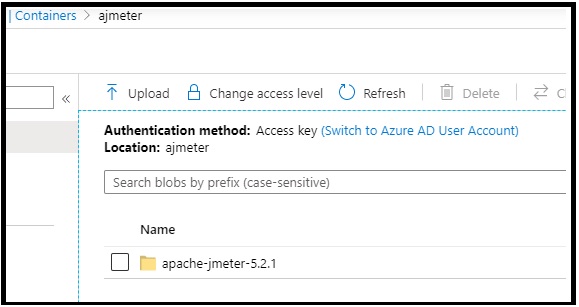
**Move scripts to Azure storage**

Let's move artefacts in Azure blob storage. Do sequentially.

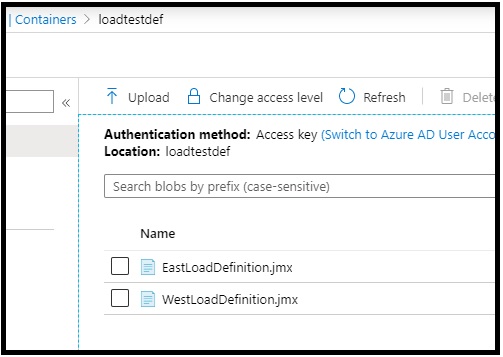
1. Create Azure blob storage and three container under it. **1) AJmeter** to store Apache Jmeter package. **2) loadtestdef** to store script file and **3) scripts**.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/StorageContainer.jpg)

1. AJmeter:- Use [azcopy](https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10) to move Apache Jmeter from local drive to Azure blob storage.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/StorageAJmeter.jpg)

1. Copy jmx files created in earlier steps under **loadtestdef** container.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/JMXStorage.jpg)

**Note**:- Load test scripts which will be executed from East US and West US

1. Upload below PowerShell script in **scripts** container. Below scripts will be executed in order once Azure VM is created.

4.1 **swinstall1.ps1**:- File store under .\Scripts. This script will install chocolatey software management tool in VM. Once chocolatey installed sucecssfully then it will call next script.

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'));

.\swInstall2.ps1

4.2 **swinstall2.ps1**:- File store under .\Scripts. This script will install openjdk. SQL JDBC driver needs Java 14 runtime hence install openjdk. Next is to install azcopy (version 10). Once installtion completed succesfully them it will call next script.

Set-ExecutionPolicy Bypass -Scope Process -Force;

choco feature enable -n=allowGlobalConfirmation;

choco install openjdk -y;

choco install azcopy10 -y;

.\MoveAJmeter.ps1

4.3 **MoveAJmeter.ps1**:- File store under .\Scripts. This script will copy Apache Jmeter from Axure Blob storage to C drive in Azure VM. Once package is copied it will call next script.

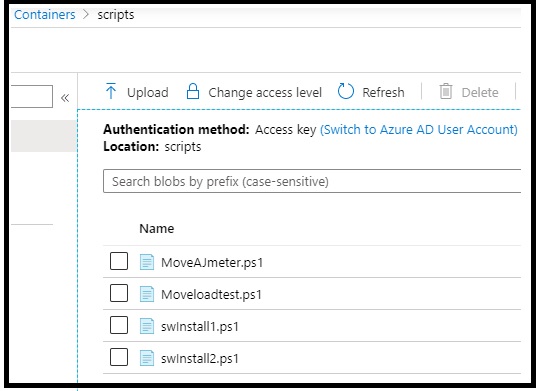
azcopy copy "https://XXXXXXX.blob.core.windows.net/ajmeter?SHARED\_ACCESS\_SIGNATURE" "C:\" --recursive=true ;

.\Moveloadtest.ps1

4.4 **Moveloadtest.ps1**:- File store under .\Scripts. This script will copy

azcopy copy "https://XXXXX.blob.core.windows.net/loadtestdef/EastLoadDefinition.jmx?SHARED\_ACCESS\_SIGNATURE" "C:\ajmeter\apache-jmeter-5.2.1\bin" --recursive=true ;

azcopy copy "https://XXXXX.blob.core.windows.net/loadtestdef/WestLoadDefinition.jmx?SHARED\_ACCESS\_SIGNATURE" "C:\ajmeter\apache-jmeter-5.2.1\bin" --recursive=true ;

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/Script_Storage.jpg)

**Execute work load**

The next step is to setup Azure VM in different regions, install software, configure, manage file and execute the work load. Copy folder **ClientSide\_Scripts** on local drive. This folder contains two **jmx** file assuming to run workload from two regions (East and West). If you want to run load from multiple region then change in the below script and execute it seperatly. Soon I will automate this as well.

Below is the extract from **.\ClientSide\_Scripts\ExecuteScriptMain.ps1** file.

<# Block 1 :- Connet to Azure Subscription #>

Connect-AzAccount

Add-AzureAccount <#Selet account if you have multiple accounts in one subscription#>

<# End of Block 1 #>

<# Block 2 :- Create a New VM on Azure #> $ResourceGroupName ="XXXXXXXXX"

$VMName = "XXXXXXXXXXX"

$VMLocation = "EastUS"

$VMSize = "Standard\_F32s\_v2" #Check for other VM sizes on Azure https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes

$VMUsername="XXXXXXXX"

$VMPassword=ConvertTo-SecureString Password@123 -AsPlainText -Force

$VirtualNetworkName = "XXXXXXXVnet"

$SubnetName = "XXXXXXXXXXX"

$PublicIpAddress="XXXXXXXXXXXXXXXX"

$SecurityGroupName = "XXXXXXXXXXX"

$ImageName = "MicrosoftWindowsDesktop:Windows-10:19h1-pro-gensecond:18362.720.2003120536"

