SQL Diagnostics Tool Belt

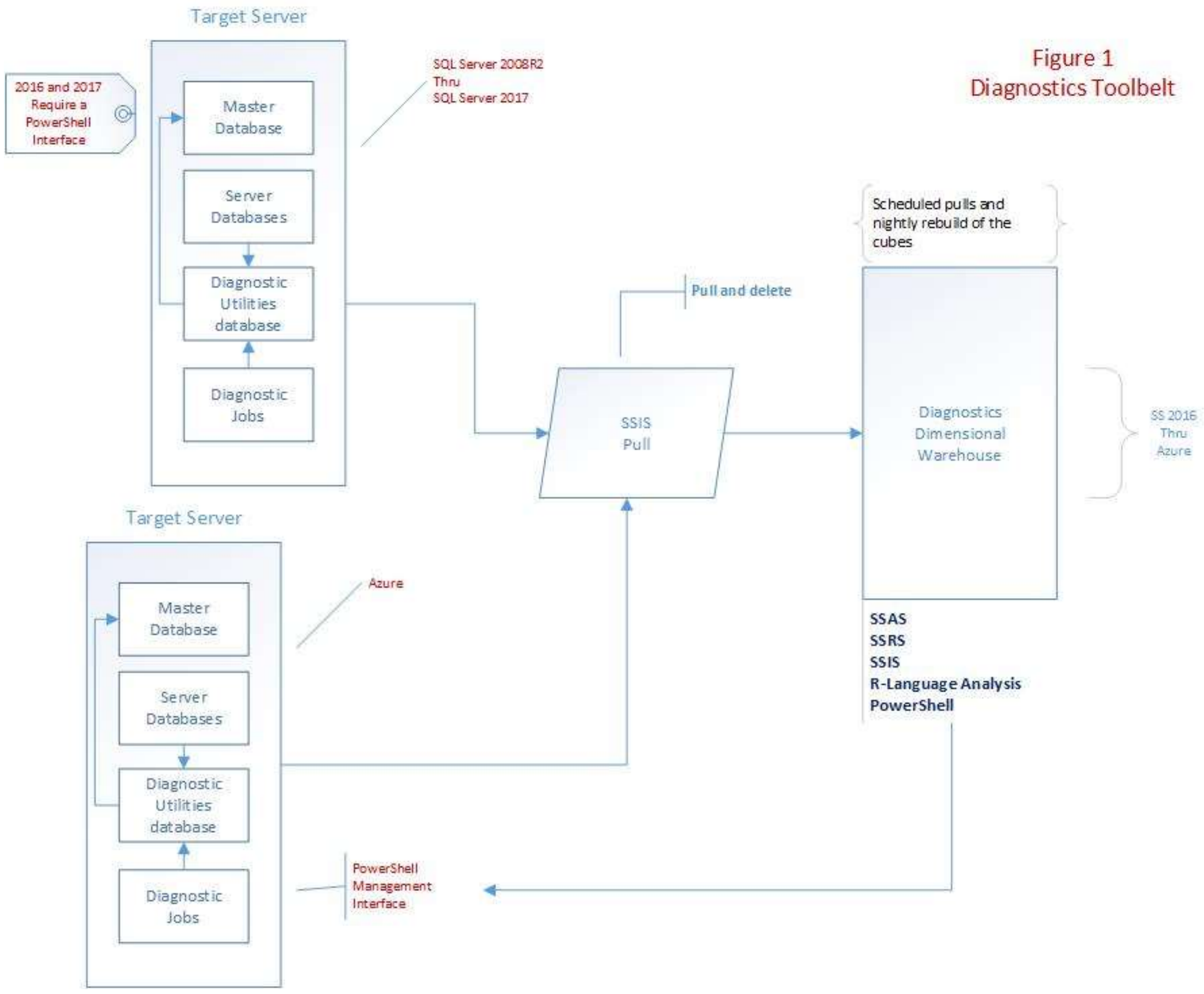
Figure 1
Diagnostics Toolbelt
Target Server
2016 and 2017 Require a PowerShell Interface
Master Database
Server Databases
Diagnostic Utilities database
Diagnostic Jobs
SQL Server 2008R2 Thru SQL Server 2017
Scheduled pulls and nightly rebuild of the cubes
Pull and delete
SSIS Pull
Diagnostics Dimensional Warehouse
SS 2016 Thru Azure
Target Server
Master Database
Server Databases
Diagnostic Utilities database
Diagnostic Jobs
Azure
SSAS
SSRS
SSIS
R-Language Analysis
PowerShell
PowerShell Management Interface

IVP

The installation verification procedure does all the work for installing the Tool Belt. This procedure is a set of SQL Scripts that create, install, compile, and validate completion. Because of differences between the various versions of SQL, the IVP will be specific for a certain version, or will contain the necessary modifications to make it run on all versions.

