DAT220x

# Delivering a Data Warehouse in the Cloud

Lab 04 | Managing Data Warehouses

## Overview

In this lab, you will manage your Azure SQL Data Warehouse. You will adjust compute, enable security features, and analyze simulated workloads.

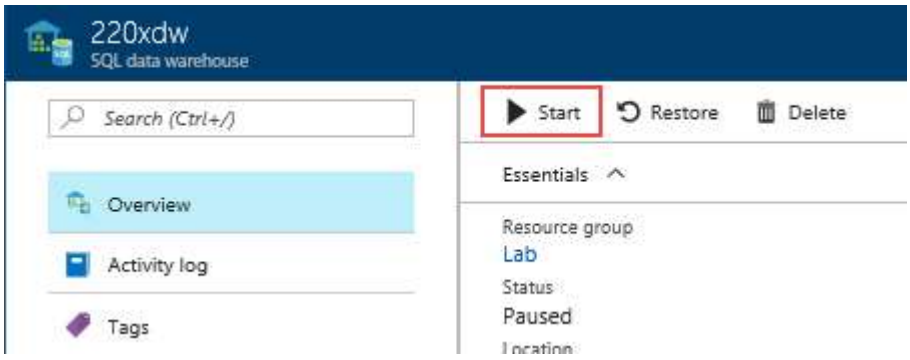
Note: The four labs in this course are cumulative. You cannot complete the following labs if this lab has not been successfully completed.

## Exercise 1: Resume the Lab Environment

In this exercise, you will resume your SQL DW from its paused state, and restart your lab virtual machine.

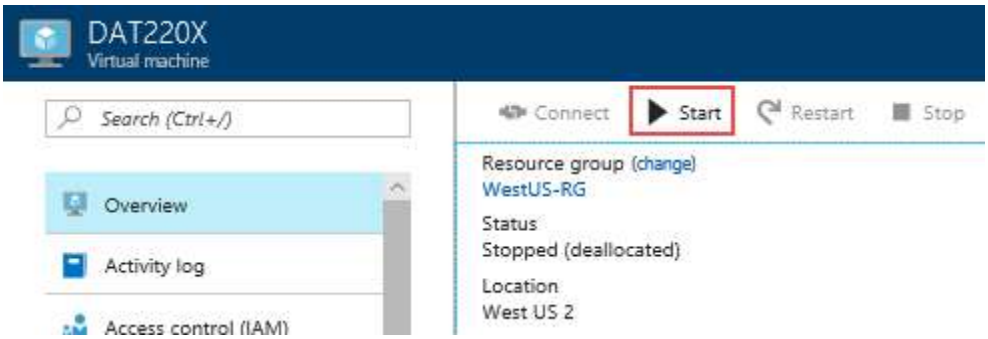
1. Sign in to the Azure Portal by using your subscription. Navigate to the Lab resource group you created in the last lab, then click to open the blade for your SQL DW.

2. If necessary, resume the SQL DW from its paused state:




3. Watch for the notification of the successful resume request.

4. Return to the Lab resource group blade, and click to open the virtual machine you created in your last lab. If necessary, start the VM




5. Once the Connect link becomes available, click Connect and open the Remote Desktop file, as you learned in Lab 1. (You may need to refresh the browser tab.)

6. Using the credentials you created in Lab 1, log in to the VM. Remember to use "use another account" when logging in.

7. Back in the Azure portal, navigate to the SQL DW created in Lab 1.

8. In the Essentials pane, click the Start button to resume your SQL Data Warehouse.

# Exercise 2: Manage Compute

In this exercise, you will adjust your DW compute using Azure PowerShell.

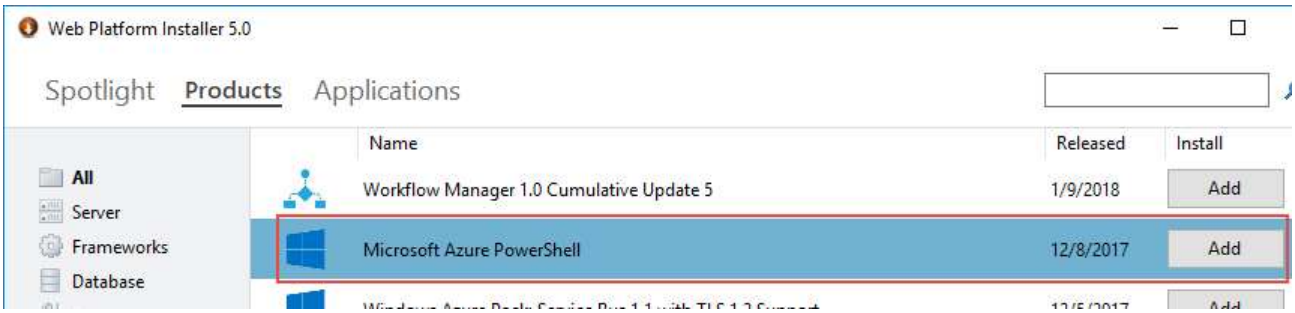
1. In the VM, open up the Edge Browser and navigate to https://www.microsoft.com/web/downloads/platform.aspx.

2. Click the Free Download button for the Web Platform Installer.


3. Run the wpilauncher.exe file to install the Web Platform Installer.

4. When the install finishes, the Web Platform Installer will automatically run.

5. In the Web Platform Installer, select the Products tab, and click the Add button for Microsoft Azure PowerShell. If needed, type Azure PowerShell in the search box to bring the Azure PowerShell option to the top of the list.