New-AzVm `

-ResourceGroupName $ResourceGroupName `

-Name $VMName `

-Location $VMLocation `

-VirtualNetworkName $VirtualNetworkName `

-SubnetName $SubnetName `

-SecurityGroupName $SecurityGroupName `

-PublicIpAddressName $PublicIpAddress `

-Image $ImageName `

-Credential $cred `

-OpenPorts 3389

<# End of Block 2 #>

<# Block 3 :- Install and configure software #>

$StorageAccountName= "XXXXXXXXXXX"

$StorageAccountKey= "XXXXXXXXXXXXXXXXXXXXXXXXXX"

$fileUri = @("https://XXXXXXX.blob.core.windows.net/scripts/swInstall1.ps1",

"https://XXXXXXXXXXX.blob.core.windows.net/scripts/swInstall2.ps1",

"https://XXXXXXXXXXXX.blob.core.windows.net/scripts/MoveAJmeter.ps1",

"https://XXXXXXXXXXXXXX.blob.core.windows.net/scripts/Moveloadtest.ps1")

$settings = @{"fileUris" = $fileUri};

$VMLocation = "East US"

$protectedSettings = @{"storageAccountName" = $storageAcctName; "storageAccountKey" = $storageKey; "commandToExecute" = "powershell -ExecutionPolicy Unrestricted -File swInstall1.ps1"};

Set-AzVMExtension -ResourceGroupName $ResourceGroupName `

-Location $VMLocation `

-VMName $VMName `

-Name "SetupLoadMachine" `

-Publisher "Microsoft.Compute" `

-ExtensionType "CustomScriptExtension" `

-TypeHandlerVersion "1.10" `

-Settings $settings `

-ProtectedSettings $protectedSettings

<# End of Block 3 #>

<# Block 4 Restart VM after s/w installation #>

Restart-AzVM -ResourceGroupName $ResourceGroupName -Name $VMName

<# End of Block 4 #>

<# Block 5 :- invoke script #>

Invoke-AzVMRunCommand -ResourceGroupName $ResourceGroupName -VMName $VMName -CommandId 'RunPowerShellScript' -ScriptPath 'LOCAL\_DRIVE:\ClientSide\_Scripts\EastRunloadtest.ps1'

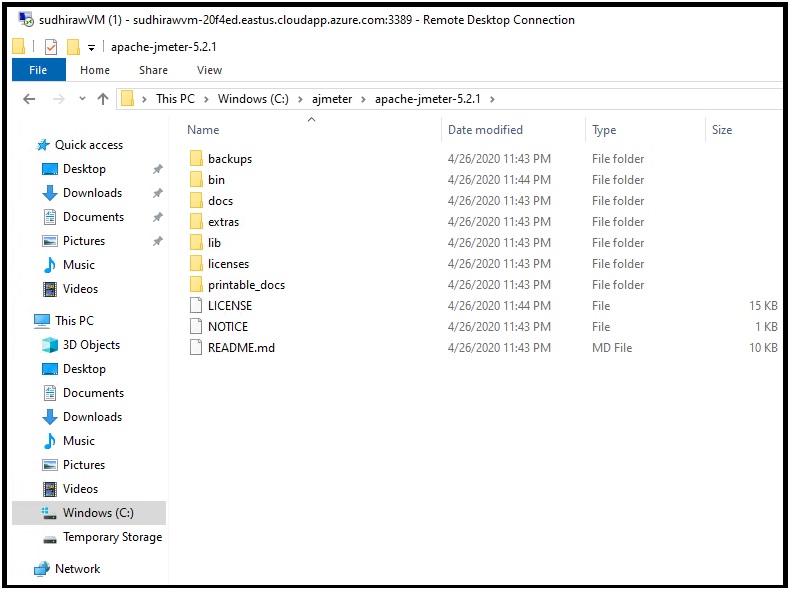
<# End of Block 5 #>

<# OPTIONAL Remove Resource Group Once perf testing is done.#>

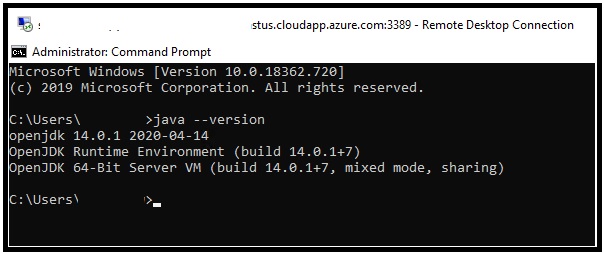
Remove-AzResourceGroup -Name "XXXXXXX"

