

# Contents

<b>1 DP-300: Administering Relational Databases on Microsoft Azure</b>	<b>3</b>
1.1 Lab 1 - Using the Azure Portal and SQL Server Management Studio . . . . .	3
1.2 Lab 2 – Deploying PaaS Databases . . . . .	3
1.3 Lab 3 – Implement a Secure Environment . . . . .	3
1.4 Lab 4 – Monitor and Optimize Resources . . . . .	3
1.5 Lab 5 -Query Performance Troubleshooting . . . . .	4
1.6 Lab 6 – Automate Resources . . . . .	4
1.7 Lab 7 – Planning and Implementing a High Availability and Disaster Recovery Environment . . . . .	4
1.8 Notes . . . . .	4
1.9 title: Online Hosted Instructions permalink: index.html layout: home . . . . .	4
<b>2 Content Directory</b>	<b>4</b>
2.1 Labs . . . . .	4
2.2 lab: title: 'Lab 1 - Using the Azure Portal and SQL Server Management Studio' module: 'The Role of the Azure Database Administrator' . . . . .	4
<b>3 Lab 1 - Using the Azure Portal and SQL Server Management Studio</b>	<b>4</b>
<b>4 Lab Overview</b>	<b>5</b>
<b>5 Lab Goal</b>	<b>5</b>
<b>6 Lab Objectives</b>	<b>5</b>
<b>7 Scenario</b>	<b>5</b>
<b>8 Exercise 1: Provision a SQL Server on an Azure Virtual Machine</b>	<b>5</b>
8.1 Task 1: Explore the Azure Portal . . . . .	5
8.2 Task 2: Provision a SQL Server on an Azure Virtual Machine . . . . .	9
<b>9 Exercise 2: Connect to SQL Server and Restore a Backup</b>	<b>16</b>
9.1 Task 1: Connect to the Virtual Machine and download the backup file . . . . .	16
9.2 Task 2: Restore the Database and Verify its Availability . . . . .	23
9.3 Task 3: Clean Up . . . . .	32
9.4 lab: title: 'Lab 2 – Deploying PaaS databases' module: 'Plan and Implement Data Platform Resources' . . . . .	32
<b>10 Lab 2 – Deploying PaaS databases</b>	<b>32</b>
<b>11 Lab Overview</b>	<b>32</b>
<b>12 Lab Objectives</b>	<b>32</b>
<b>13 Scenario</b>	<b>33</b>
<b>14 Exercise 1: Configure Basic Resources</b>	<b>33</b>
14.1 Task 1: Create a Resource Group . . . . .	33
14.2 Task 2: Create a Virtual Network . . . . .	34
<b>15 Exercise 2: Deploy an Azure SQL Database</b>	<b>36</b>
15.1 Task 1: Deploy an Azure SQL Database . . . . .	36
15.2 Task 2: Enable All Azure Services access to new SQL Server . . . . .	41
<b>16 Exercise 3: Connect to Azure SQL Database</b>	<b>42</b>
16.1 Task 1: Register Azure SQL Database Instance in Azure Data Studio . . . . .	42
16.2 Task 2: Query Azure SQL Database with a SQL Notebook . . . . .	46
<b>17 Exercise 4: Deploy an Azure Database for PostgreSQL Database</b>	<b>48</b>
17.1 Task 1: Deploy a PostgreSQL Database . . . . .	48
17.2 Task 2: Enable All Azure Services Access to PostgreSQL Database . . . . .	52
17.3 Task 3: Connect to the PostgreSQL Database with Azure Data Studio . . . . .	52

17.4 lab: title: 'Lab 3 – Implement a Secure Environment' module: 'Implement a Secure Environment'	54
<b>18 Lab 3 – Implement a Secure Environment</b>	<b>54</b>
<b>19 Lab overview</b>	<b>54</b>
<b>20 Lab objectives</b>	<b>54</b>
<b>21 Scenario</b>	<b>54</b>
21.1 Exercise 1: Configure an Azure SQL Database Firewall and connect to a new database . . . . .	54
21.2 Exercise 2: Authorize Access to Azure SQL Database with Azure Active Directory . . . . .	66
21.3 Exercise 3: Enable Azure Defender for SQL and Data Classification . . . . .	68
21.4 Exercise 4: Manage access to database objects . . . . .	74
21.5 <b>Do not remove any of the resources created in this lab as they will be used in subsequent lab exercises.</b> . . . . .	76
21.6 lab: title: 'Lab 4 – Monitor and Optimize Resources' module: 'Monitor and Optimize Operational Resources'	76
<b>22 Lab 4 – Monitor and Optimize Resources</b>	<b>76</b>
<b>23 Lab overview</b>	<b>76</b>
<b>24 Lab objectives</b>	<b>76</b>
<b>25 Scenario</b>	<b>76</b>
<b>26 Exercise 1: Isolate problems by monitoring</b>	<b>77</b>
26.1 Task 1: Review CPU Utilization in Azure Portal . . . . .	77
26.2 Task 2: Identify High CPU Consuming Queries . . . . .	78
<b>27 Exercise 2: Identify and Resolve Blocking Issues</b>	<b>80</b>
27.1 Task 1: Run blocked queries report . . . . .	80
27.2 Task 2: Enabling Read Committed Snapshot Isolation . . . . .	84
<b>28 Exercise 3: Detect/Correct Fragmentation Issues</b>	<b>85</b>
28.1 Task 1: Identify fragmentation . . . . .	85
28.2 Task 2: Rebuild indexes . . . . .	87
28.3 lab: title: 'Lab 5 -Query Performance Troubleshooting' module: 'Optimize Query Performance'	87
<b>29 Lab 5 -Query Performance Troubleshooting</b>	<b>87</b>
<b>30 Lab overview</b>	<b>87</b>
<b>31 Lab objectives</b>	<b>87</b>
<b>32 Scenario</b>	<b>88</b>
<b>33 Exercise 1: Identify issues with database design in AdventureWorks2017</b>	<b>88</b>
33.1 Task 1: Examine the query and identify the problem. . . . .	88
33.2 Task 2: Identify two ways to fix the warning issue . . . . .	90
<b>34 Exercise 2: Isolate problem areas in poorly performing queries in AdventureWorks2017</b>	<b>92</b>
34.1 Task 1: Run a query to generate the actual execution plan . . . . .	92
34.2 Task 2: Resolve a Performance Problem from an Execution Plan . . . . .	93
<b>35 Exercise 3: Use Query Store to detect and handle regression in AdventureWorks2017.</b>	<b>94</b>
35.1 Task 1: Run a workload to generate query stats for Query Store . . . . .	95
35.2 Task 2: Examine Top Resource Consuming Queries to identify poor performance . . . . .	96
35.3 Task 3: Force a better execution plan . . . . .	99
<b>36 Exercise 4: Use query hints to impact performance in AdventureWorks2017</b>	<b>100</b>
36.1 Task 1: Run a workload . . . . .	100
36.2 Task 2: Change the query to use a parameterized stored procedure and use a Query Hint . . . . .	101
36.3 lab: title: 'Lab 6 – Automate Resources' module: 'Automation of Tasks'	103

<b>37 Lab 6 – Automate Resources</b>	<b>103</b>
<b>38 Lab overview</b>	<b>104</b>
<b>39 Lab objectives</b>	<b>104</b>
<b>40 Scenario</b>	<b>104</b>
<b>41 Exercise 1: Deploy an Azure Quickstart Template</b>	<b>104</b>
41.1 Task: Deploy an Azure SQL Database from a template . . . . .	104
<b>42 Exercise 2: Configure Performance Metrics Based Alerts</b>	<b>107</b>
42.1 Task: Creating an alert when CPU exceeds an average of 80 percent. . . . .	107
<b>43 Exercise 3: Deploy an Automation Runbook</b>	<b>113</b>
43.1 Task: Deploy an Automation Runbook to rebuild indexes in an Azure SQL Database. . . . .	113
43.2 lab: title: 'Lab 7 – Planning and Implementing a High Availability and Disaster Recovery Environment' module: 'Planning and Implementing a High Availability and Disaster Recovery Environment' . . . . .	128
<b>44 Lab 7 – Planning and Implementing a High Availability and Disaster Recovery Environment</b>	<b>128</b>
<b>45 Lab overview</b>	<b>128</b>
<b>46 Lab objectives</b>	<b>128</b>
<b>47 Scenario</b>	<b>128</b>
<b>48 Exercise 1: Enable Geo-Replication for Azure SQL Database</b>	<b>128</b>
<b>49 Exercise 2: Backup to URL and Restore from URL</b>	<b>134</b>
49.1 Task 1: Configure Backup to URL . . . . .	135
49.2 Task 2: Back Up WideWorldImporters . . . . .	137
49.3 Task 3: Restore WideWorldImporters . . . . .	139

## 1 DP-300: Administering Relational Databases on Microsoft Azure

This repository contains the instructions, script files and images for the lab exercises for the Instruction Led Training for Microsoft DP-300.

