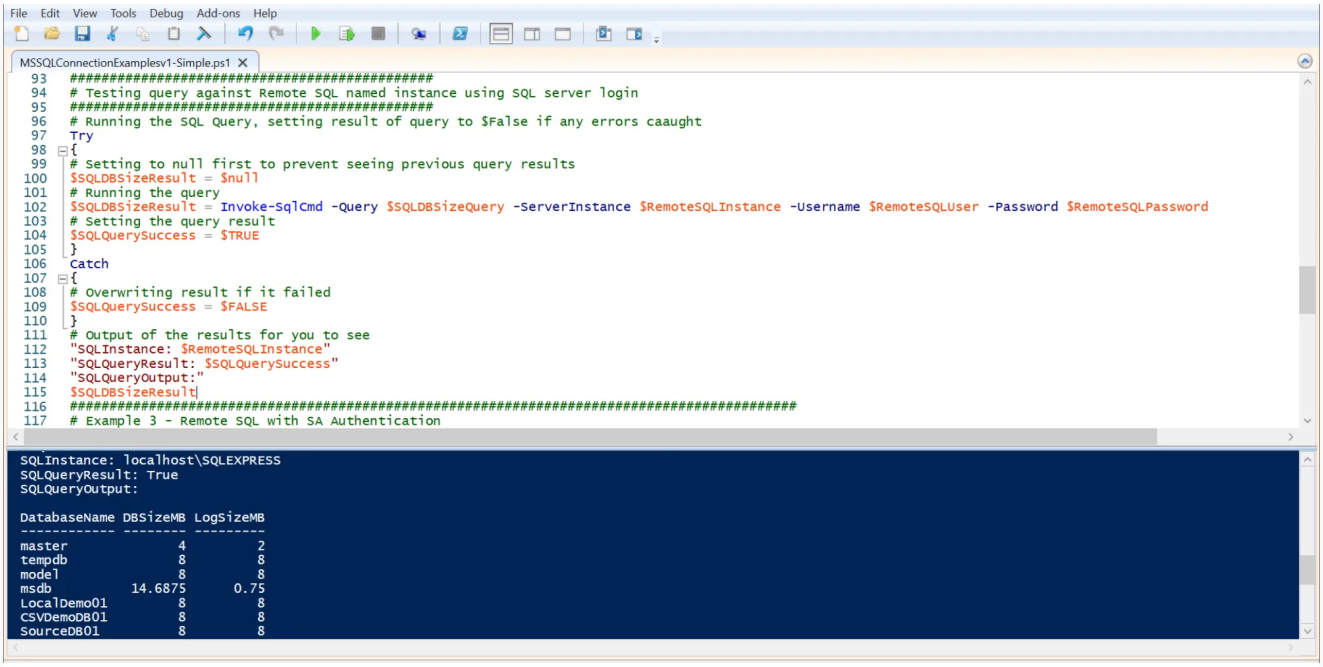
Connecting to Microsoft SQL Databases using PowerShell Invoke-Sqlcmd

1

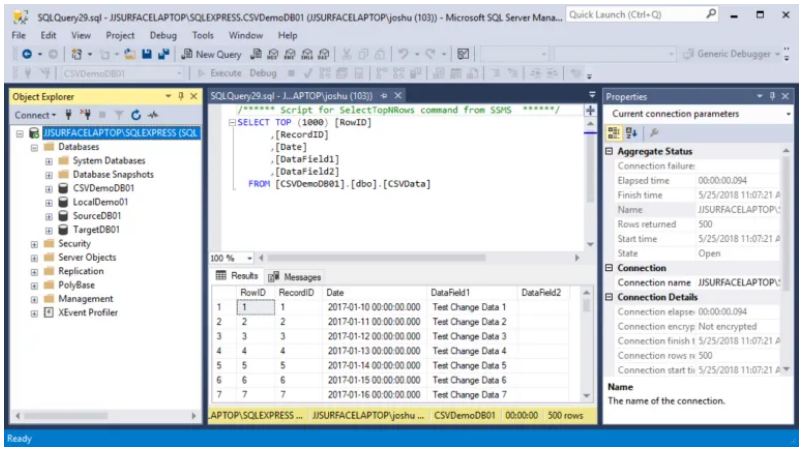
Published May 30, 2018 by Joshua Stenhouse



Since installing Drift on my blog I get to see and answer questions from my readers real-time. Lately, I've had a whole bunch of questions about Microsoft SQL and PowerShell Invoke-Sqlcmd. From basic things such as connecting to different SQL servers to importing a CSV, reporting on SQL DB sizes, deleting DBs, and transforming SQL data using PowerShell. It's been
challenge.