<# END #>

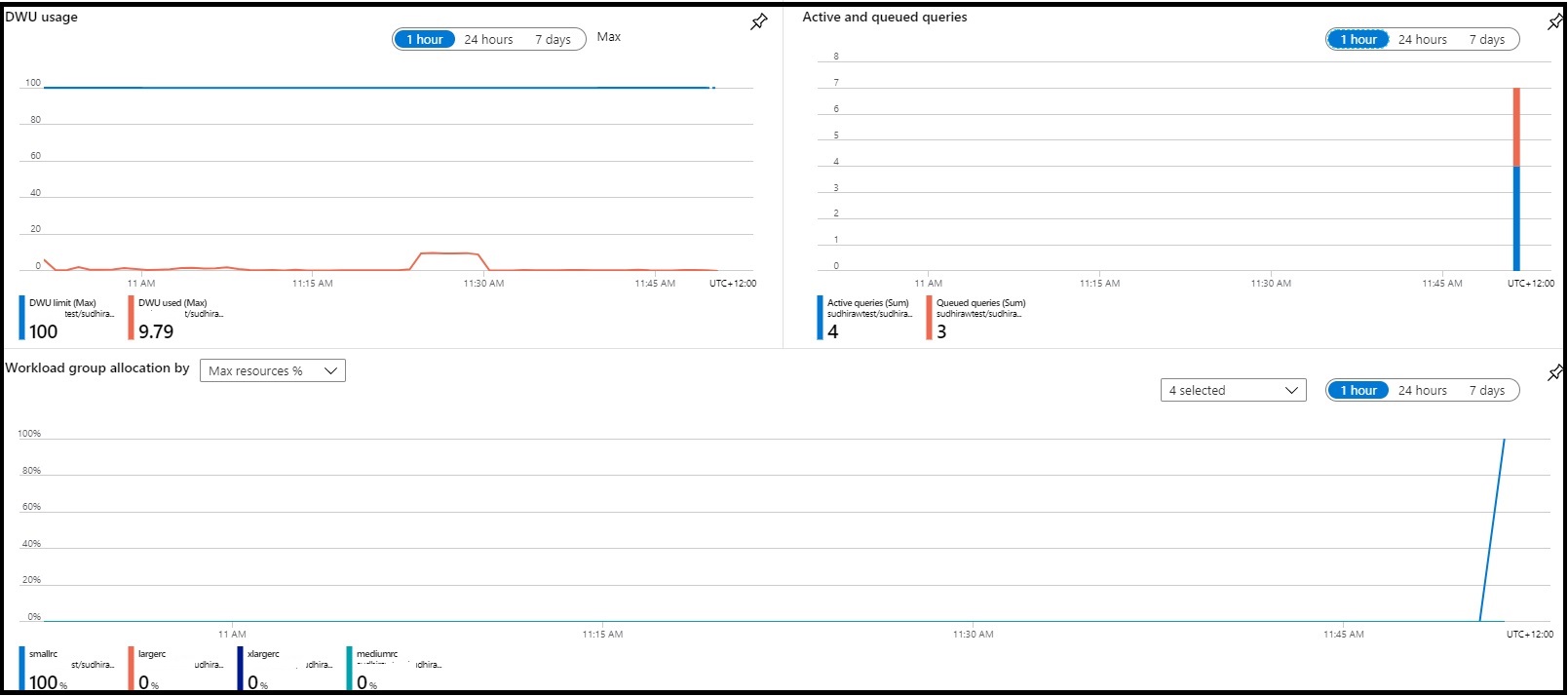
Once Azure VM is setup and software is configured. Below is the screenshot of files copied.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/fileCopied.jpg)

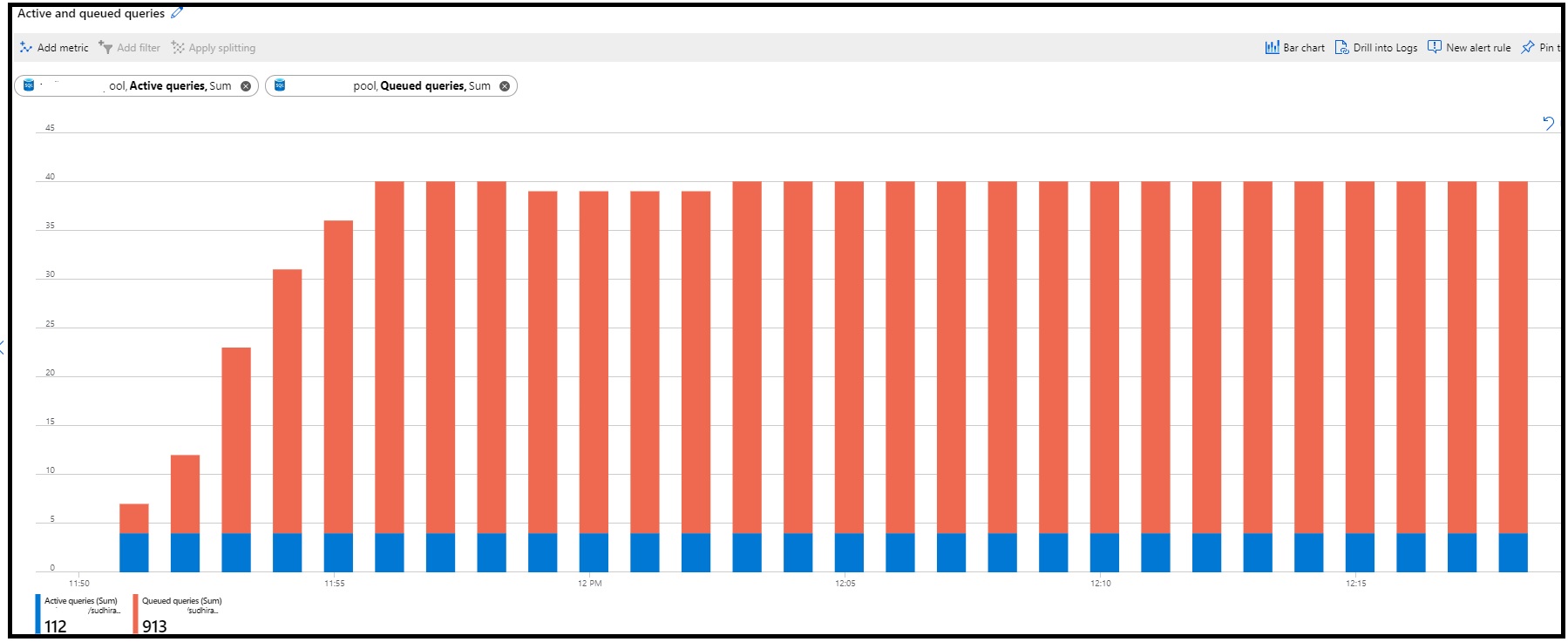
Below is the screenshot of Java installed on VM