There are seven labs.

### 1.1 Lab 1 - Using the Azure Portal and SQL Server Management Studio

Students will explore the Azure Portal and use it to create an Azure VM with SQL Server 2019 installed. Then they will connect to the virtual machine through Remote Desktop Protocol and restore a database using SQL Server Management Studio.

### 1.2 Lab 2 – Deploying PaaS Databases

The student will configure basic resources needed to deploy an Azure SQL Database with a Virtual Network Endpoint. Connectivity to the SQL Database will be validated using Azure Data Studio from the lab VM. Finally, an Azure Database for PostgreSQL will be created.

### 1.3 Lab 3 – Implement a Secure Environment

The students will take the information gained in the lessons to configure and subsequently implement security in the Azure Portal and within the AdventureWorks database.

### 1.4 Lab 4 – Monitor and Optimize Resources

The students will take the information gained in the lessons to scope out the deliverables for a digital transformation project within AdventureWorks. Examining the Azure portal as well as other tools, students will

determine how to utilize native tools to identify and resolve performance related issues. Finally, students will be able to identify fragmentation within the database as well as learn steps to resolve the issue appropriately.

## 1.5 Lab 5 -Query Performance Troubleshooting

The students will evaluate a database design for problems with normalization, data type selection and index design. They will run queries with suboptimal performance, examine the query plans, and attempt to make improvements within the AdventureWorks2017 database.

## 1.6 Lab 6 – Automate Resources

The students will take the information gained in the lessons to configure and subsequently implement automate processes within AdventureWorks.

## 1.7 Lab 7 – Planning and Implementing a High Availability and Disaster Recovery Environment

The students will execute two main tasks: make Azure SQL Database geo-redundant, and backup to and restore from a URL which uses Azure.

## 1.8 Notes

You can download a .zip file containing PDFs of all the lab instructions including images. The file is in the Instructions folder.

It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.

When copying code from the lab exercises into the lab Virtual Machine, occasionally line breaks will not come across properly. Please verify that the code is correctly copied before executing it.

---

## 1.9 title: Online Hosted Instructions permalink: index.html layout: home

## 2 Content Directory

Hyperlinks to each of the lab exercises and demos are listed below.

### 2.1 Labs

```
{% assign labs = site.pages | where_exp:"page", "page.url contains '/Instructions/Labs'" %} | Module | Lab || --- | --- | {% for activity in labs %}| {{ activity.lab.module }} | [{{ activity.lab.title }}]{% if activity.lab.type %} - {{ activity.lab.type }}{% endif %}](/home/l1/Azure_clone/Azure_new/DP-300T00-Administering-Relational-Databases-on-Azure/{{ site.github.url }}{{ activity.url }}) | {% endfor %}
```

---

### 2.2 lab: title: 'Lab 1 - Using the Azure Portal and SQL Server Management Studio' module: 'The Role of the Azure Database Administrator'

## 3 Lab 1 - Using the Azure Portal and SQL Server Management Studio

**Estimated Time:** 30 minutes

**Pre-requisites:** None

**Lab files:** None

## **4 Lab Overview**

Students will explore the Azure Portal and use it to create an Azure VM with SQL Server 2019 installed. Then they will connect to the virtual machine through Remote Desktop Protocol and restore a database using SQL Server Management Studio.

## **5 Lab Goal**

The student will be able to interact with the Azure portal to create a SQL Server Virtual Machine. They should then be able to connect to the virtual machine and use SQL Server Management Studio to restore a database.

## **6 Lab Objectives**

The students will:

- Interact with the Azure portal
- Create a SQL Server Virtual Machine
- Use SQL Server Management Studio to restore a database

## **7 Scenario**

You are a database administrator for AdventureWorks. You need to create a test environment for use in a proof of concept. The proof of concept will use SQL Server on an Azure Virtual Machine and a backup of the AdventureWorksDW database. You need to set up the Virtual Machine, restore the database, and query it to ensure it is available.

## **8 Exercise 1: Provision a SQL Server on an Azure Virtual Machine**

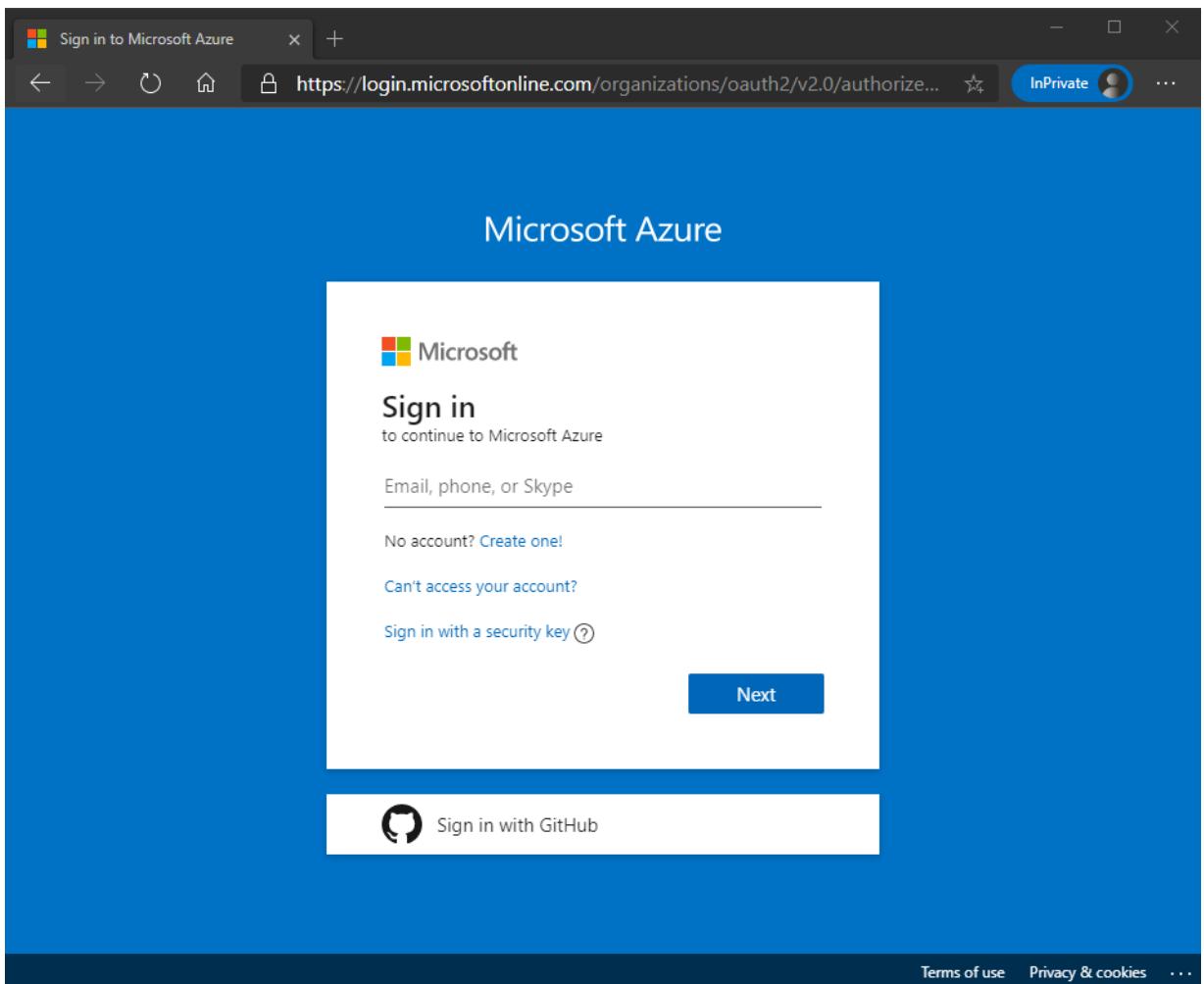
**Estimated Time: 20 minutes**

The main tasks for this exercise are as follows:

1. Explore the Azure Portal and locate important functionality.
2. Create a SQL Server on an Azure Virtual Machine using the Azure Portal.

### **8.1 Task 1: Explore the Azure Portal**

1. From the lab virtual machine, if you are not already in the portal, start a browser session and navigate to <https://portal.azure.com>.



2. In the Azure Portal, type subscription in the search bar at the top of the screen. Click on the subscriptions icon. (You may need to log in first.)