6. Click Install.

7. Click I Accept on the Microsoft License Terms dialog.

8. Once Azure PowerShell has finished installing, click Finish.

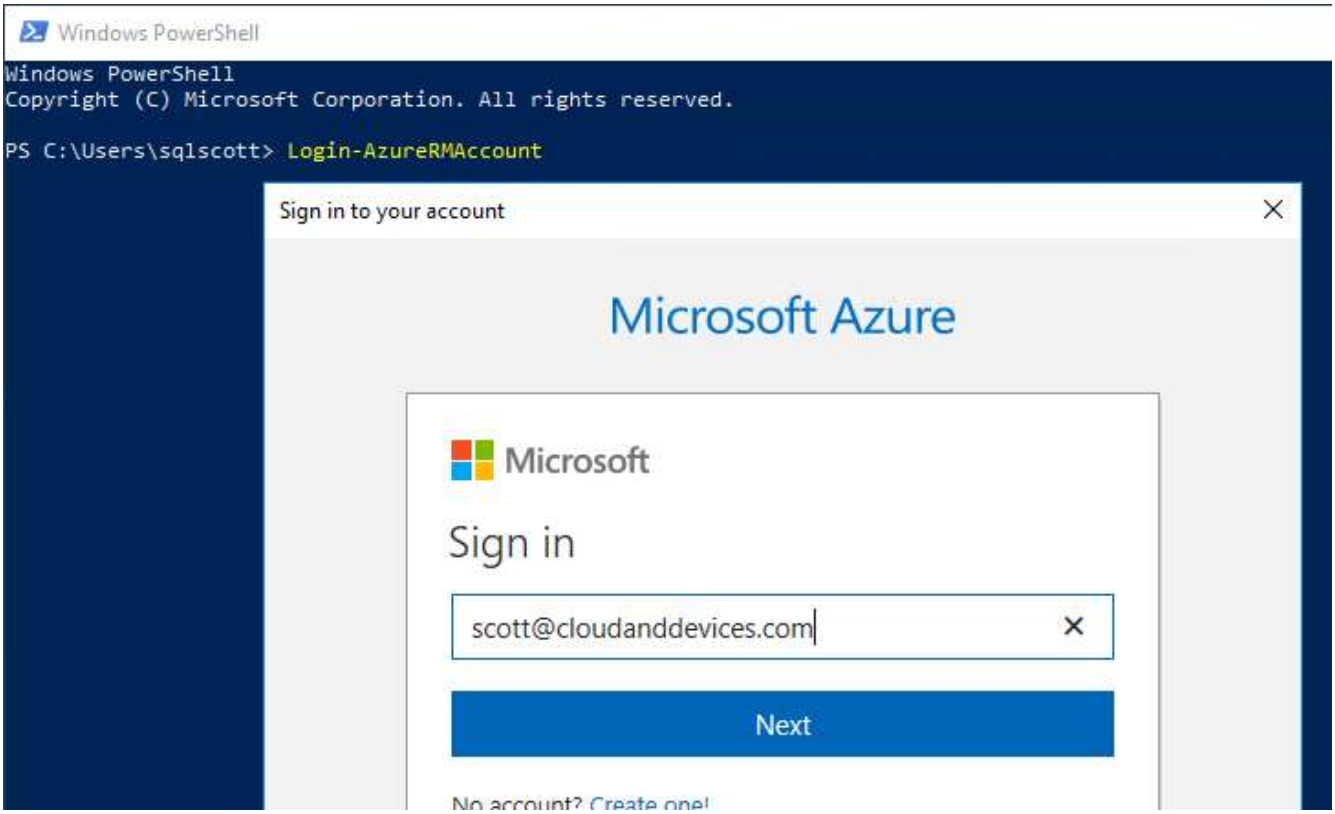
9. Click Exit on the Web Platform Installer.

10. In the VM, type PowerShell in the search box next to the Start button.


11. Select Windows PowerShell from the list of results.

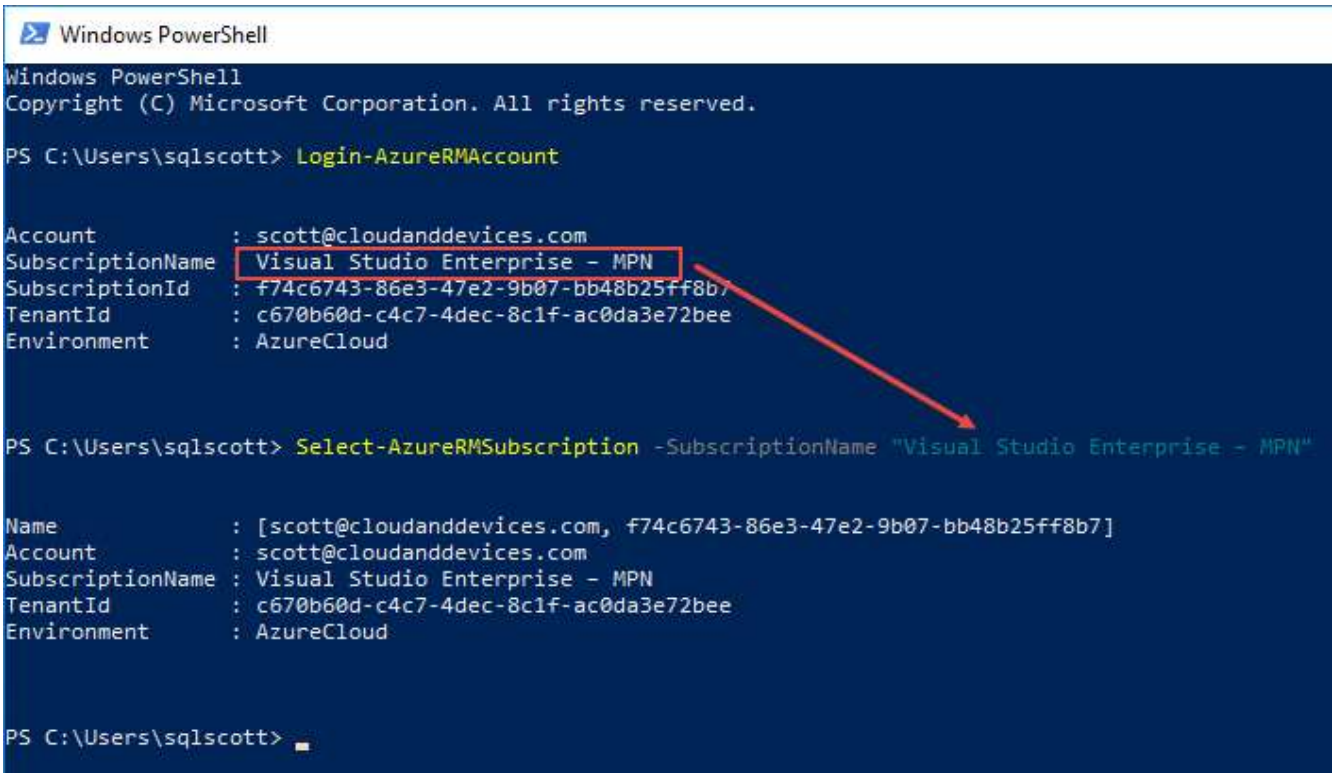
12. In the Windows PowerShell window, type Login-AzureRMAccount, and press Enter.

13. Sign in to your Azure account.



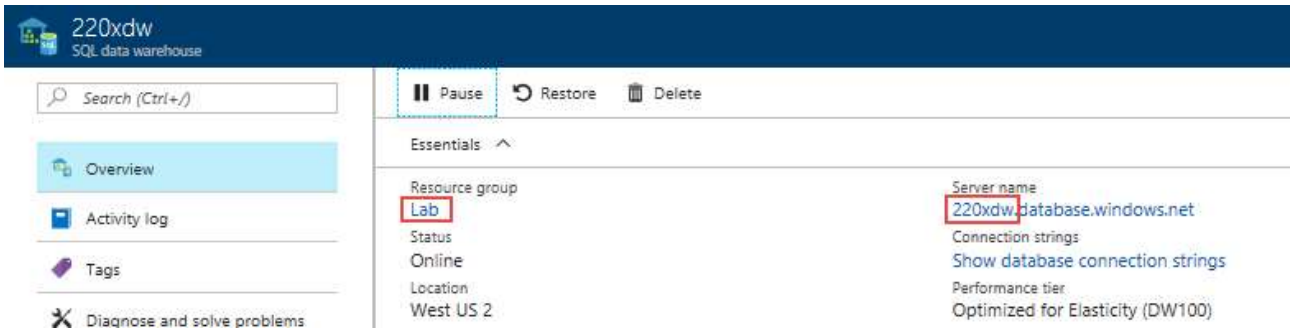

14. Once authenticated, type Select-AzureRMSubscription -SubscriptionName "<>". DO NOT PRESS ENTER YET. Replace the <> with your subscription name. You can get this from the information returned from logging in in the previous step.