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/JREinVM.jpg)

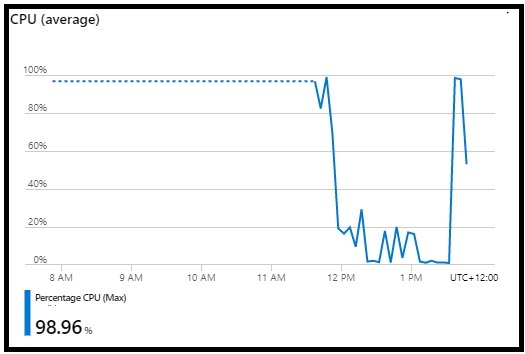
Once you execute block-5 in Powershell, after sometime notice request started coming in...

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/LoadStarted.jpg)

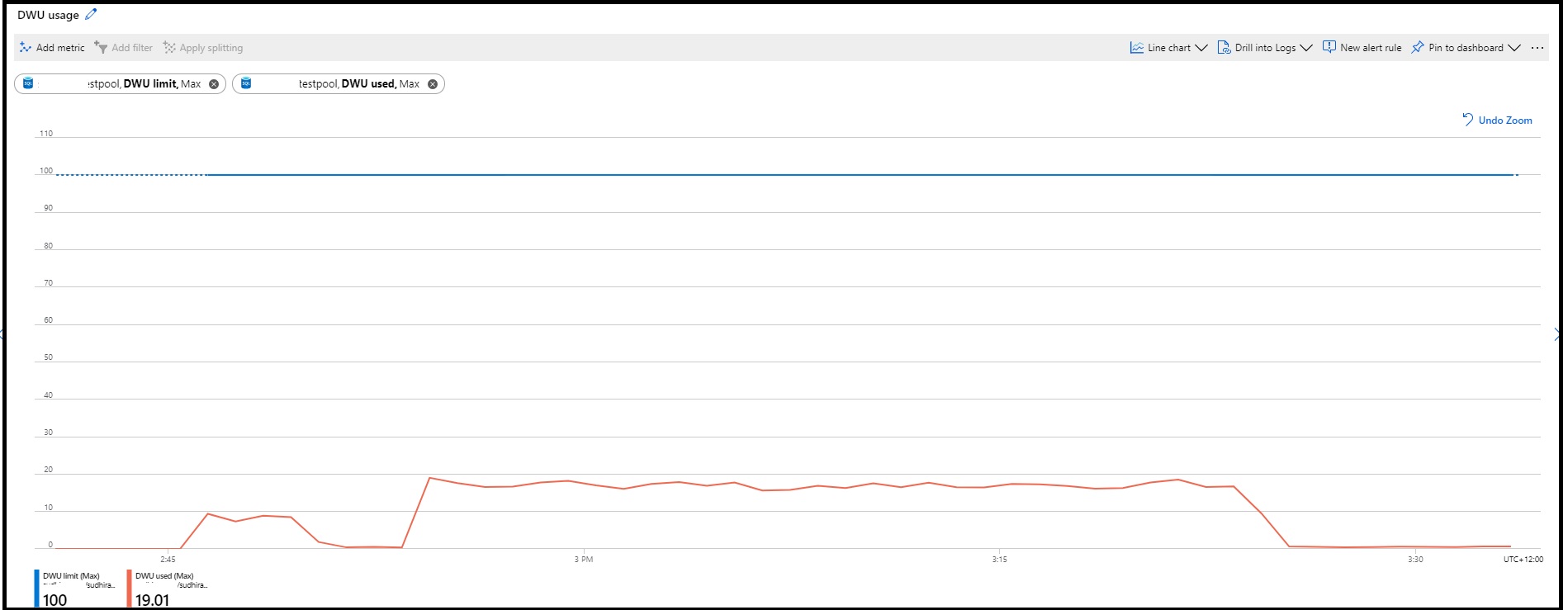
Over the period of time...

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/QueryExecutionStatus.jpg)

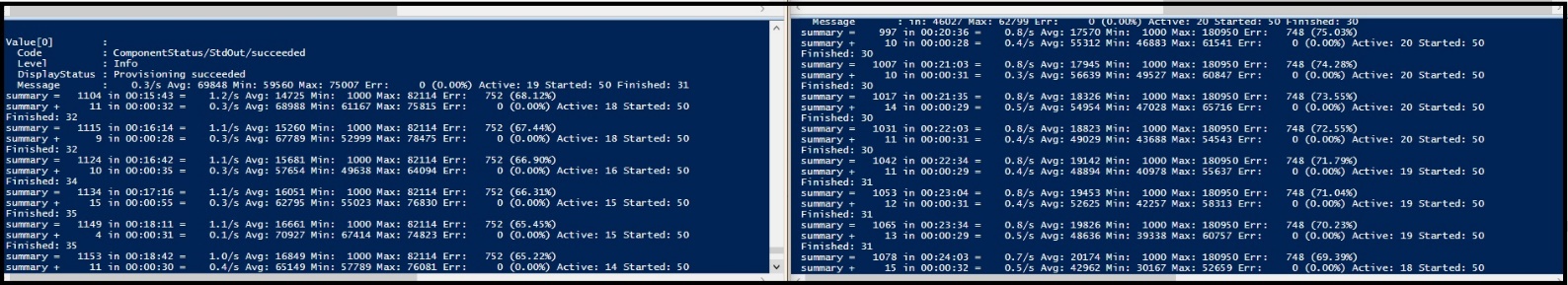
VM cpu usage....