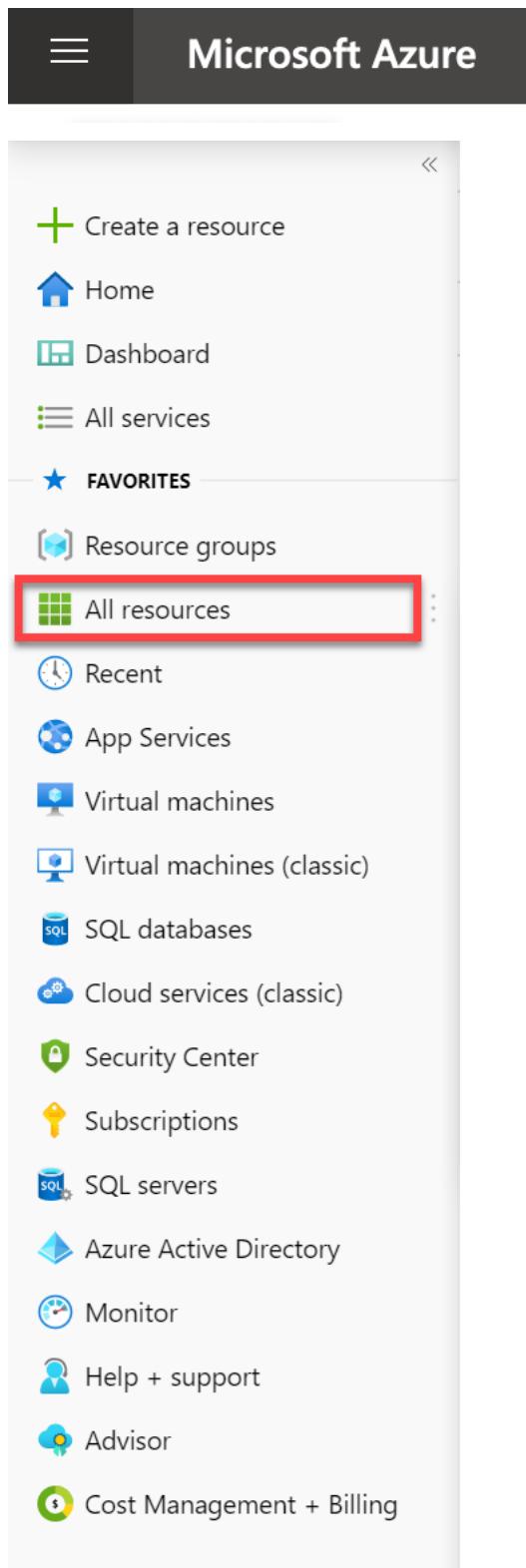
A screenshot of the Microsoft Azure portal search results. The search bar at the top contains the text 'subscription'. Below the search bar, the 'Azure services' section is visible, showing options like 'Subscriptions', 'Event Grid Subscriptions', and 'Resource groups'. A 'Create a' button is also present.

3. On the Subscriptions blade, you will have one or more subscriptions. Review your Subscription name, subscription ID, and your role.

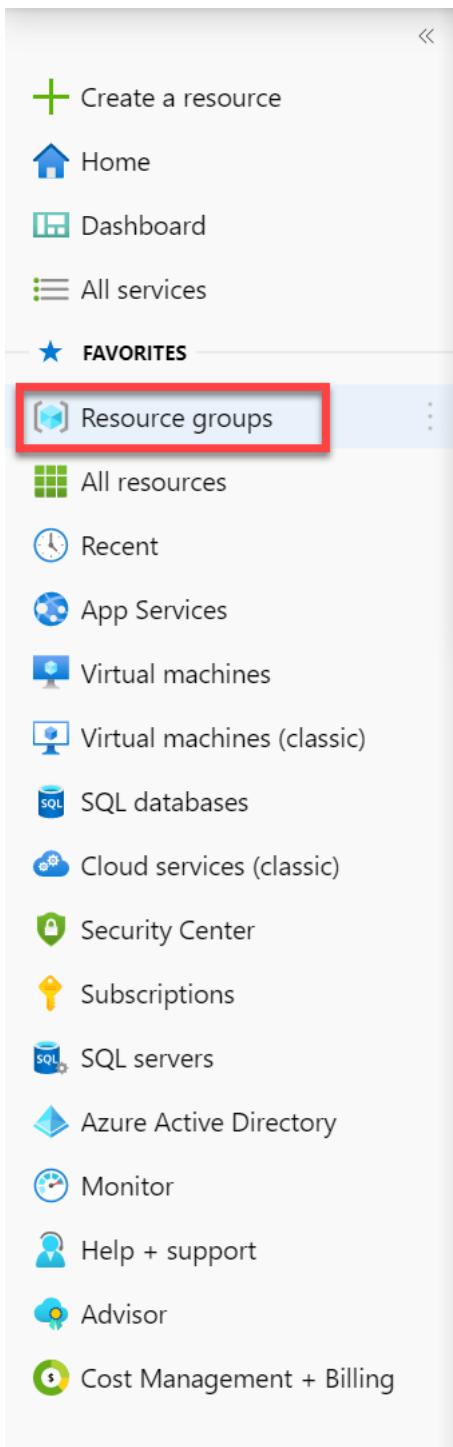
A screenshot of the Microsoft Azure 'Subscriptions' blade. The left navigation pane shows 'Contoso Ltd.' under 'Subscriptions'. The main area displays a single subscription entry: 'Contoso Ltd.' with '8 selected' and '3 selected'. The 'Status' is 'Active'. The table below lists the subscription details: 

Subscription name	Subscription ID	My role	Current cost	Status
Contoso Ltd.		Account admin	Not available	Active

4. Launch the left navigation pane by clicking the horizontal lines next to Microsoft Azure in the top left of the portal. Then select All resources.



5. The All resources page will list all Azure resources within your selected subscriptions.
6. Select Resource groups in the left navigation.



The Resource groups blade will list all Azure resource groups within your selected subscriptions.

7. On the Resource groups blade, select the Create button.

A screenshot of the 'Resource groups' blade. At the top, it says 'Resource groups' with a gear icon and three dots. Below that is a 'Default Directory' dropdown. The main area has a 'Create' button highlighted with a red box. Other buttons include 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Feedback', and 'Assign tags'. At the bottom, there are filter options: 'Filter for any field...', 'Subscription == all', 'Location == all', and 'Add filter'.

8. Ensure you have selected the appropriate subscription for the lab. Name the resource group DP-300-Lab01. Select the nearest Azure region to your location. Select the Review + Create button. Once the review completes, select the Create button.

## Create a resource group

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

### Project details

Subscription \* ⓘ



Resource group \* ⓘ

DP-300-Lab01



### Resource details

Region \* ⓘ



[Review + create](#)

[< Previous](#)

[Next : Tags >](#)

9. Locate the search bar at the top of the page. Search for Azure SQL. Select the search result for Azure SQL that appears in the results under Services.

The screenshot shows the Microsoft Azure portal's search interface. In the top navigation bar, there is a search bar containing the text "Azure SQL". Below the search bar, the left sidebar includes links for "Create a resource", "Home", "Dashboard", "All services", "FAVORITES" (which contains "Resource groups", "All resources", "Recent", "App Services", "Virtual machines", "Virtual machines (classic)", and "SQL databases"), and "Marketplace". The main content area has three sections: "Services", "Marketplace", and "Documentation". The "Services" section lists several database-related services: "Azure SQL" (highlighted with a red box), "Azure Database for MySQL servers", "SQL databases", "Azure Cosmos DB", "Azure Synapse Analytics (formerly SQL DW)", "Azure Lighthouse", "Azure Sentinel", "Azure Database for PostgreSQL servers", "Azure Blockchain Service", and "Azure Spring Cloud". The "Marketplace" section lists "Azure SQL", "Azure SQL Analytics (Preview)", "Azure SQL Managed Instance", and "Discovery Hub® with Azure SQL MI". The "Documentation" section provides links to "What is the Azure SQL Database service? - Azure SQL ...", "Documentation - Azure SQL Database | Microsoft Docs", "Types of query performance issues in Azure SQL Database ...", and "Azure Backup support matrix for SQL Server Backup in Azure ...". The "Resource Groups" section indicates "No results were found". At the bottom of the page, it says "Searching all subscriptions. [Change](#)".

## 8.2 Task 2: Provision a SQL Server on an Azure Virtual Machine

1. On the Azure SQL blade, select the Create button.

The screenshot shows the Azure SQL blade. At the top, there's a header with 'Azure SQL' and a 'Default Directory'. Below the header is a toolbar with various icons: 'Create' (highlighted with a red box), 'Reservations', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Feedback', 'Assign tags', and 'Delete'. There are also filter options for 'Subscription == all', 'Resource group == all', 'Location == all', and a 'Add filter' button.

2. On the Select SQL deployment option blade, open the drop-down box under SQL virtual machines. Select the option labeled Free SQL Server License: SQL 2019 Developer on Windows Server 2019. Then select the Create button.