15. Once the subscription name is entered, press the Enter key.



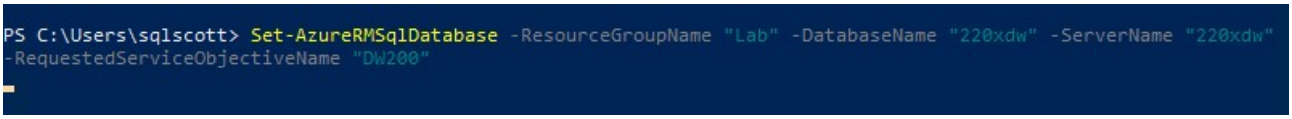

16. In the Azure portal, click on the Data Warehouse you created.

17. In the Essentials pane, make a note of the Resource Group name, Server name, and database name.




18. Back in Windows PowerShell, type in the following, using the Resource Group, Database name, Server name, and Service object information obtained in step 17. This cmdlet changes the performance level from DW100 to DW200.

    Set-AzureRMSqlDatabase -ResourceGroupName "" -DatabaseName "" -ServerName "" -RequestedServiceObjectiveName "DW200"



19. Press Enter.

20. The process of changing compute levels may take several minutes, after which PowerShell will return the results of the compute change.

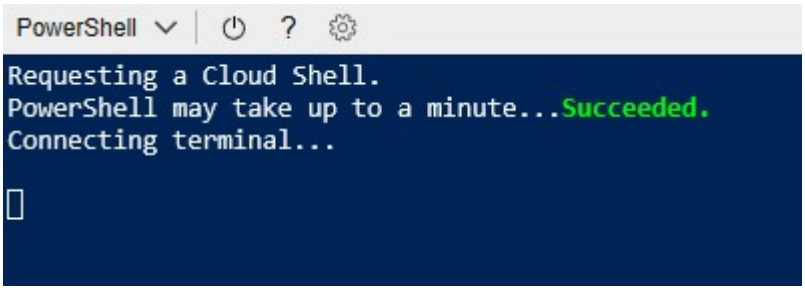
```
PS C:\Users\sqlscott> Set-AzureRMSqlDatabase -ResourceGroupName "Lab" -DatabaseName "220xdw" -ServerName "220xdw"
-RequestedServiceObjectiveName "DW200"


ResourceGroupName             : Lab
ServerName                    : 220xdw
DatabaseName                  : 220xdw
Location                      : West US 2
DatabaseId                    : d4eaeb4e-38a7-4c19-ae3d-f4fb95327fe4
Edition                       : DataWarehouse
CollationName                 : SQL_Latin1_General_CP1_CI_AS
CatalogCollation              :
MaxSizeBytes                  : 263882790666240
Status                        : Online
CreationDate                  : 1/17/2018 2:10:16 AM
CurrentServiceObjectiveId     : 99e78a92-d724-4e1b-857b-2be661f3d153
CurrentServiceObjectiveName   : DW200
RequestedServiceObjectiveId   : 99e78a92-d724-4e1b-857b-2be661f3d153
RequestedServiceObjectiveName :
ElasticPoolName               :
EarliestRestoreDate           :
Tags                          :
ResourceId                    : /subscriptions/f74c6743-86e3-47e2-9b07-bb48b25ff8b7/resourceGroups/Lab/providers/
                                Microsoft.Sql/servers/220xdw/databases/220xdw
CreateMode                    :
ReadScale                     : Disabled
ZoneRedundant                 : False


PS C:\Users\sqlscott>
```

21. Close the Windows PowerShell window.

22. In the portal, open the SQL Data Warehouse and select the Scale option in the Common Tasks group. The performance level now shows DW200.


23. On the toolbar in the Azure portal, click on the Cloud Shell button.

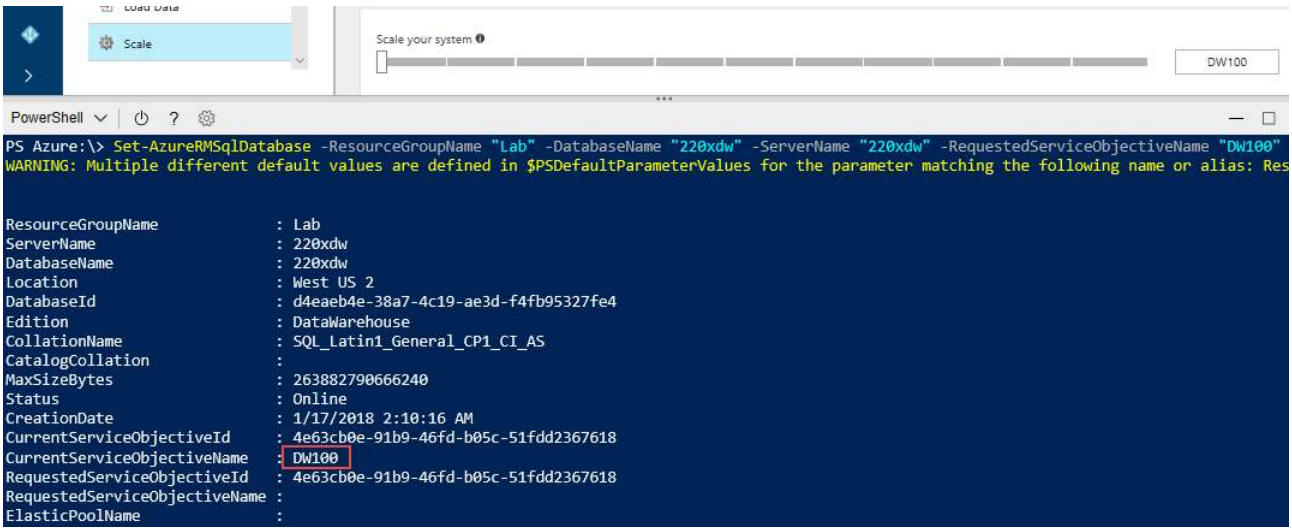
24. The Cloud Shell window will open at the bottom of the portal, creating and opening a Cloud Shell environment for both PowerShell and Bash. Ensure PowerShell is selected.




25. Once the Cloud Shell environment is created, at the Ps Azure:\> prompt, type the same Set-AzureRMSqlDatabase statement you used in step 18, **replacing the scale units with DW100**.

26. Press Enter.

27. Similar to scaling up, the process of changing compute levels and scaling back down to DW100 may take several minutes, after which the Cloud Shell will return the results of the compute change.