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/CPU_VM.jpg)

Data warehouse unit usgae....

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/DWU.jpg)

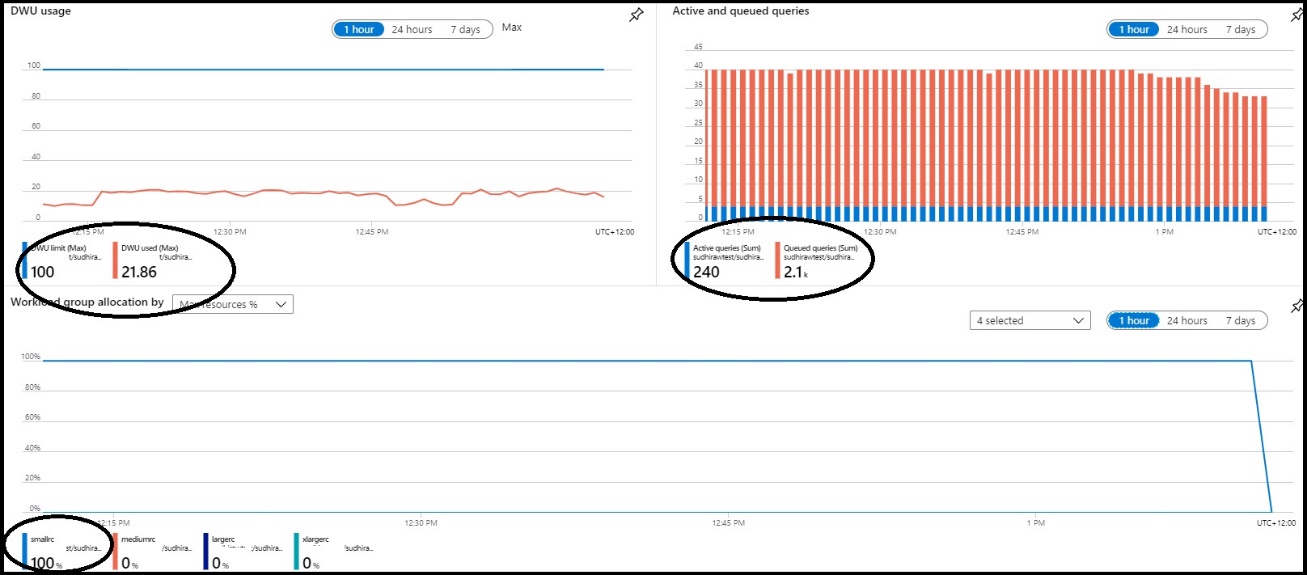
Powershell script execution ends...

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/PowerShellOutput.jpg)

**Analyse the Test Results**

**Investigate result over dashboard**

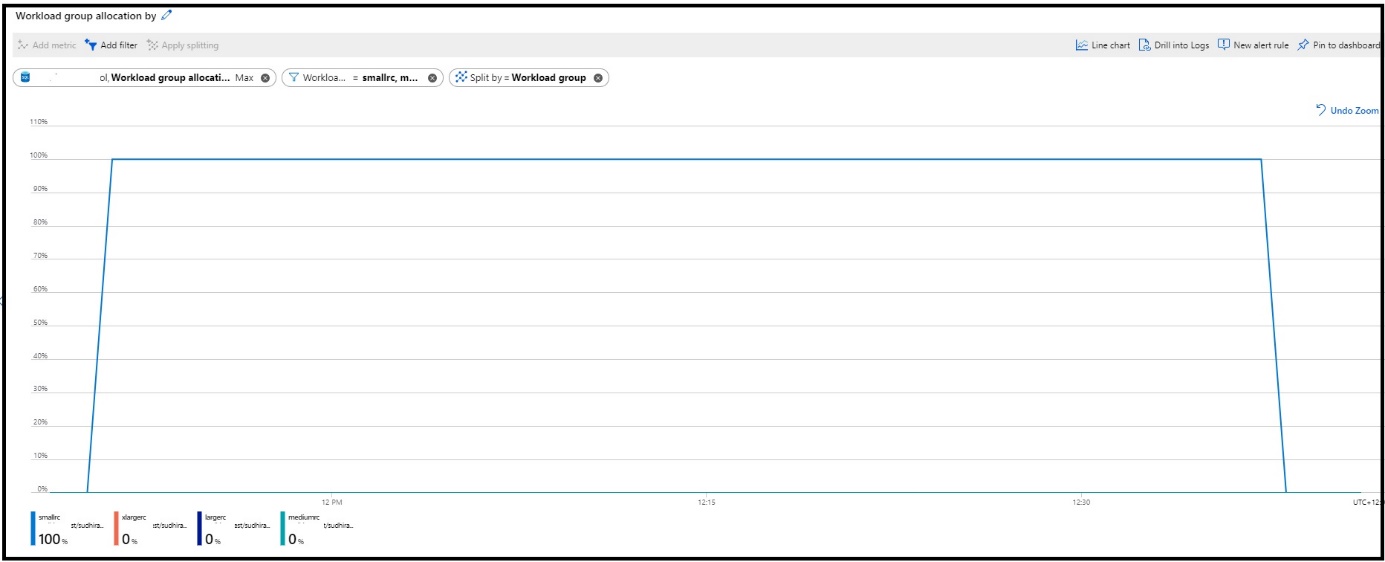
Let's investigate from the dashboard in Azure portal.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/Problem_monitor.jpg)

By looking onto the dashboard it's clear that service is not utilizing the allocated resources. The service is utilizing around 22% of total number of DWU selected. The queries getting queued and waiting for resources to be availble. Though 75% is still unused resources. Why this is happening and how to solve it?