The screenshot shows the 'Select SQL deployment option' blade. It has three main sections: 'SQL databases', 'SQL managed instances', and 'SQL virtual machines'. The 'SQL virtual machines' section is expanded, showing a dropdown menu for 'Image'. The 'Free SQL Server License: SQL 2019 Developer on Windows Server 2019' option is highlighted with a red box. Other options listed in the dropdown include 'SQL Server 2019 Enterprise Windows Server 2019', 'SQL Server 2019 Standard on Windows Server 2019', '(BYOL) SQL Server 2019 on Windows Server 2019', '(BYOL) SQL Server 2019 Enterprise Windows Server 2019', '(BYOL) SQL Server 2019 Standard on Windows Server 2019', 'SQL Server 2019 on RHEL74', 'Free SQL Server License: SQL Server 2019 Developer on Red Hat Enterprise Linux 7.4', 'SQL Server 2019 Enterprise on Red Hat Enterprise Linux 7.4', and 'SQL Server 2019 Standard on Red Hat Enterprise Linux 7.4'.

3. On the Create a virtual machine blade, enter the following information:

- Subscription: <Your subscription>
- Resource group: **DP-300-Lab01**
- Virtual machine name: **dp300sqlvmlab01**
- Region: <your local region, same as the selected region for your resource group>
- Availability Options: **No infrastructure redundancy required**
- Image: **Free SQL Server License: SQL 2019 Developer on Windows Server 2019 - Gen1**
- Azure spot instance: **No**
- Size: **Standard D2s\_v3** (2 vCPUs, 8 GiB memory; you may need to select the "Select size" link to see this option)
- Administrator account username: **dp300sqlvmlab01admin**
- Administrator account password: **pwd!DP300lab01** (or your own password that meets the criteria)
- Selected inbound ports: **RDP (3389)**
- Already have a Windows Server license?: **No**

Make note of the username and password for later use.

## Create a virtual machine

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Microsoft Azure Sponsorship

Resource group \* ⓘ

DP-300-Lab01



[Create new](#)

### Instance details

Virtual machine name \* ⓘ

dp300sqlvmlab01



Region \* ⓘ

(US) West US 2



Availability options ⓘ

No infrastructure redundancy required



Image \* ⓘ

Free SQL Server License: SQL 2019 Developer on Windows Server 2019



[Browse all public and private images](#)

Azure Spot instance ⓘ

Yes  No

Size \* ⓘ

**Standard D2s v3**

2 vcpus, 8 GiB memory (\$137.24/month)

[Change size](#)

### Administrator account

Username \* ⓘ

dp300sqlvmlab01admin



Password \* ⓘ

\*\*\*\*\*



4. Navigate to the Disks tab and review the configuration.

## Create a virtual machine

Basics

**Disks**

Networking

Management

Advanced

SQL Server settings

Tags

Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

### Disk options

OS disk type \* ⓘ

Premium SSD



Encryption type \*

(Default) Encryption at-rest with a platform-managed key



▼ Advanced

5. Navigate to the Networking tab and review the configuration.

Home > Select SQL deployment option > Create a virtual machine

## Create a virtual machine

**Networking**

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.

[Learn more](#)

**Network interface**

When creating a virtual machine, a network interface will be created for you.

**Virtual network \*** (new) DP-300-Lab01-vnet [Create new](#)

**Subnet \*** (new) default (10.0.0.0/24)

**Public IP** (new) dp300sqlvmlab01-ip [Create new](#)

**NIC network security group**  Basic  None  Advanced

**Public inbound ports \***  Allow selected ports  None

**Select inbound ports \*** RDP (3389)

**Accelerated networking**  Off  On

The selected VM size does not support accelerated networking.

[Review + create](#) [< Previous](#) [Next : Management >](#)

6. Navigate to the Management tab and review the configuration.

Home > Virtual machines >

## Create a virtual machine

Basics Disks Networking **Management** Advanced Tags Review + create

Configure monitoring and management options for your VM.

### Azure Security Center

Azure Security Center provides unified security management and advanced threat protection across hybrid cloud workloads.

[Learn more](#)

Your subscription is protected by Azure Security Center basic plan.

### Monitoring

Boot diagnostics  Enable with managed storage account (recommended)  Enable with custom storage account  Disable

OS guest diagnostics  Off  On

### Identity

System assigned managed identity  Off  On

### Auto-shutdown

Enable auto-shutdown  Off  On

Make sure that **Boot diagnostics** has the button for **Enable with managed storage account (recommended)** selected. Verify that **Enable auto\_shutdown** is off.

7. Navigate to the Advanced tab and review the configuration.

The screenshot shows the 'Create a virtual machine' wizard in the Azure portal. The 'Advanced' tab is selected. The 'Extensions' section is visible, showing a note that the selected image does not support cloud init. The 'Host' and 'Proximity placement group' sections are also shown, both indicating no options found. At the bottom, there are 'Review + create' and navigation buttons for the previous and next steps.

Home > Select SQL deployment option > Create a virtual machine

X Create a virtual machine

Basics Disks Networking Management Advanced SQL Server settings Tags Review + create

Add additional configuration, agents, scripts or applications via virtual machine extensions or cloud-init.

**Extensions**

Extensions provide post-deployment configuration and automation.

Extensions ⓘ Select an extension to install

**Cloud init**

Cloud init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files or to configure users and security. [Learn more](#)

ⓘ The selected image does not support cloud init.

**Host**

Azure Dedicated Hosts allow you to provision and manage a physical server within our data centers that are dedicated to your Azure subscription. A dedicated host gives you assurance that only VMs from your subscription are on the host, flexibility to choose VMs from your subscription that will be provisioned on the host, and the control of platform maintenance at the level of the host. [Learn more](#)

Host group ⓘ No host group found

**Proximity placement group**

Proximity placement groups allow you to group Azure resources physically closer together in the same region. [Learn more](#)

Proximity placement group ⓘ No proximity placement groups found

**Review + create** < Previous Next : SQL Server settings >

8. Navigate to the SQL Server settings tab and review the configuration.

Home > Select SQL deployment option > Create a virtual machine

X Create a virtual machine

Basics Disks Networking Management Advanced **SQL Server settings** Tags Review + create

**Security & Networking**

SQL connectivity \* Private (within Virtual Network) ▾

Port \* 1433

**SQL Authentication**

SQL Authentication ⓘ Disable Enable

Azure Key Vault integration ⓘ Disable Enable

**Storage configuration**

Customize performance, size, and workload type to optimize storage for this virtual machine. For optimal performance, separate drives will be created for data and log storage by default. [Learn more about SQL Server best performance practices.](#)

**Storage**

**Storage optimization: Transactional processing**

SQL Data: 1024 GiB, 5000 IOPS, 200 MB/s  
SQL Log: 1024 GiB, 5000 IOPS, 200 MB/s  
SQL TempDb: Use local SSD drive  
[Change configuration](#)

**SQL Server License**

**Review + create** < Previous Next : Tags >

Note—you can also configure the storage for your SQL Server VM on this screen. By default, the SQL Server Azure VM templates create one premium disk with read caching for data, one premium disk without caching for transaction log, and uses the local SSD (D:\ on Windows) for tempdb.

9. Select the Review + create button. Then select the Create button.

Home > Select SQL deployment option > Create a virtual machine

**Create a virtual machine**

Validation passed

Basics Disks Networking Management Advanced SQL Server settings Tags **Review + create**

**PRODUCT DETAILS**

Standard D2s v3 by Microsoft **Subscription credits apply** 0.1880 USD/hr **Pricing for other VM sizes**

**TERMS**

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

**⚠ You have set RDP port(s) open to the internet.** This is only recommended for testing. If you want to change this setting, go back to Basics tab.

**Basics**

Subscription	Microsoft Azure Sponsorship
Resource group	DP-300-Lab01
Virtual machine name	dp300sqlvmlab01
Region	West US 2
Availability options	No infrastructure redundancy required

**Create** < Previous Next > Download a template for automation

- On the deployment blade, wait until the deployment is complete. The VM will take approximate 5-10 minutes to deploy. Then select the Go to resource button.

Home > CreateVm-microsoftsqlserver.sql2019-ws2019-sqldev-20200508150309 | Overview

**CreateVm-microsoftsqlserver.sql2019-ws2019-sqldev-20200508150309 | Overview**

Deployment

Search (Ctrl+ /) Delete Cancel Redeploy Refresh

**Your deployment is complete**

Deployment name: CreateVm-microsoftsqlserver.sql2019-ws2019-... Start time: 5/8/2020, 4:09:25 PM  
Subscription: Microsoft Azure Sponsorship Correlation ID: e13b9524-7c7a-4b2c-aff3-5c4d2b94eb28  
Resource group: DP-300-Lab01

Deployment details (Download)

Next steps

**Go to resource**

- On the overview page for the virtual machine, scroll through the menu options for the resource to review what is available.