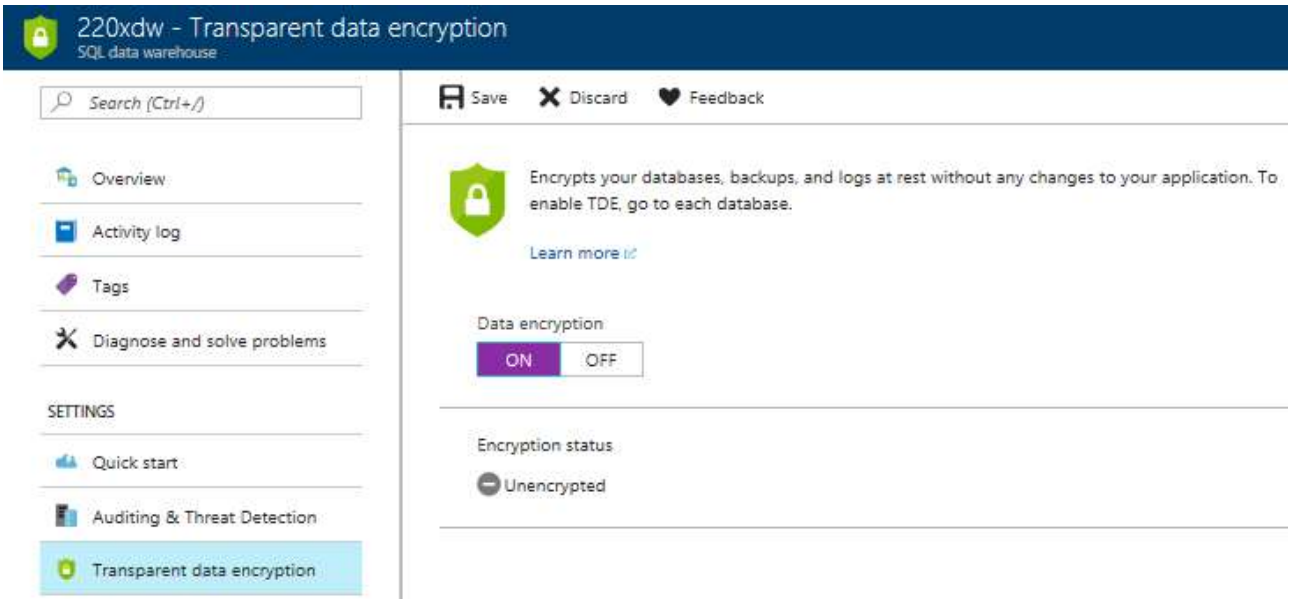



28. Close the Cloud Shell window.

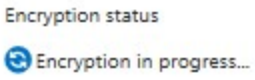
# Exercise 3: Enable SQL DW Security Features

In this exercise, you will enable SQL DW security features in the portal, including TDE, Auditing, and Threat Detection.

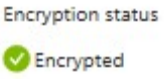
1. In the Azure portal for the SQL Data Warehouse, select the Transparent Data Encryption option in the Settings group.
2. In the Transparent Data Encryption pane, turn Data Encryption On.



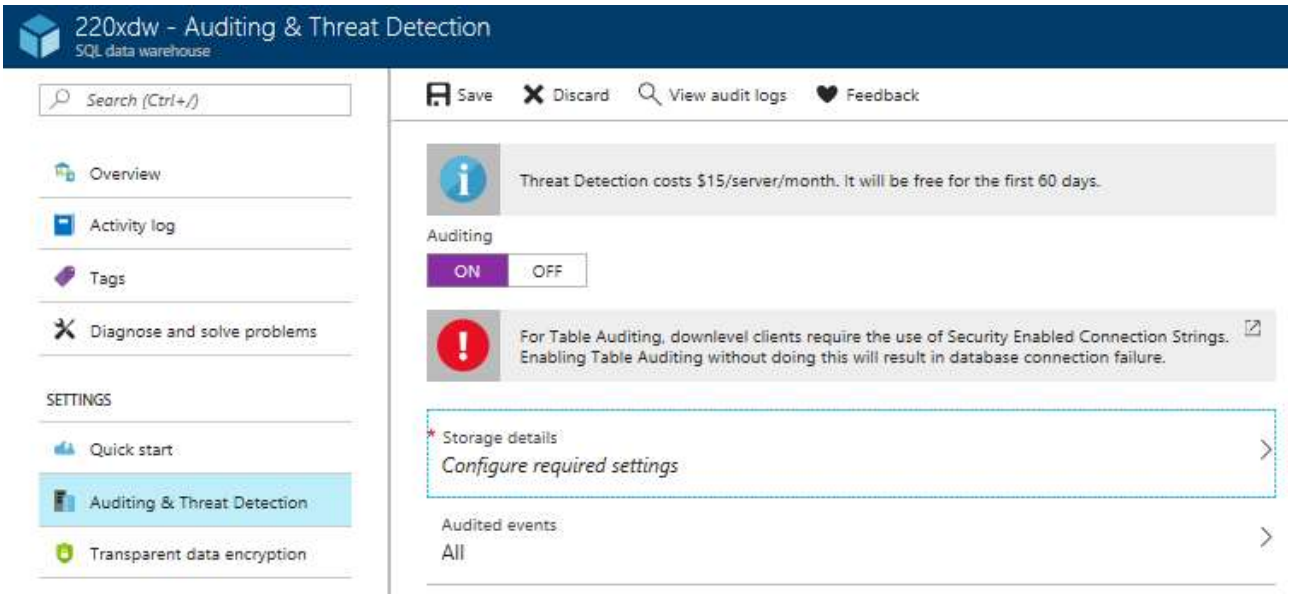

3. Click Save.
4. The encryption process will begin. For the sample data, this process takes 1-2 minutes.




5. Once the encryption process is complete, the status will change to Encrypted.




6. Select the Auditing and Threat Detection option.




7. In the Auditing and Threat Detection pane, turn Auditing On.

8. Select the Storage Details option.

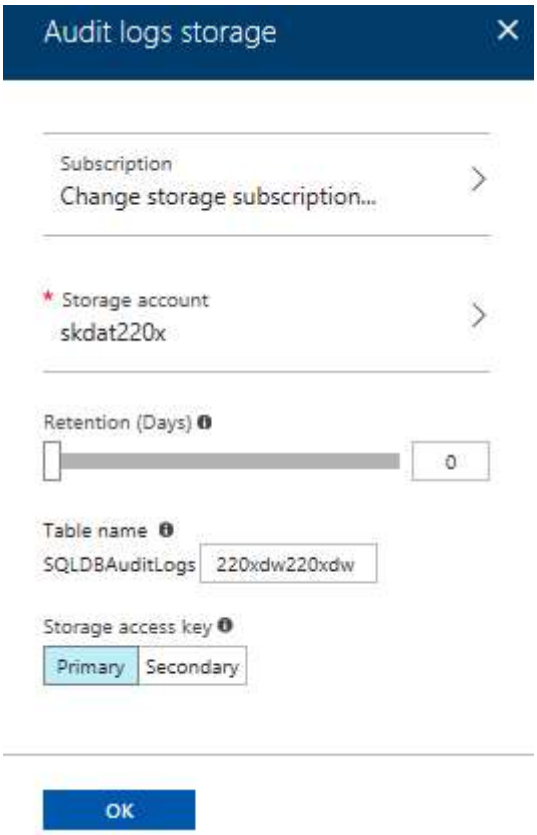
9. In the Audit logs storage pane, click the Storage account option.