Notice all queries executes under smallrc workload group. Workload group in Azure Synapse DWH enables to reserve resources for an defined workload group. We can define different workload group like "business\_group", "Q&A\_group", "Dev\_group", "normal\_user" etc. Setting importance to each workload group define the priority given to individual workload group. So query received from "business\_group" workload group gets higher priority than "normal\_user" workload group. Apart from definining resource and importance we can define query timeout as well. This will help to kill long running queries in production.

In this case, we didn't plan for workload group and went ahead with default workload group. There were 4 workload group defined by deault (smallrc, mediumrc, largerc and xlargerc). The memory allocation for smallrc is 25% (based on the service level). By this allocation Azure Synapse DWH utilized 88% (or 100%) of allocated resources.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/ResourceAllocation.jpg)

**Leason learned** :- Plan and setup workload group.

More information about setting up different resource class is explained [here](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/resource-classes-for-workload-management).

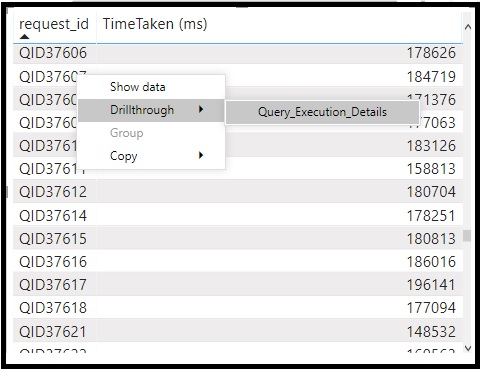
**Analyse in PBI**

PowerBI has connector to Azure Synapse DWH. After successful connection, query **sys.dm\_pdw\_exec\_requests** system table. This table contains information about current or recent queries received. There is another table **sys.dm\_pdw\_request\_steps** provides detail information about the query execution.

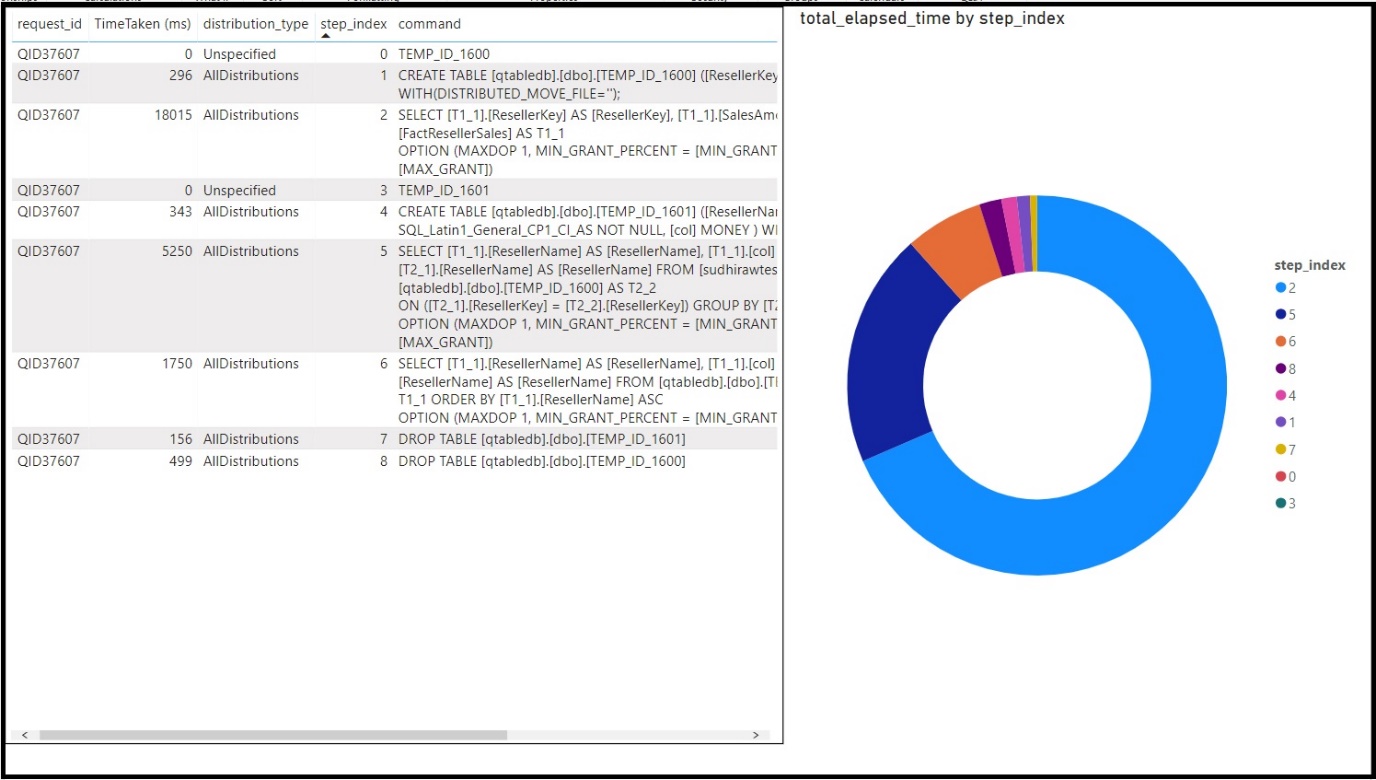
Below is the PBI dashboard to analyse the queries execution on server.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/QueryExecutedReport.jpg)

Look at the each query and time taken can help in identify the queries taking so much time.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/QuerySummary.jpg)