The IVP is generated through command line batch files and reside in the "D:\dev\SQL\DFINAnalytics\BATCH_FILES" directory. For the "run on all" build, execute the batch file "Build_Client_DFINAnalytics_IVP.bat". This execution will generate a file, "IVP_DFINAnalytics.sql", and this generated script should be executed on a target server to install the needed procs and tables.

The first step is to create a database named specifically "DFINAnalytics" and run the IVP from within this database.

Target Server

This is a SQL Server between the versions 2008R2 and Azure present. Basically there is no limit to the number of databases housed on the server. For SQL version 2008R2 thru 2017, ***permission to write procedures to the Master Database*** is required.

TDR Warehouse

This is the warehouse that will keep the data that is pulled from each of the monitored servers. The data is then placed into cubes through a nightly run and dashboards and reports can be generated to report potential problems with load, performance, scheduling, etc. as needed.

SSIS Pull

This is an SSIS job that polls each monitored target server, transfers the data from the target to the TDR Warehouse, and then removes that captured data from the monitored target server. The SSIS job can run on an SSIS Server or be implemented through the TDR Warehouse

SQL 2008R2 thru SQL 2014

In these versions of SQL, the non-supported functions For Each DB, For Each Table, and the "sp_" prefixed stored procedures within the Master DB are heavily used.

SQL 2016 & SQL 2017

In these versions of SQL, the non-supported functions For Each DB and For Each Table are still available, The "sp_" prefixed stored procedures within the Master DB are NO LONGER AVAILABLE. The workaround was to develop code that would take an existing procedure and convert it on the fly to be compiled and run under TempDB.

Our final solution, which will replace the above needs, will be the same PowerShell implementation that will work across ALL versions of SQL Server.

Azure

In Azure, For Each DB, For Each Table, and the "sp_" functionality is no longer available. Therefore, this required functionality is implemented through PowerShell.

PowerShell Management Interface

This interface uses a small set of PowerShell scripts to manage Tool Belt scripts that have to traverse all databases on a given target server. These scripts read from a file containing databases to process OR from the table in DFINAnalytics that houses databases to be processed.

The tables used for this functionality are:

- `[DFIN_TRACKED_DATABASES]`
- `[DFIN_TRACKED_DATABASE_PROCS]`

- `NOTE: There is a zero-to-many relationship between [DFIN_TRACKED_DATABASES] and [DFIN_TRACKED_DATABASE_PROCS]. This allows a procedure, by name, to be applied to one or ALL target databases.`

SQL Server Jobs

From SQL Server 2008R2 thru 2017, the agent is used to schedule and run jobs. These jobs are installed using a set of scripts or one Master Job Script. Once installed, be sure and manually fire off each job to make certain it runs under the newly installed environment without incident.

Azure Scheduling

The Automation service in Azure is the primary tool for those needing to schedule tasks. Like the Azure Scheduler service, this is very different to SQL Server Agent, but this is the cloud PaaS world. The Azure Automation service uses ***runbooks*** to manage, schedule and define its jobs. These contain *command script JSON documents* that store PowerShell commands.

For those un-familiar with JSON and PowerShell, there's a gallery of pre-written runbooks that can be used. Helpfully, for those looking to replace SQL Server Agent, there's one that executes a T-SQL command. The tools are there, it just needs the time to set them up.

Simplicity, however, is often traded for capability and the Azure Automation is perhaps sadly a good example of this. It uses an Azure Active Directory based security model that can take time to setup, it can't schedule something to run less than once an hour and its best management interface is the Azure Portal.

However, I've seen application development teams who never thought to use it to schedule index maintenance for their Azure SQL Database service databases. For them, it was 30 seconds of clicks, copy and pasting, and scheduling. For database administrators, I recommend it's something they add to their learning plan.