In this clip, you will learn to create an Azure SSIS integration runtime.

You can create the SSIS integration runtime from the getting started page, Or you can create it from the Integration Runtime section too.

If you are creating either the Azure integration runtime or the Self-hosted integration runtime, you would choose the first option, as they can perform both the data movement and the dispatch activities to other computers.

For executing SSIS packages in Azure, choose the 2nd option, which creates an SSIS integration runtime. In the SSIS integration runtime, you cannot perform any data movement or you cannot dispatch any activity to external computers. They can only be used to execute SSIS packages.

Location of the runtime, I will select East US 2, as the SQL Database is also located in East US 2.

It is Microsoft’s recommendation that you create the Integration Runtime in the same region as your Database Server, as this allows the runtime to write execution logs to SSISDB without crossing Azure regions.

Also, I had created the data factory in the East US location, to show you that it is not mandatory to have the integration runtime and the data factory in same regions and that it is a common practice.

Select a node size for the underlying VM that will execute your SSIS Packages. Mostly you have the D-Series and the E-Series machines at the time of recording. I will select one that has the least configuration, as I am not doing any heavy operation and, if there is a need, I can always scale up the machine.

And set the node number to 3. So this will create 3 VMs, loaded with SQL Server ready to run your packages.

Choose either a standard or the Enterprise edition of the SSIS integration runtime. The Enterprise edition will allow you to use Premium features like Change Data Capture, Oracle connectors, SQL Server Analysis Services, Fuzzy grouping, and so on. And as expected, the per hour cost of the VMs, if you choose the enterprise edition, is higher than the standard edition.

To allow customers to move to Azure in a more cost-effective way, Microsoft has provided the Azure Hybrid benefit, so if you have SQL Server licences with Software Assurance, you save some money there.

At the bottom, you also get to see the cost of running 3 VMs per hour.

When you create an Azure SSIS Integration Runtime, the Integration Runtime connects to your SQL Database to prepare the SSISDB. So, specify a SQL Database that does not already have the SSIS Catalog. Else, you will get an error.

Configure the connection settings with Username and password. If you are a little concerned about storing the password, your concerns are not misplaced. Wait until the next module.

The Service Tier for SSIS Database affects the speed to queue a package for execution in Integration Runtime, and the speed to load the execution log. If you don't care about those, choose the lowest database pricing tier.

Having said that, you need to scale the database, according to the number of cores selected for the Integration Runtime, else the database will be the bottleneck and the package execution performance will be affected.

Let’s Test the connection.

Under Advanced Settings, select the maximum number of packages to execute per node, at any point in time, So this is one of the decisions you need to make while creating the runtime.

If you have a few heavy SSIS packages, select a low number. If you have many light-weight SSIS packages, Select a high number, and this number, whether high or low, is in reference to the no. of. cores in the Integration Runtime.

Suppose, there is a node with 4 cores and a SQL database deployed with 4 heavy packages.

In this case, choose a value lower than the no. of. cores for the maximum parallel execution per node, so if set to 2, Only 2 packages will be executed concurrently, so 2 heavy packages will be executed on 4 cores, or, it uses more than 1 core to execute your package.

If instead of 4 heavy packages, there are 8 light-weight packages, in this case, set a value higher than the no. of. cores for the maximum parallel executions per node, so if set to 6, some of the cores will execute more than 1 package, so one core executes multiple packages.

Let’s leave the Custom setup for Integration Runtime and the VNet integration for the next module, and create.

Creating this runtime will take some time, approximately 20 minutes, so in the meantime, I will show you the JSON definition for this integration runtime.

All the objects that you create in Data Factory will have an associated JSON definition, so you can check out the properties in this definition.

So you can view the properties of the node and the SQL database.

And once the Integration Runtime is created, let’s confirm that it has created the SSIS database as well.

This SSIS integration runtime is billed every hour for the underlying Virtual Machines. So to save cost, you can stop the runtime when you do not execute any package.