To share this work, I've decided to wr
I'm being asked, starting with the mo
different types of SQL instance with A

Joshua Stenhouse
Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.
Type your message…
Chat ⚡ by Drift



If you can't connect and authenticate with the SQL Instance, then you'll never be able to work with any SQL DB using PowerShell. So, to help you get over this initial hurdle I'm going to share with you the following Invoke-Sqlcmd connection examples:

- Local SQL instance using Windows authentication
- Remote SQL instance using SQL authentication
- Remote SQL default instance using SA authentication

Download all 3 examples from the .zip file below:

MSSQLConnectionExamplesv1.zip

Included are 2 scripts, simple and advanced. Simple stores the password in plaintext so you can see the method. Advanced securely stores the credentials in an xml file and it includes a ping test to the SQL hostname to help you troubleshoot. The scripts were tested using Windows 10, PowerShell 5.1 and a mix of SQL Server 2016 and 2017. I see no reason they won't work on earlier versions of SQL, but I advise against anything lower than PowerShell 5. Included in both is a sample query for getting the size of any DBs on the SQL instance, which is use
to whatever you want.

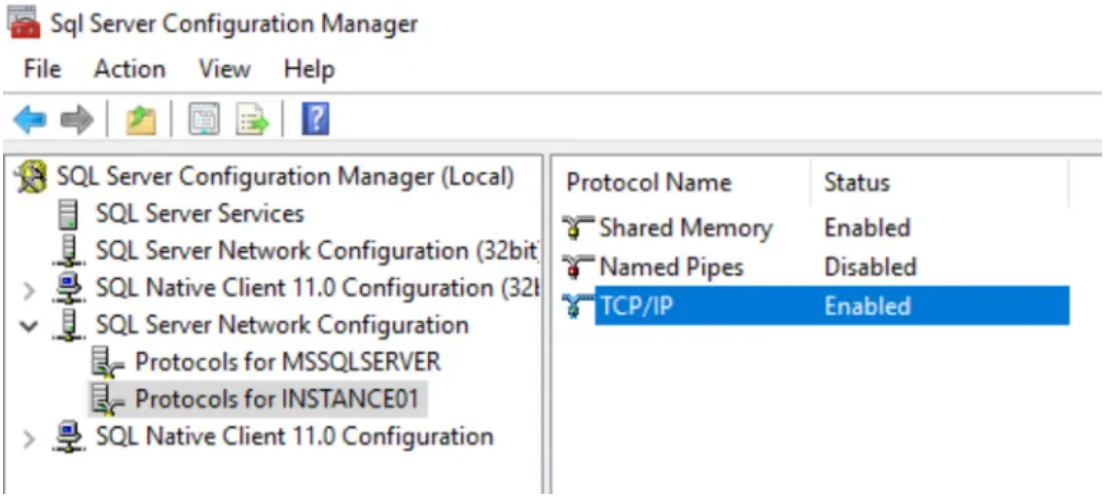
Make sure remote connections are er

Joshua Stenhouse

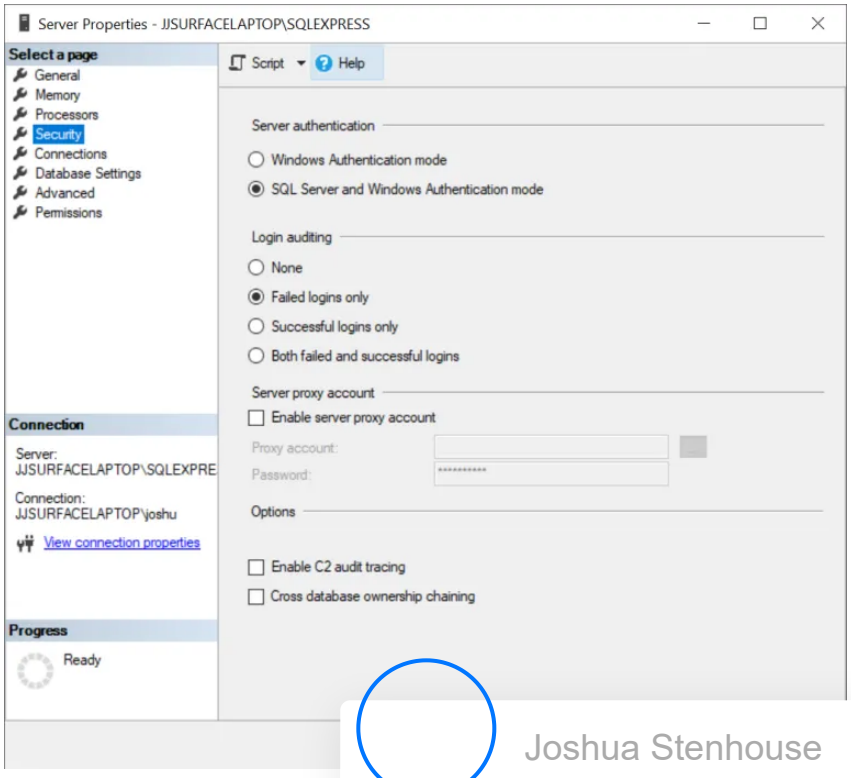
Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.

Type your message…

Chat ⚡ by Drift



And when using SQL/SA, that their login is enabled:



If you'd rather not download a zip file from below;

```
########################
# Configure the variables below
########################
# Local SQL instance using Windows
script
$LocalSQLInstance = "localhost\SQLEXPRESS"
# Remote SQL named instance using SQL server login
$RemoteSQLInstance = "SQL16-VM01.lab.local\INSTANCE01"
$RemoteSQLUser = "localsqluser"
```

Joshua Stenhouse
Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.
Type your message…
Chat ⚡ by Drift

```
$RemoteSQLPassword = "Srt1234!"