## 9 Exercise 2: Connect to SQL Server and Restore a Backup

**Estimated Time: 10 minutes**

The main tasks for this exercise are as follows:

1. Create an RDP (Remote Desktop Protocol) connection to SQL Server on an Azure Virtual Machine
2. Download a database backup file.
3. Restore the database in SQL Server using SQL Server Management Studio
4. Query the database to confirm its availability

### 9.1 Task 1: Connect to the Virtual Machine and download the backup file

1. On the Overview page for the virtual machine, select the Connect button and choose RDP.

Home > CreateVm-microsoftsqlserver.sql2019-ws2019-sqldev-20200508150309 | Overview > dp300sqlvmlab01

**dp300sqlvmlab01**  
Virtual machine

Search (Ctrl+/  
)

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

**Settings**

Networking Connect Disks Size Security Extensions

**RDP** (highlighted with a red box)

SSH

Bastion

Location: West US 2

Subscription (change): Microsoft Azure Sponsorship

Subscription ID: 0192e014-eabc-49cb-845e-fc66e11fa7b3

Computer name: dp300sqlvmlab01

Operating system: Windows (Windows Server 2019 Datacenter)

Size: Standard D2s v3 (2 vcpus, 8 GiB memory)

Tags (change): Click here to add tags

Azure Spot N/A

Public IP address: 52.250.116.104

Private IP address: 10.0.0.4

Public IP address (IPv6): -

Private IP address (IPv6): -

Virtual network/subnet: DP-300-Lab01-vnet/default

DNS name: Configure

2. On the RDP tab, select the Download RDP File button.

Home > CreateVm-microsoftsqlserver.sql2019-ws2019-sqldev-20200508150309 | Overview > dp300sqlvmlab01 | Connect

**dp300sqlvmlab01 | Connect**  
Virtual machine

Search (Ctrl+/  
)

To improve security, enable just-in-time access on this VM. →

**RDP** (highlighted with a red box), SSH, BASTION

**Connect with RDP**

To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file.

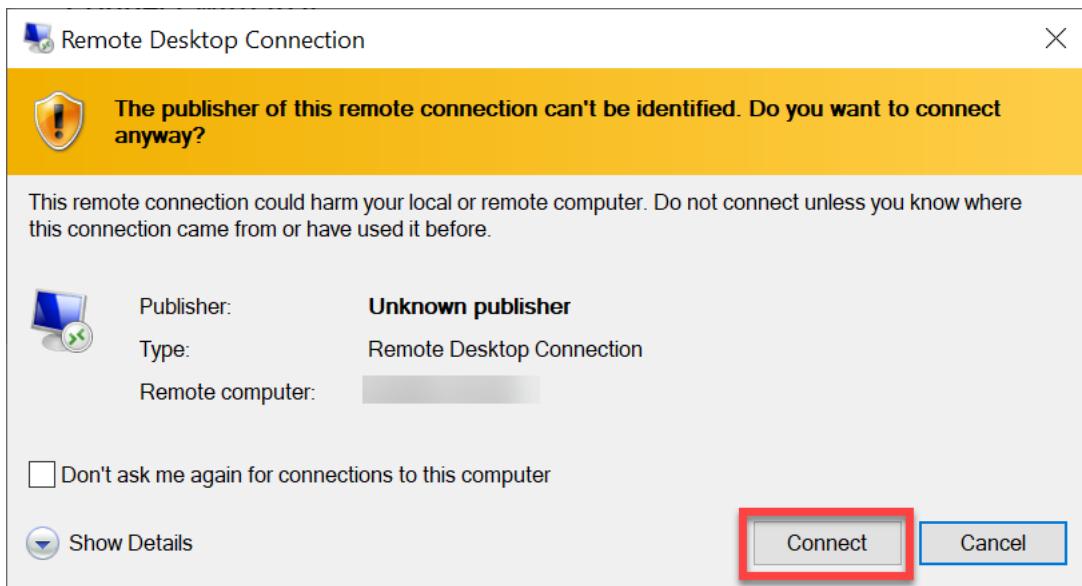
IP address \*: Public IP address (dropdown menu)

Port number \*: 3389

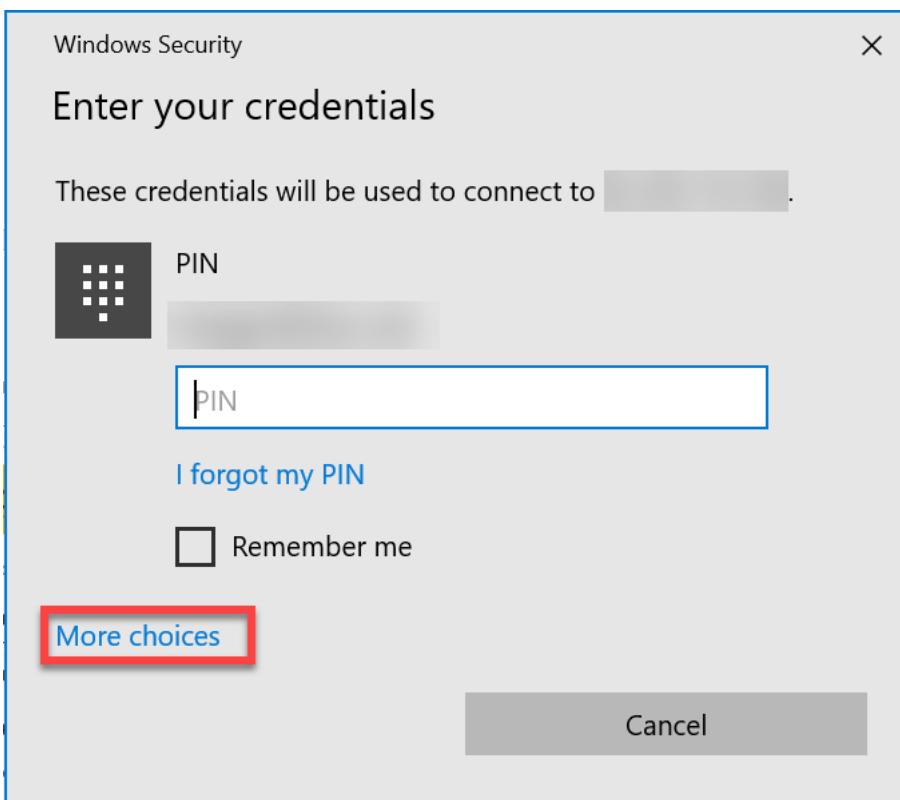
**Download RDP File** (highlighted with a red box)

Can't connect?  
Test your connection  
Troubleshoot RDP connectivity issues

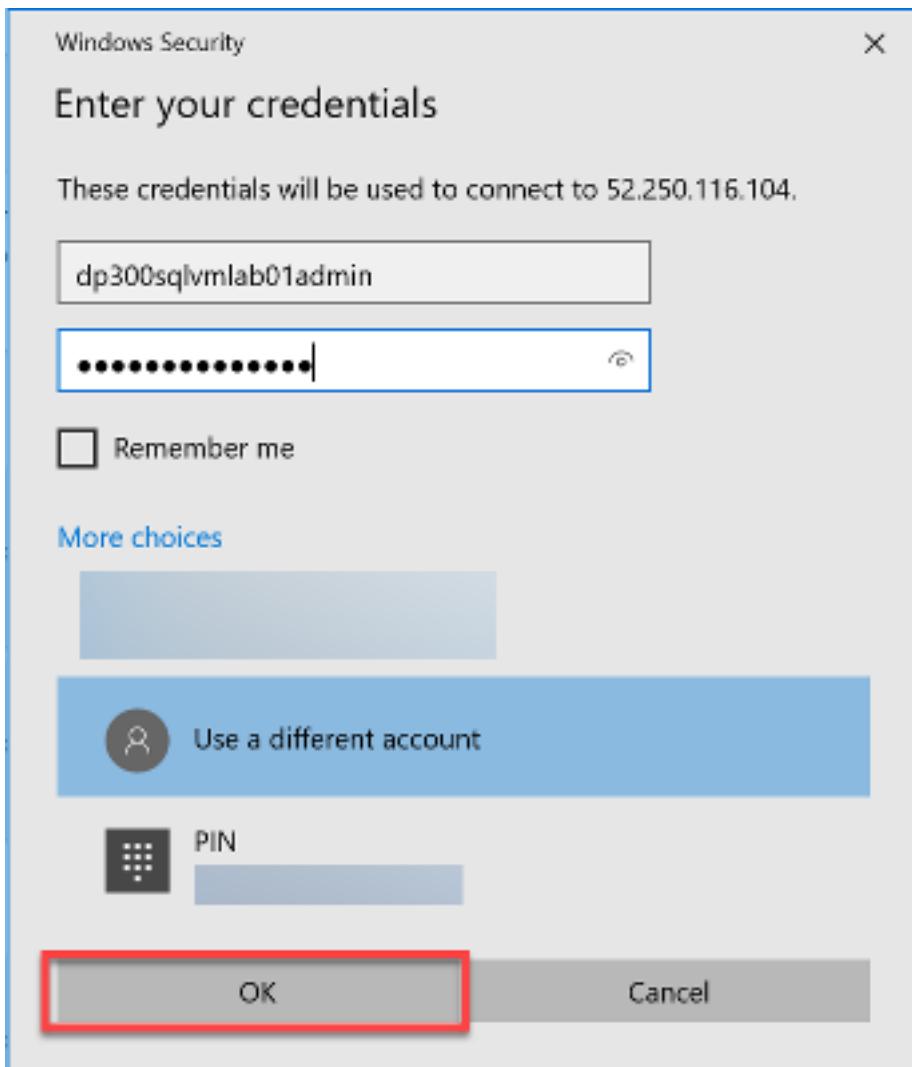
3. Open the RDP file that was just downloaded. When a dialog appears asking if you want to connect, select the Connect button.