10. In the Choose Storage Account pane, select the storage account created in Lab 1.

11. In the Audit logs storage pane, leave the retention days to 0. This option sets the number of days in which to retain the audit information. A value of 0 signifies an unlimited retention period.

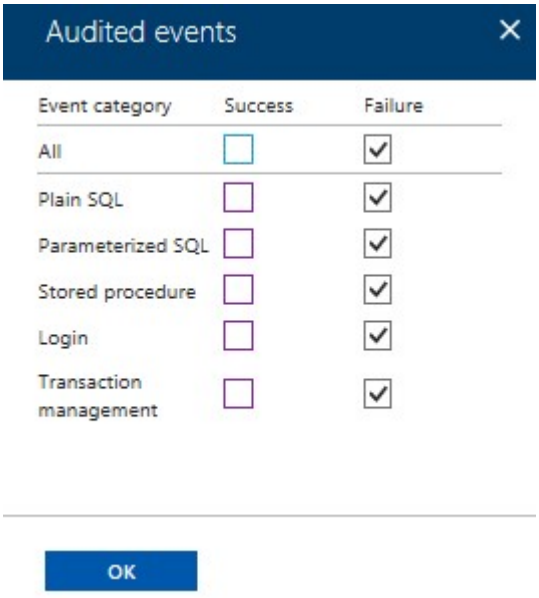
12. Leave the Table name and Storage access key default values.

13. Click OK.




14. In the Auditing and Threat Detection pane, select the Audited Events option.

15. In the Audited Events pane, ensure that all Event categories are selected for the Failure option. Uncheck all the Success options.




16. Click OK.

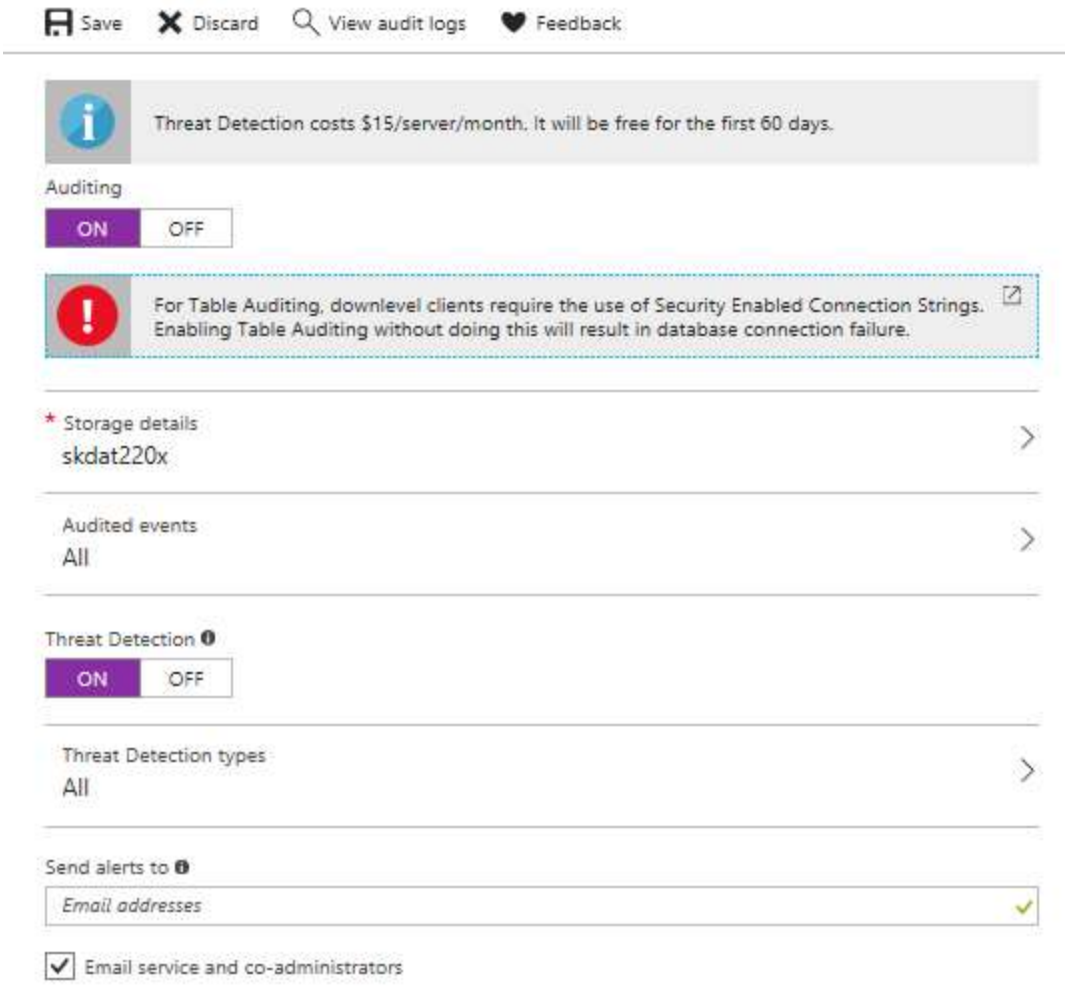
17. Turn on Threat Detection.

18. Select Threat Detection Types.

19. Ensure that all the threat detection types are selected.

20. Click OK.

21. In the Auditing and Threat Detection pane, select the Email service and co-administrators checkbox.




22. Click Save.

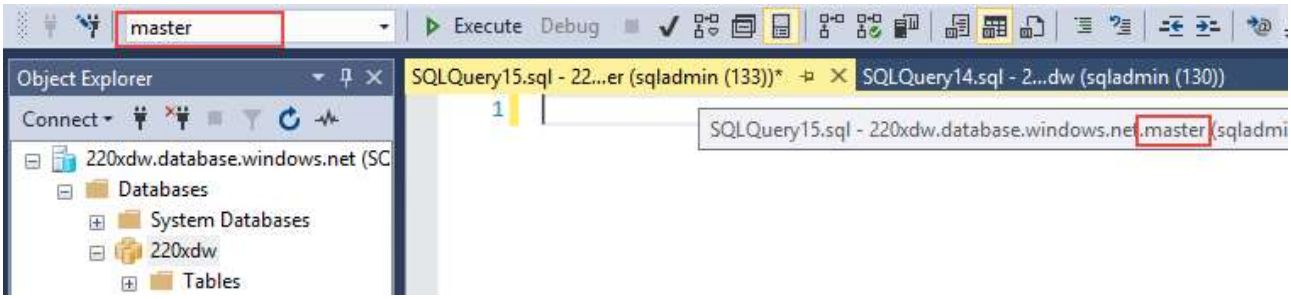
23. Azure SQL Data Warehouse is now proactively monitoring and detecting anomalous database activities and logging all findings in the specified storage account.

# Exercise 4: Working with SQL DW User Accounts and Groups

In this exercise, you will create and work with SQL DW users, groups, and permissions.

1. Use your taskbar shortcut (or other preferred method) to launch SQL Server Management Studio (SSMS).
2. In the Connect to Server dialog, enter the Server name of your SQL DW, using the information in the Portal.