# Remote SQL default MSSQLSERVER instance using sa login
$RemoteDefaultSQLInstance  = "SQL16-VM01.lab.local"
$RemoteDefaultSQLInstanceUser = "sa"
$RemoteDefaultSQLInstancePassword = "Srt1234!"
#########################################################################
# Nothing to change below this line, commented throughout to explain
#########################################################################
###############################################
# Checking to see if the SqlServer module is already installed, if not
installing it for the current user
###############################################
$SQLModuleCheck = Get-Module -ListAvailable SqlServer
if ($SQLModuleCheck -eq $null)
{
write-host "SqlServer Module Not Found - Installing"
# Not installed, trusting PS Gallery to remove prompt on install
Set-PSRepository -Name PSGallery -InstallationPolicy Trusted
# Installing module
Install-Module -Name SqlServer –Scope CurrentUser -Confirm:$false -
AllowClobber
}
###############################################
# Importing the SqlServer module
###############################################
Import-Module SqlServer
###############################################
# Creating SQL Query to get the size of all DBs on each Instance
###############################################
# Change this to whatever you want to test, but this is a good one to start
with!
$SQLDBSizeQuery = "SELECT
    DB_NAME(db.database_id) DatabaseName,
    (CAST(mfrows.RowSize AS FLOAT)*8)/1024 DBSizeMB,
    (CAST(mflog.LogSize AS FLOAT)*8)/1024 LogSizeMB
FROM sys.databases db
    LEFT JOIN (SELECT database_id
WHERE type = 0 GROUP BY database_
db.database_id
    LEFT JOIN (SELECT database_id
WHERE type = 1 GROUP BY database_
db.database_id"
#################################
##
# Example 1 - Local SQL with Wind
#################################
##
###############################################
# Testing query against Local SQL instance using Windows auth
###############################################
# Running the SQL Query, setting result of query to $False if any errors
```

Joshua Stenhouse

Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.