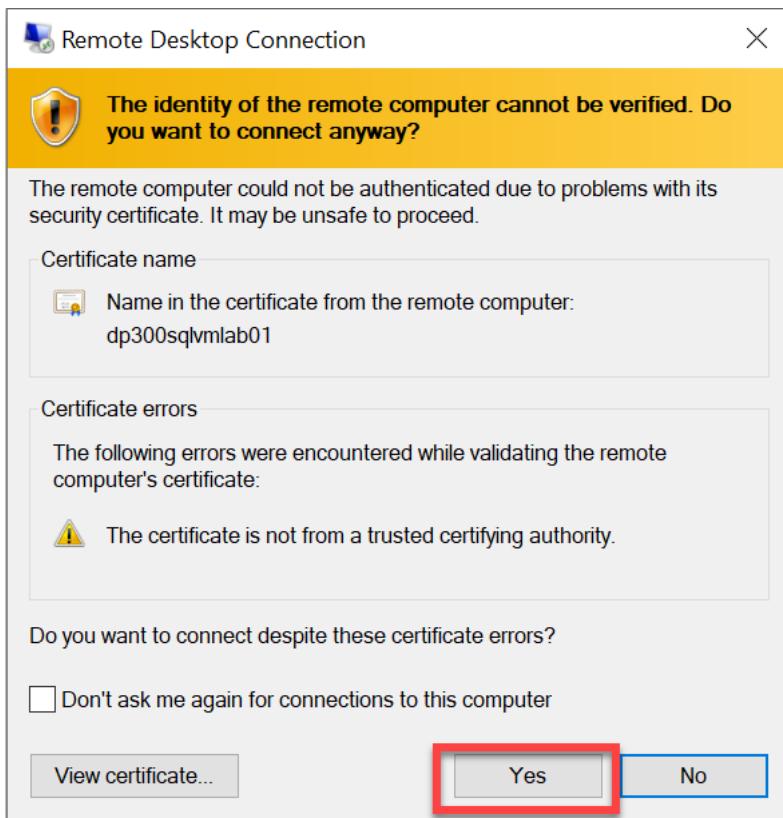
4. In the Windows Security dialog if you receive the PIN dialog, select More choices. Then choose Use a different account. If you don't receive the PIN dialog, you can proceed to Step 5.



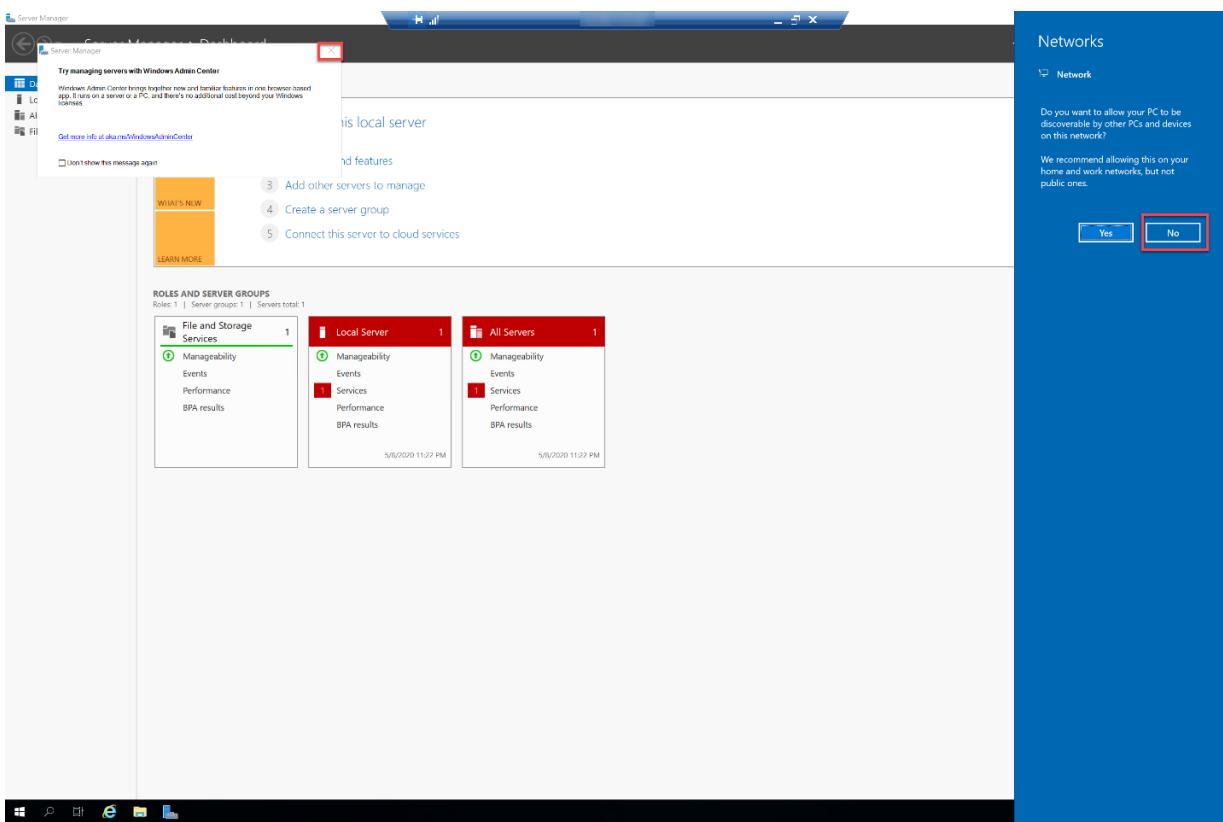
5. Enter the username and password selected during the virtual machine provisioning process. Then select the OK button.



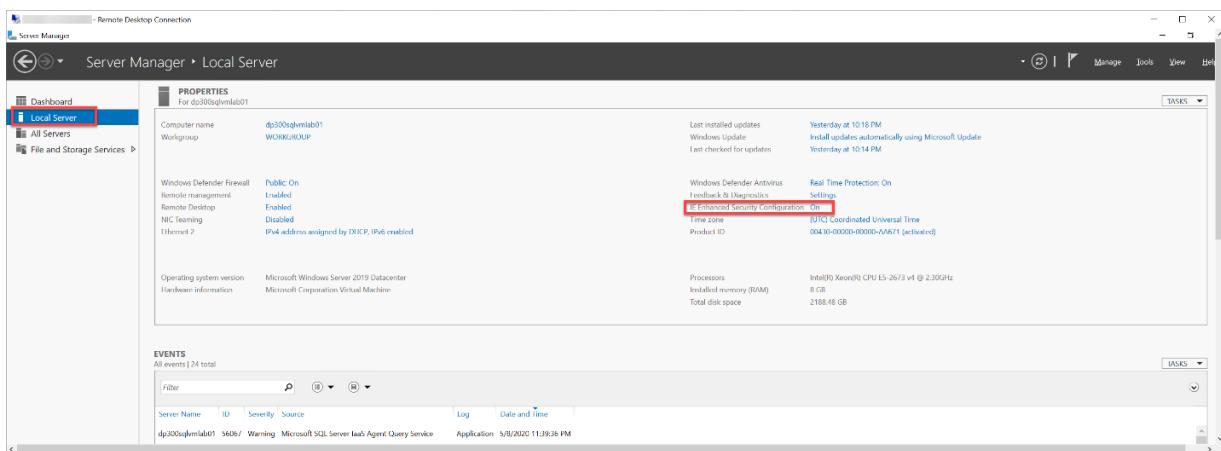
- When the Remote Desktop Connection dialog appears asking if you want to connect, select the Yes button.