3. Change the Authentication drop-down list to SQL Server Authentication.
4. In the Login and Password boxes, provide the Server admin and password you configured during provisioning of the SQL DW.
5. In the Object Explorer pane, expand the databases node and select the database created in Lab1.
6. In the Connect to Server dialog, enter the Server name of your SQL DW, using the information in the Portal.
7. On the SSMS toolbar, click the New Query button twice to create two new query windows.
8. In the first query window, change the database connection to master, and ensure that the second query window is connected to the SQL DW database.




9. In the first query window that is connected to the master database, type in the following T-SQL which creates a new SQL login named ApplicationLogin with a strong password.

```
CREATE LOGIN ApplicationLogin WITH PASSWORD = 'Awes0me_P@ssw0rd!';
```

10. Click the Execute button to run the T-SQL statement.

11. In the second query window, ensure that it is connected to the SQL DW database, and type in the following T-SQL, which creates a new user in the SQL DW database called ApplicationUser, based on the login created in step 9.

```
CREATE USER ApplicationUser FOR LOGIN ApplicationLogin;
```

12. Enter the following T-SQL into the query window, then highlight this code and click the Execute button on the toolbar. This code creates a new role called dw_admin.

```
CREATE ROLE dw_admin;
```

13. Click the Execute button on the toolbar.

14. Enter the following T-SQL into the query window, then highlight this code and click the Execute button on the toolbar. This T-SQL code grants, or gives, permissions to the dw_admin role to perform certain functions.

```
GRANT
  ALTER,
  CONTROL,
  DELETE,
  EXECUTE,
  INSERT,
  SELECT,
  UPDATE,
  TAKE OWNERSHIP,
  VIEW DEFINITION
ON SCHEMA::dbo
TO dw_admin;
```

15. Enter the following T-SQL into the query window, then highlight this code and click the Execute. This code adds the user ApplicationUser to the dw_admin role.

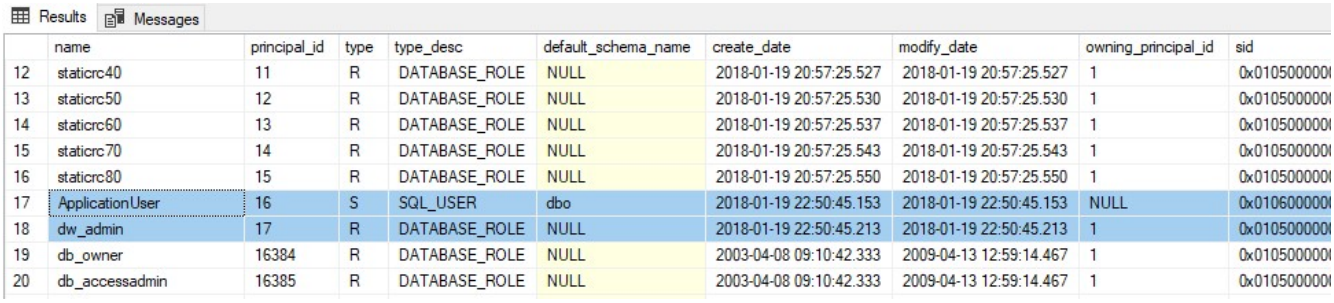
```
EXECUTE sp_addrolemember N'dw_admin', N'ApplicationUser';
```

16. Enter the following T-SQL into the query window, then highlight this code and click the

Execute. This query queries the sys.database_principals system table and returns a row for each security principal in the database.

```
SELECT * FROM sys.database_principals
```

17. In the query results you will see two rows; one for the aw_admin role and another for the ApplicationUser user.



Results | Messages

| | name | principal_id | type | type_desc | default_schema_name | create_date | modify_date | owning_principal_id | sid |
|---|---|---|---|---|---|---|---|---|---|
| 12 | staticrc40 | 11 | R | DATABASE_ROLE | NULL | 2018-01-19 20:57:25.527 | 2018-01-19 20:57:25.527 | 1 | 0x010500000( |
| 13 | staticrc50 | 12 | R | DATABASE_ROLE | NULL | 2018-01-19 20:57:25.530 | 2018-01-19 20:57:25.530 | 1 | 0x010500000( |
| 14 | staticrc60 | 13 | R | DATABASE_ROLE | NULL | 2018-01-19 20:57:25.537 | 2018-01-19 20:57:25.537 | 1 | 0x010500000( |
| 15 | staticrc70 | 14 | R | DATABASE_ROLE | NULL | 2018-01-19 20:57:25.543 | 2018-01-19 20:57:25.543 | 1 | 0x010500000( |
| 16 | staticrc80 | 15 | R | DATABASE_ROLE | NULL | 2018-01-19 20:57:25.550 | 2018-01-19 20:57:25.550 | 1 | 0x010500000( |
| 17 | ApplicationUser | 16 | S | SQL_USER | dbo | 2018-01-19 22:50:45.153 | 2018-01-19 22:50:45.153 | NULL | 0x010600000( |
| 18 | dw_admin | 17 | R | DATABASE_ROLE | NULL | 2018-01-19 22:50:45.213 | 2018-01-19 22:50:45.213 | 1 | 0x010500000( |
| 19 | db_owner | 16384 | R | DATABASE_ROLE | NULL | 2003-04-08 09:10:42.333 | 2009-04-13 12:59:14.467 | 1 | 0x010500000( |
| 20 | db_accessadmin | 16385 | R | DATABASE_ROLE | NULL | 2003-04-08 09:10:42.333 | 2009-04-13 12:59:14.467 | 1 | 0x010500000( |

18. SQL Data Warehouse includes two special roles for managing databases and logins; DBMANAGER and LOGINMANAGER. The following T-SQL adds the ApplicationUser to these two roles.

```
EXECUTE sp_addrolemember N'DBMANAGER', N'ApplicationUser';
EXECUTE sp_addrolemember N'LOGINMANAGERD', N'ApplicationUser';
```

19. Close the query window.

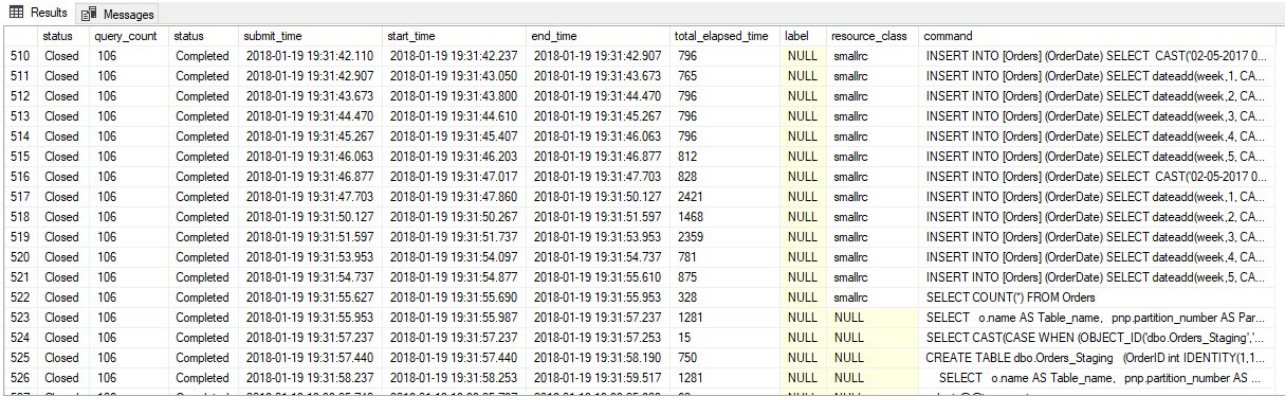
# Exercise 5: Analyzing Workloads

In this exercise, you will look at a few of the DMVs that help analyze query prioritization for your SQL DW workload.