Further drill down explains steps taken to prepare the resultset.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/QueryDetails.jpg)

Investigate

1) Table Distribution

2) Partition stratergy

2) Index

3) Cache

4) Statistics

**Table Distribution**

Generally, in DWH world there are two types of tables available 1) Dimension 2) Fact. Fact table store information about transaction like sales. As per the dimensional modeelling fundameentals this type of table contains numerical value. Dimension table store information about dimension of facts. For example

Fact Table

| **ID** | **Country** | **Sales** |
| --- | --- | --- |
| 1 | 1 | 5000 |
| 2 | 4 | 4000 |

Dimenson Table

| **ID** | **Country** | **Name** |
| --- | --- | --- |
| 1 | 1 | US |
| 2 | 2 | IN |
| 3 | 3 | SIN |
| 4 | 4 | NZ |

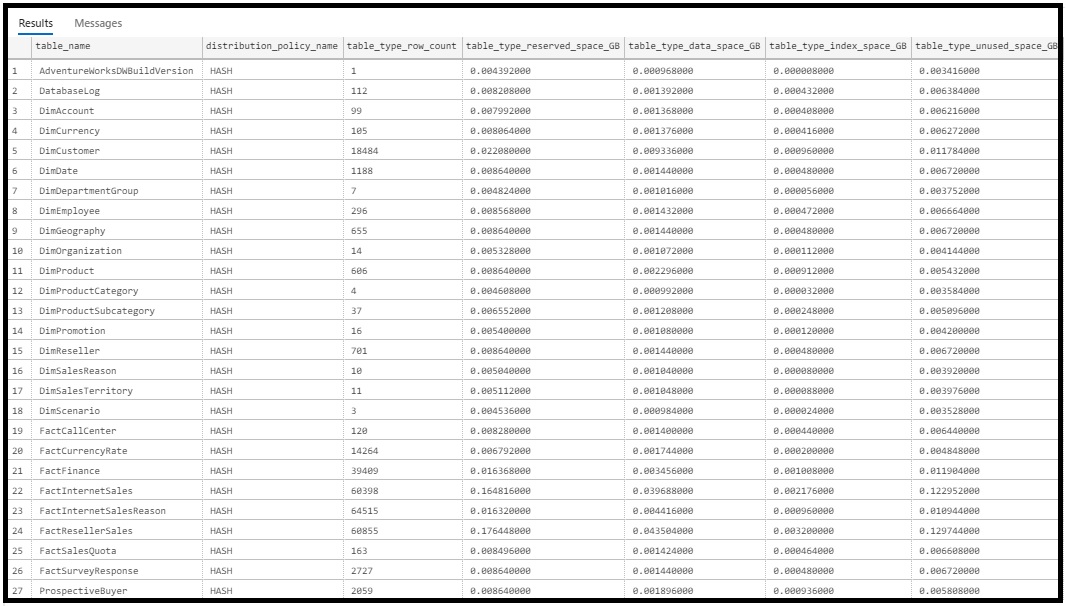
There are three ways in [Table Distribution](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/massively-parallel-processing-mpp-architecture#distributions) by which a table is distributed in compute node.

1) Hash distributed

2) Round-robin

3) Replicated

In this blog we used sample database, the service by default applied hash distribution. However genral rule of thumb is to select **Replicated** distribution type when size of the table is low (generally dimensional table). **Hash distribution** provdes high performance for joins and aggregated queries. A hash alogrithm assign each row to the distribution. This will good for a table expecting high performaing queries (generally fact table). **Round robin** distribution will store a row in random fashion. Hence it will be good for loading data but not for queries. Hence it's good to use round robin distribution while staging the data.

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/Table_Distribution.jpg)

**Partition Stratergy**

Partition allows rows to store in a range which allow query to find result quickly as compare to no partition. Use [Partition startergy](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-tables-partition) to understand how partition can be benefit while load and query. Also consideration to design partition.

**Index**

Index makes reading a table become faster. There are clustered, nonclustered, clustered columnstore index and non-index you can define on a table in Azure synapse DWH. More information about benefits and designing index stratergy can be found [here](https://docs.microsoft.com/bs-latn-ba/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-tables-index)

**Cache**

Some queries with same or no filter run frequently on the server. Each time query completion takes let's say x time. Think about query outcome cache in such scenario to speed the query performance. Can any feature in Azure Synapse DWH help in reducing the time for similar data which changes infrequently? The answer is Yes. Azure synapse DWH has two features 1) Result Cache and 2) Materialize view.

**Result cache** :- Enable result cache allow subsequent query execute faster. It hits if receive same query which is used to built the cache also result should apply to entire query. Query result cache in SQL Pool and it's available even pause and resume DWH.

Below quries executed in **Azure Data Studio** to test the Result cache feature.

SELECT [SalesTerritoryCountry]

, [SalesTerritoryRegion]

, SUM(SalesAmount) AS TotalSalesAmount

FROM dbo.factInternetSales s

JOIN dbo.DimSalesTerritory t ON s.SalesTerritoryKey = t.SalesTerritoryKey

GROUP BY ROLLUP (

[SalesTerritoryCountry]

, [SalesTerritoryRegion]

)

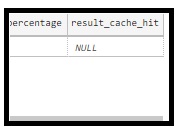
OPTION (LABEL = 'ResultCache session level on');

--Total execution time: 00:00:02.300

SELECT \*

FROM sys.dm\_pdw\_exec\_requests

WHERE request\_id='QID44845'

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/Miss_Cache.jpg)

After RESULT\_SET\_CACHING ON, first query took around 00:00:02.325, similar query next time hits cache and completes in 00:00:00.734.