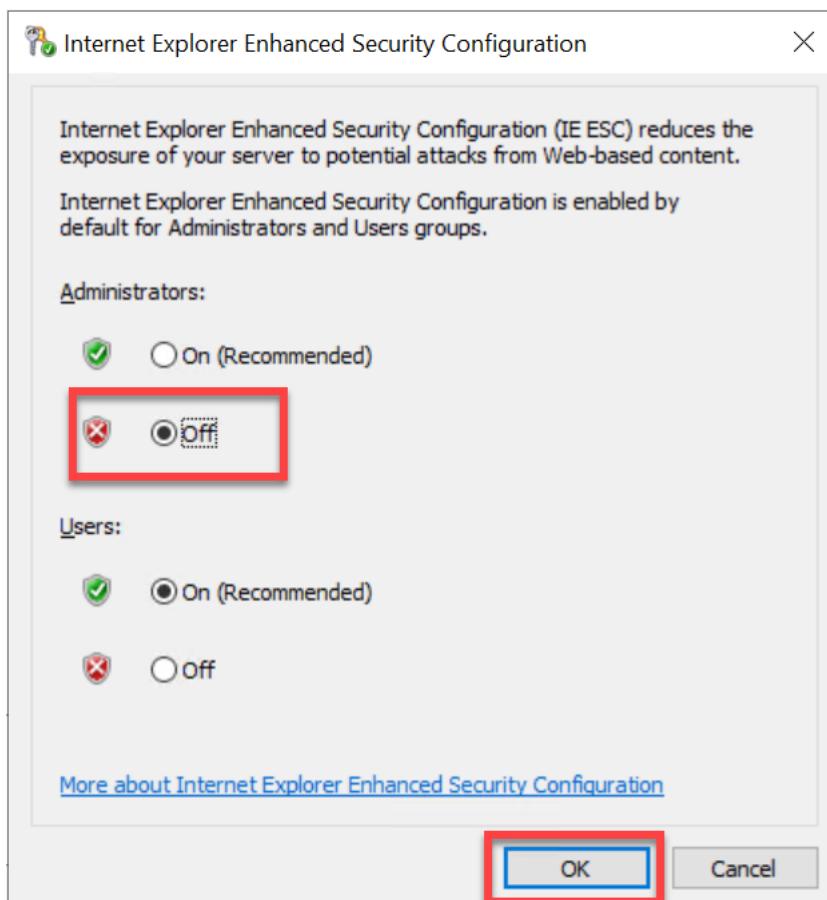
7. A Remote Desktop window will open. On the right, select the No button to indicate you do not want the computer to be discoverable by other devices on the network. Close the small Server Manager popup dialog by clicking the X in the top right of the box. Keep the big Server Manager Window open.



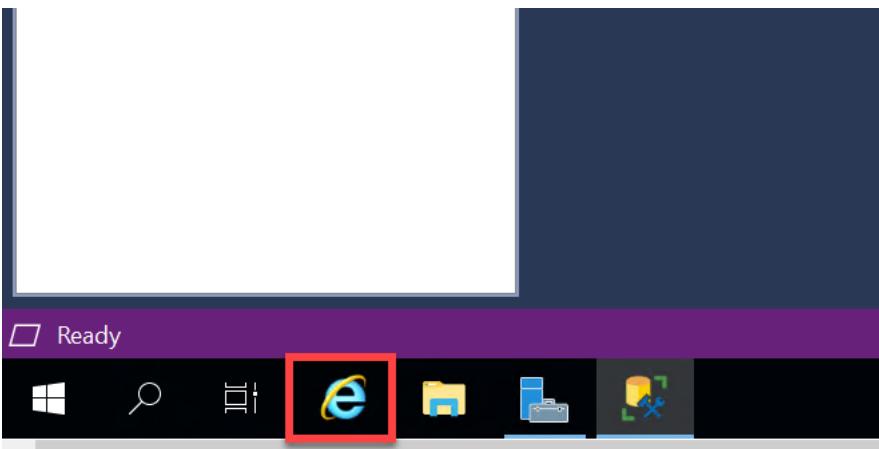
8. In the Server Manager window, select Local Server on the left and then select On next to IE Enhanced Security Configuration.



9. When the Internet Explorer Enhanced Security Configuration dialog appears, set Administrators to Off. Then select the OK button.



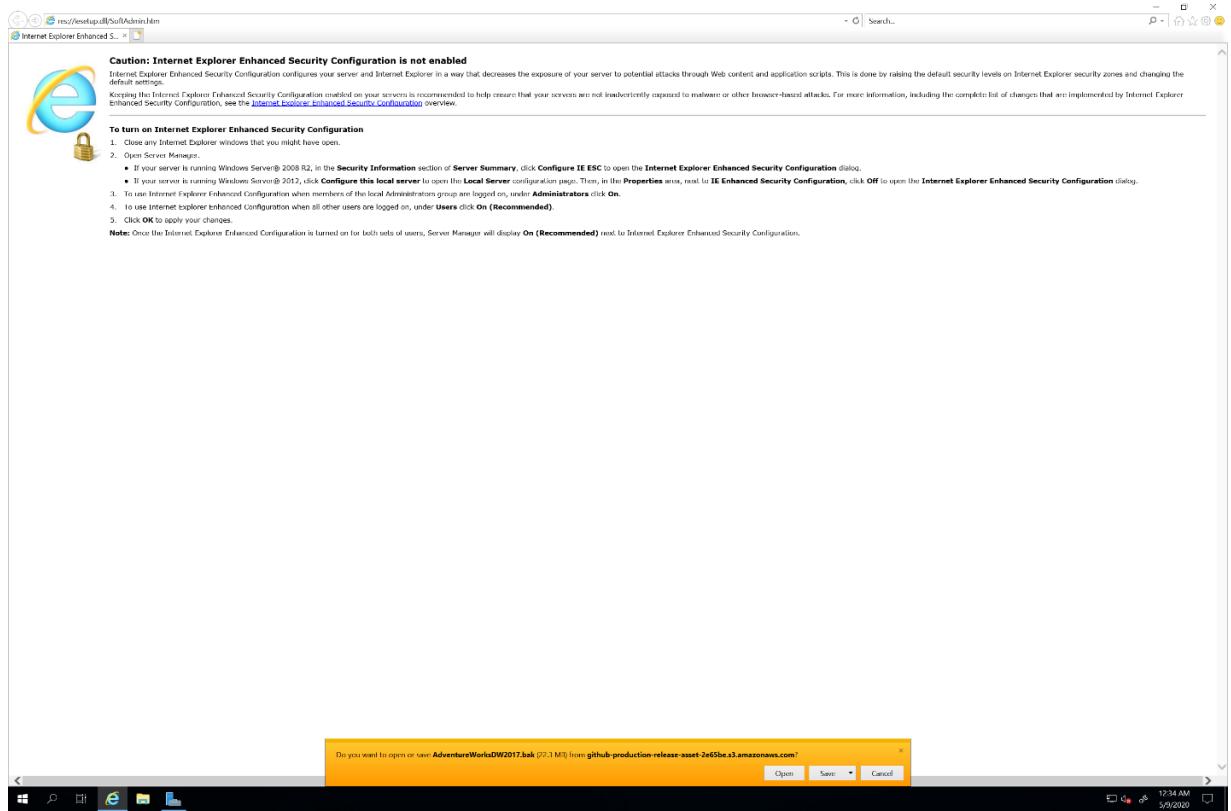
10. Open Internet Explorer by selecting the icon from the task bar.



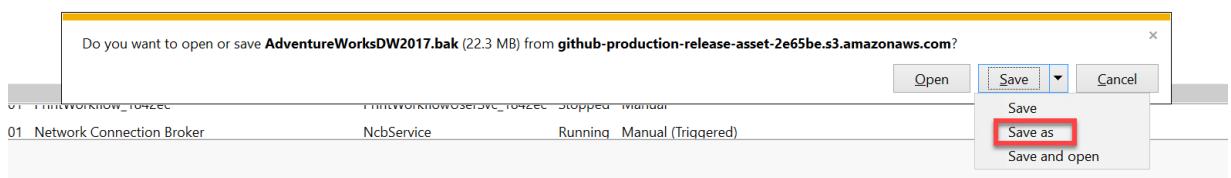
11. In Internet Explorer, navigate to the following URL:

<https://github.com/Microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorksDW2017.bak>

A dialog will appear asking if you would like to run or save the file.