1. On the SSMS toolbar, click the New Querybutton to create a new query window.
2. In the query window, enter the following T-SQL. This query uses two DMVs; sys.dm_pdw_exec requests and sys.dm_pdw_exec_sessions. These two DVMs provide information about all the requests currently or recently executed.

```
SELECT   es.[status],
         es.query_count,
         er.[status],
         er.submit_time,
         er.start_time,
         er.end_time,
         er.total_elapsed_time,
         er.[label],
         er.resource_class,
         er.command
FROM sys.dm_pdw_exec_requests er
JOIN sys.dm_pdw_exec_sessions es ON er.session_id = es.session_id
```

3. The query results show critical information, such as the actual T-SQL statement, the status of the query, the query count, and its start time and end time. It also show, type resource class type was used, if any.



Results | Messages

| | status | query_count | status | submit_time | start_time | end_time | total_elapsed_time | label | resource_class | command |
|---|---|---|---|---|---|---|---|---|---|---|
| 510 | Closed | 106 | Completed | 2018-01-19 19:31:42.110 | 2018-01-19 19:31:42.237 | 2018-01-19 19:31:42.907 | 796 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT CAST('02-05-2017 0... |
| 511 | Closed | 106 | Completed | 2018-01-19 19:31:42.907 | 2018-01-19 19:31:43.050 | 2018-01-19 19:31:43.673 | 765 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,1, CA... |
| 512 | Closed | 106 | Completed | 2018-01-19 19:31:43.673 | 2018-01-19 19:31:43.800 | 2018-01-19 19:31:44.470 | 796 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,2, CA... |
| 513 | Closed | 106 | Completed | 2018-01-19 19:31:44.470 | 2018-01-19 19:31:44.610 | 2018-01-19 19:31:45.267 | 796 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,3, CA... |
| 514 | Closed | 106 | Completed | 2018-01-19 19:31:45.267 | 2018-01-19 19:31:45.407 | 2018-01-19 19:31:46.063 | 796 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,4, CA... |
| 515 | Closed | 106 | Completed | 2018-01-19 19:31:46.063 | 2018-01-19 19:31:46.203 | 2018-01-19 19:31:46.877 | 812 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,5, CA... |
| 516 | Closed | 106 | Completed | 2018-01-19 19:31:46.877 | 2018-01-19 19:31:47.017 | 2018-01-19 19:31:47.703 | 828 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT CAST('02-05-2017 0... |
| 517 | Closed | 106 | Completed | 2018-01-19 19:31:47.703 | 2018-01-19 19:31:47.860 | 2018-01-19 19:31:50.127 | 2421 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,1, CA... |
| 518 | Closed | 106 | Completed | 2018-01-19 19:31:50.127 | 2018-01-19 19:31:50.267 | 2018-01-19 19:31:51.597 | 1468 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,2, CA... |
| 519 | Closed | 106 | Completed | 2018-01-19 19:31:51.597 | 2018-01-19 19:31:51.737 | 2018-01-19 19:31:53.953 | 2359 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,3, CA... |
| 520 | Closed | 106 | Completed | 2018-01-19 19:31:53.953 | 2018-01-19 19:31:54.097 | 2018-01-19 19:31:54.737 | 781 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,4, CA... |
| 521 | Closed | 106 | Completed | 2018-01-19 19:31:54.737 | 2018-01-19 19:31:54.877 | 2018-01-19 19:31:55.610 | 875 | NULL | smallrc | INSERT INTO [Orders] (OrderDate) SELECT dateadd(week,5, CA... |
| 522 | Closed | 106 | Completed | 2018-01-19 19:31:55.627 | 2018-01-19 19:31:55.690 | 2018-01-19 19:31:55.953 | 328 | NULL | smallrc | SELECT COUNT(*) FROM Orders |
| 523 | Closed | 106 | Completed | 2018-01-19 19:31:55.953 | 2018-01-19 19:31:55.987 | 2018-01-19 19:31:57.237 | 1281 | NULL | NULL | SELECT o.name AS Table_name, pnp.partition_number AS Par... |
| 524 | Closed | 106 | Completed | 2018-01-19 19:31:57.237 | 2018-01-19 19:31:57.237 | 2018-01-19 19:31:57.253 | 15 | NULL | NULL | SELECT CAST(CASE WHEN (OBJECT_ID('dbo.Orders_Staging','... |
| 525 | Closed | 106 | Completed | 2018-01-19 19:31:57.440 | 2018-01-19 19:31:57.440 | 2018-01-19 19:31:58.190 | 750 | NULL | NULL | CREATE TABLE dbo.Orders_Staging (OrderID int IDENTITY(1,1... |
| 526 | Closed | 106 | Completed | 2018-01-19 19:31:58.237 | 2018-01-19 19:31:58.253 | 2018-01-19 19:31:59.517 | 1281 | NULL | NULL | SELECT o.name AS Table_name, pnp.partition_number AS ... |

4. The query can be modified to include the sys.dm_pdw_waits DMV, which we can then use to see which resources current or recently executed queries were waiting for.

```
SELECT   w.[type],
         w.[object_type],
         w.[object_name],
         w.[state],
         es.[status],
         es.query_count,
         er.[status],
         er.submit_time,
         er.start_time,
         er.end_time,
         er.total_elapsed_time,
         er.[label],
         er.resource_class,
         er.command
FROM sys.dm_pdw_exec_requests er
JOIN sys.dm_pdw_exec_sessions es ON er.session_id = es.session_id
JOIN sys.dm_pdw_waits w ON es.session_id = w.session_id
```

5. By including a few columns in the query from the sys.dm_pdw_waits DMV, you can determine the wait type, the object that is affected by the wait, and the state of the wait.