SELECT [SalesTerritoryCountry]

, [SalesTerritoryRegion]

, SUM(SalesAmount) AS TotalSalesAmount

FROM dbo.factInternetSales s

JOIN dbo.DimSalesTerritory t ON s.SalesTerritoryKey = t.SalesTerritoryKey

GROUP BY ROLLUP (

[SalesTerritoryCountry]

, [SalesTerritoryRegion]

)

OPTION (LABEL = 'ResultCache session level on');

--Total execution time: 00:00:02.325

--ALTER DATABASE XXXXXXXX --at master DB level

--SET RESULT\_SET\_CACHING ON; --at user DB level

SELECT [SalesTerritoryCountry]

, [SalesTerritoryRegion]

, SUM(SalesAmount) AS TotalSalesAmount

FROM dbo.factInternetSales s

JOIN dbo.DimSalesTerritory t ON s.SalesTerritoryKey = t.SalesTerritoryKey

GROUP BY ROLLUP (

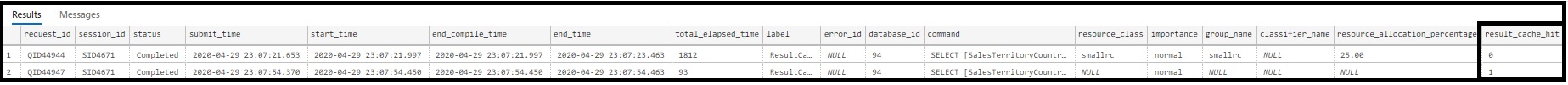
[SalesTerritoryCountry]

, [SalesTerritoryRegion]

)

OPTION (LABEL = 'ResultCache session level on');

--Total execution time: 00:00:00.734

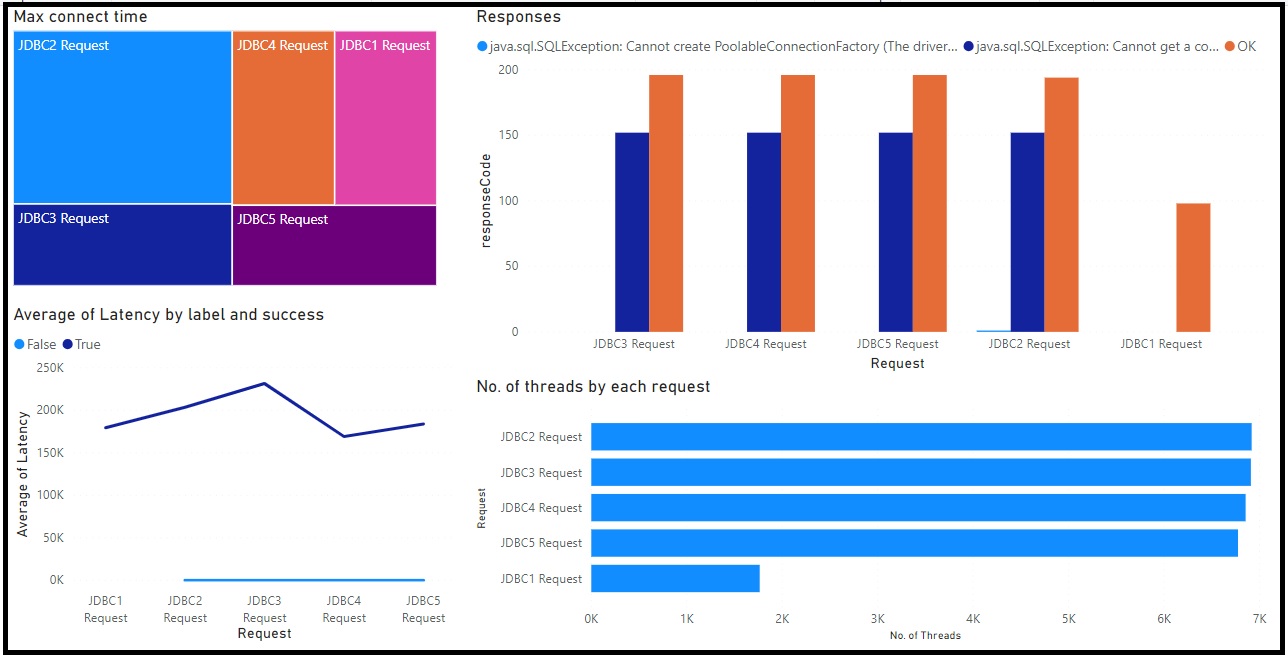
[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/CacheHit.jpg)

**Materialize View** :- Use when there were multiple joins and complex data computation presence in the query. Materialize view stores the result in logical tables. So there will be extra storage taken by this feature and hence provides result faster. This is the main difference between standard and materialize view. As compare to Result Cache, with Materialize view we can get subset of view.

**Leason learned** Plan for Result cache and/or Materialize view if possible to speed up query performance.

**Apache Jmeter Report**

Latency, connection or any other errors received at client side will be store in csv file (Azure VM c:\ drive as that's what configured while setting up jmx file).

[](https://github.com/rawatsudhir1/AzureSynapseDWH/blob/master/images/AJmeter.jpg)

Find below some more resources to know more about Azure Synapse Analytics

[here](https://docs.microsoft.com/en-us/learn/paths/implement-sql-data-warehouse/) is the learning path to learn more about Azure Synapse Analytics.

[here](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-how-to-manage-and-monitor-workload-importance) to manage and monitor workload performance.

[here](https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-cheat-sheet) for cheat sheet.

I will add more soon in future but for now thanks for reading till here 😃

**Eat Healthy, Stay Fit and Keep Learning**