Type your message…

Chat ⚡ by Drift

```
caught
Try
{
# Setting to null first to prevent seeing previous query results
$SQLDBSizeResult = $null
# Running the query
$SQLDBSizeResult = Invoke-SqlCmd -Query $SQLDBSizeQuery -ServerInstance
$LocalSQLInstance
# Setting the query result
$SQLQuerySuccess = $TRUE
}
Catch
{
# Overwriting result if it failed
$SQLQuerySuccess = $FALSE
}
# Output of the results for you to see
"SQLInstance: $LocalSQLInstance"
"SQLQueryResult: $SQLQuerySuccess"
"SQLQueryOutput:"
$SQLDBSizeResult
################################################################################
##
# Example 2 - Remote SQL with SQL User Authentication
################################################################################
##
#############################################
# Testing query against Remote SQL named instance using SQL server login
#############################################
# Running the SQL Query, setting result of query to $False if any errors
caaught
Try
{
# Setting to null first to prevent seeing previous query results
$SQLDBSizeResult = $null
# Running the query
$SQLDBSizeResult = Invoke-SqlCmd
$RemoteSQLInstance -Username $Rem
# Setting the query result
$SQLQuerySuccess = $TRUE
}
Catch
{
# Overwriting result if it failed
$SQLQuerySuccess = $FALSE
}
# Output of the results for you to see
"SQLInstance: $RemoteSQLInstance"
"SQLQueryResult: $SQLQuerySuccess"
"SQLQueryOutput:"
$SQLDBSizeResult
```

Joshua Stenhouse

Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.

Type your message…

Chat ⚡ by Drift

```
#############################################################################
##
# Example 3 - Remote SQL with SA Authentication
#############################################################################
##
#############################################
# Testing query against Remote SQL default MSSQLSERVER instance using sa
login
#############################################
# Running the SQL Query, setting result of query to $False if any errors
caaught
Try
{
# Setting to null first to prevent seeing previous query results
$SQLDBSizeResult = $null
# Running the query
$SQLDBSizeResult = Invoke-SqlCmd -Query $SQLDBSizeQuery -ServerInstance
$RemoteDefaultSQLInstance -Username $RemoteDefaultSQLInstanceUser -Password
$RemoteDefaultSQLInstancePassword
# Setting the query result
$SQLQuerySuccess = $TRUE
}
Catch
{
# Overwriting result if it failed
$SQLQuerySuccess = $FALSE
}
# Output of the results for you to see
"SQLInstance: $RemoteSQLInstance"
"SQLQueryResult: $SQLQuerySuccess"
"SQLQueryOutput:"
$SQLDBSizeResult
#############################################
# End of script
#############################################
```

I hope you found this useful. If so, twe
on Drift. Happy scripting,

Joshua

Share this:

Twitter Facebook LinkedIn

Joshua Stenhouse

Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.

Type your message…

Chat ⚡ by Drift

Like this:

Like 

3 bloggers like this.

Related

Working with SQL databases using PowerShell
July 10, 2017
In "Microsoft SQL"

Using PowerShell to report on thousands of Microsoft SQL Instances & Databases
June 14, 2018
In "Microsoft SQL"

Importing CSV files into a Microsoft SQL DB using PowerShell
June 6, 2018
In "Microsoft SQL"

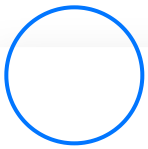
Published in **Microsoft SQL** and **Scripting Basics**

invoke-sqlcmd Microsoft PowerShell SQL

Working with SQL databases using PowerShell – Virtually Sober

Reply

[…] Connecting to Microsoft SQL Databases using PowerShell Invoke-Sqlcmd […]



Joshua Stenhouse

Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.

Type your message…

Chat ⚡ by Drift

Leave a Reply

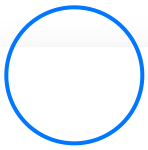
Enter your comment here...

This site uses Akismet to reduce spam. Learn how your comment data is processed.

Virtually Sober

If there is free booze and Virtualization; I'm there!

Startup Blog by Compete Themes.



Joshua Stenhouse

Hey. Thanks for visiting my blog. I'm online so if you have any questions, or need help with a script, give me a shout.

Type your message…

Chat ⚡ by Drift