Results | Messages

| | type | object_type | object_name | state | status | query_count | status | su |
|---|---|---|---|---|---|---|---|---|
| 77 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Completed | 20 |
| 78 | LocalQueriesConcurrencyResourceType | SYSTEM | 7c672a48-1e47-4e4f-8025-0792fd1f3e1b | Granted | Active | 46 | Completed | 20 |
| 79 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Completed | 20 |
| 80 | LocalQueriesConcurrencyResourceType | SYSTEM | 7c672a48-1e47-4e4f-8025-0792fd1f3e1b | Granted | Active | 46 | Completed | 20 |
| 81 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Completed | 20 |
| 82 | LocalQueriesConcurrencyResourceType | SYSTEM | 7c672a48-1e47-4e4f-8025-0792fd1f3e1b | Granted | Active | 46 | Completed | 20 |
| 83 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Completed | 20 |
| 84 | LocalQueriesConcurrencyResourceType | SYSTEM | 7c672a48-1e47-4e4f-8025-0792fd1f3e1b | Granted | Active | 46 | Completed | 20 |
| 85 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Completed | 20 |
| 86 | LocalQueriesConcurrencyResourceType | SYSTEM | 7c672a48-1e47-4e4f-8025-0792fd1f3e1b | Granted | Active | 46 | Completed | 20 |
| 87 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Failed | 20 |
| 88 | LocalQueriesConcurrencyResourceType | SYSTEM | 7c672a48-1e47-4e4f-8025-0792fd1f3e1b | Granted | Active | 46 | Failed | 20 |
| 89 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Completed | 20 |
| 90 | LocalQueriesConcurrencyResourceType | SYSTEM | 7c672a48-1e47-4e4f-8025-0792fd1f3e1b | Granted | Active | 46 | Completed | 20 |
| 91 | Shared | DATABASE | MyTestDW | Granted | Active | 46 | Running | 20 |

6. LocalQueriesConcurrencyResourceType wait type signifies that the query is sitting outside of the concurrency slot framework. UserConcurrencyResourceType wait type means that queries are inside the concurrency slot and are awaiting execution within the workload group.

7. The sys.dm_pdw_wait_status DMV can be used at a historical trend of waits in the DW Database.

```
SELECT * FROM sys.dm_pdw_wait_stats
```

8. This DMV returns information related to the SQL Server OS state regarding instances running on the different nodes.

| | pdw_node_id | wait_name | max_wait_time | request_count | signal_time | completed_count | wait_time |
|---|---|---|---|---|---|---|---|
| 134 | 66 | LOG_RATE_GOVERNOR | 0 | 0 | 0 | 0 | 0 |
| 135 | 66 | POOL_LOG_RATE_GOVERNOR | 0 | 0 | 0 | 0 | 0 |
| 136 | 66 | SLEEP_MEMORYPOOL_ALLOCATEP... | 0 | 0 | 0 | 0 | 0 |
| 137 | 66 | SLEEP_WORKSPACE_ALLOCATEPAGE | 0 | 0 | 0 | 0 | 0 |
| 138 | 66 | SLEEP_RETRY_VIRTUALALLOC | 0 | 0 | 0 | 0 | 0 |
| 139 | 66 | MEMORY_ALLOCATION_EXT | 9 | 95684 | 0 | 95684 | 290 |
| 140 | 66 | RESERVED_MEMORY_ALLOCATION... | 0 | 0 | 0 | 0 | 0 |
| 141 | 66 | IO_QUEUE_LIMIT | 0 | 0 | 0 | 0 | 0 |
| 142 | 66 | SESSION_WAIT_STATS_CHILDREN | 0 | 0 | 0 | 0 | 0 |
| 143 | 66 | PREEMPTIVE_XE_DISPATCHER | 0 | 1 | 0 | 0 | 0 |
| 144 | 66 | XE_BUFFERMGR_ALLPROCESSED_E... | 8 | 3 | 0 | 3 | 20 |
| 145 | 66 | XE_BUFFERMGR_FREEBUF_EVENT | 1 | 2 | 0 | 2 | 2 |
| 146 | 66 | PREEMPTIVE_XE_CALLBACKEXECUTE | 22 | 213 | 0 | 213 | 25 |
| 147 | 66 | XE_SESSION_CREATE_SYNC | 0 | 0 | 0 | 0 | 0 |
| 148 | 66 | PREEMPTIVE_XE_GETTARGETSTATE | 0 | 0 | 0 | 0 | 0 |

9. More information regarding the different types of waits, visit: https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-os-wait-stats-transact-sql

You have now completed the lab. Be sure to complete the Finishing Up exercise to shut down and stop the VM, and to pause the SQL DW.

# Finishing Up

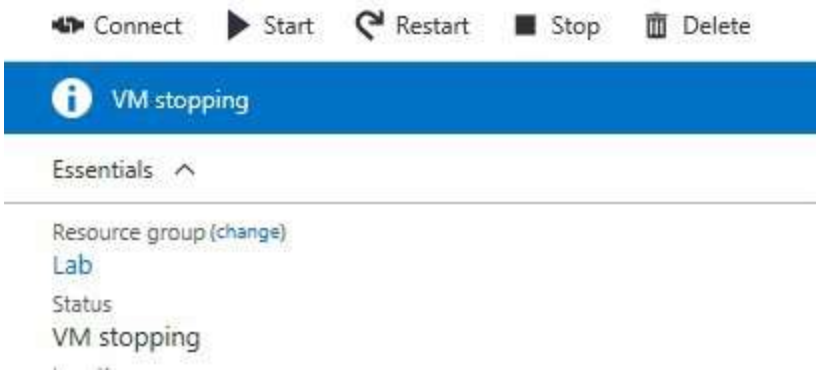
In this task, you will shut down and stop the VM, and pause the SQL DW. If you are immediately continuing to further labs in this course, you can skip this task. However, costs will continue to be incurred by the VM and SQL DW until they are deallocated or paused using the steps below.

1. Close all open applications.

2. Right-click on the Windows button in the bottom left corner of your screen. Click on Shut down or sign out. Be sure this is the menu for you VM and not your desktop!

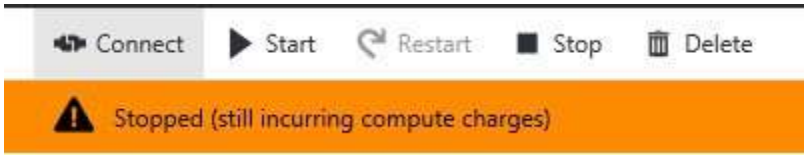

3. Click shut down.

4. In the Azure Portal Web browser page on your desktop, wait until the status of the VM updates.

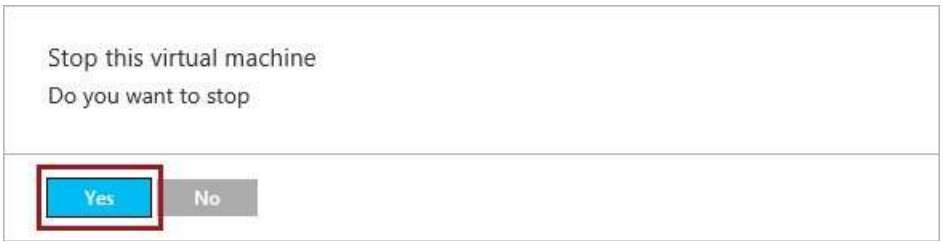



In this state, however, the VM is still billable.

5. To deallocate the VM, click Stop.




6. When prompted to stop the VM, click Yes.




The deallocation can take several minutes to complete.

7. Verify that the VM status updates to Stopped (Deallocated).


In this state, the VM is now not billable.

Note that a deallocated VM will likely acquire a different IP address the next time it is started.

8. Browse the Azure Portal to open the blade for your SQL DW.

9. Click the Pause button. Watch the status change, and confirm that the SQL DW has paused before you finish the lab.

10. Sign out of the Azure Portal.