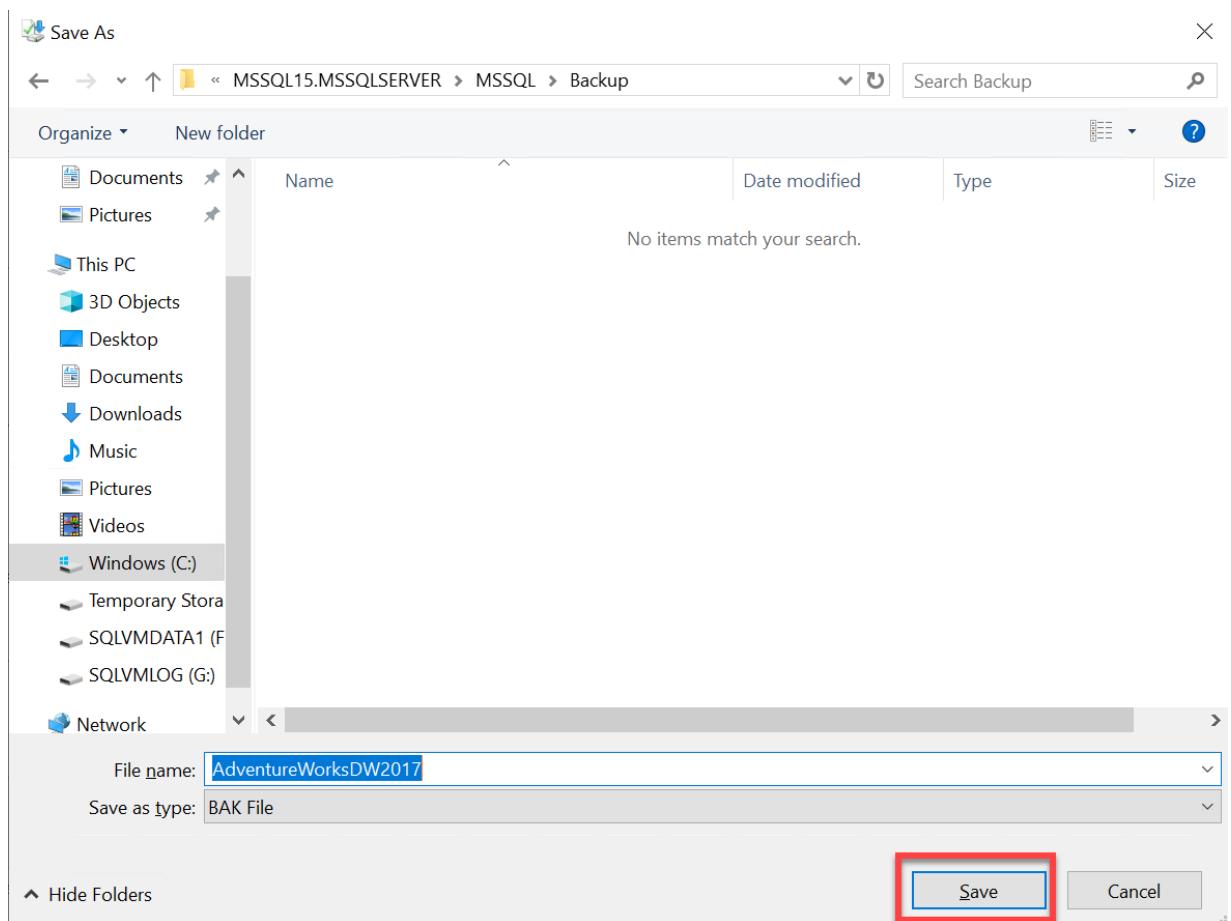
12. In the dialog, select the dropdown next to Save and choose Save As.



13. In the Save As dialog, navigate to the following folder location:

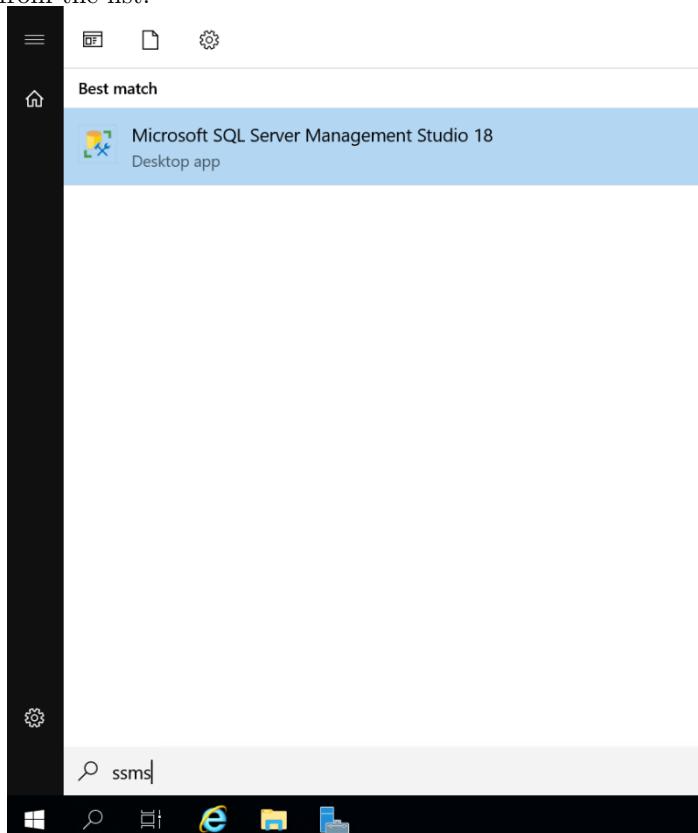
C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup.

Leave the file name as AdventureWorksDW2017 and the type as BAK file. Select the Save button.

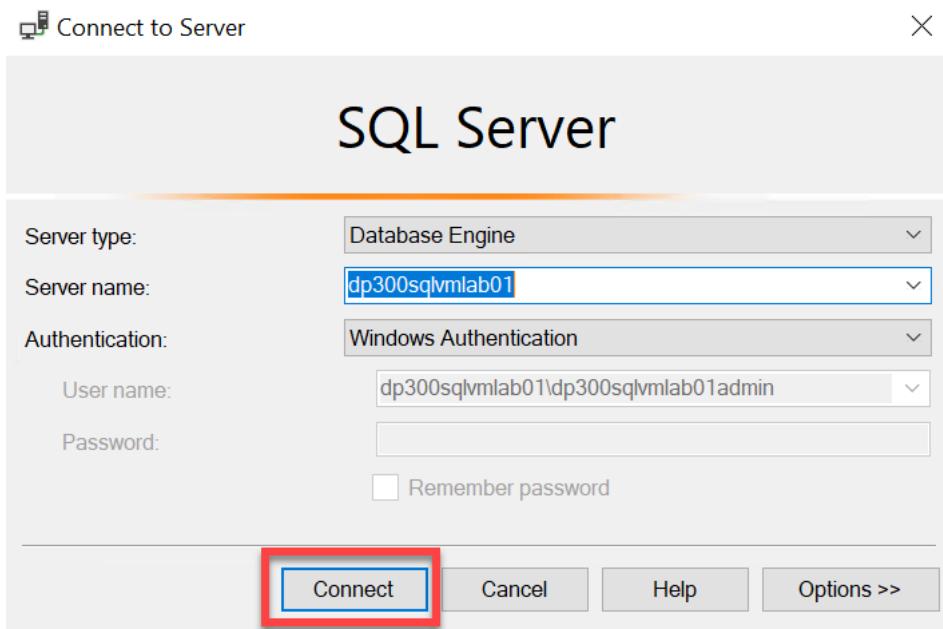


## 9.2 Task 2: Restore the Database and Verify its Availability

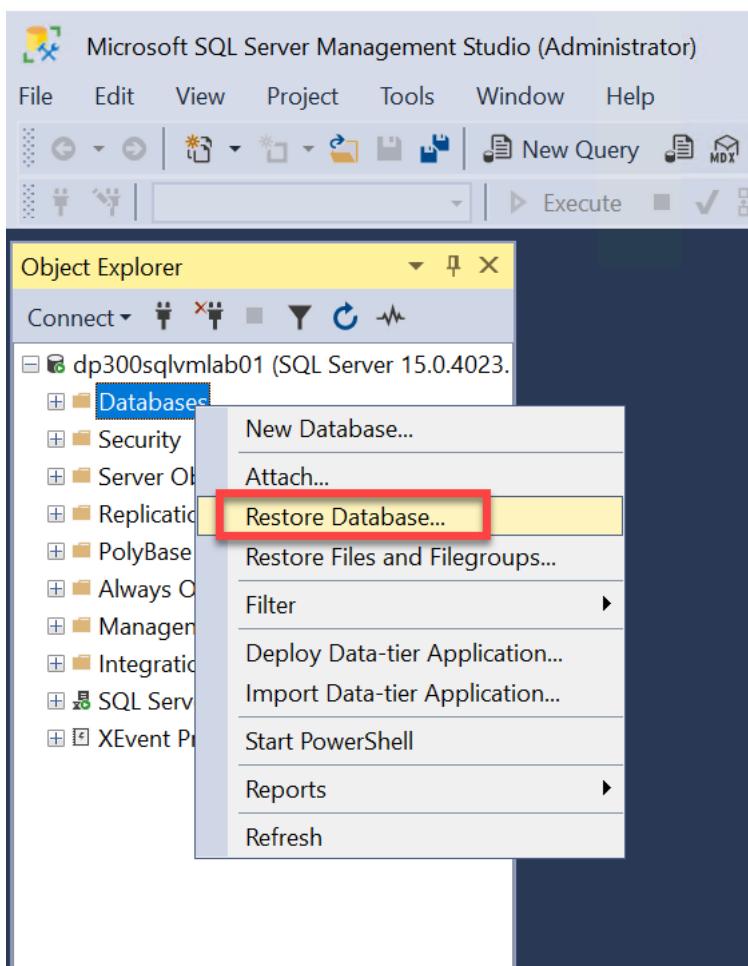
1. Select the Windows Start button and type SSMS. Select Microsoft SQL Server Management Studio 18 from the list.



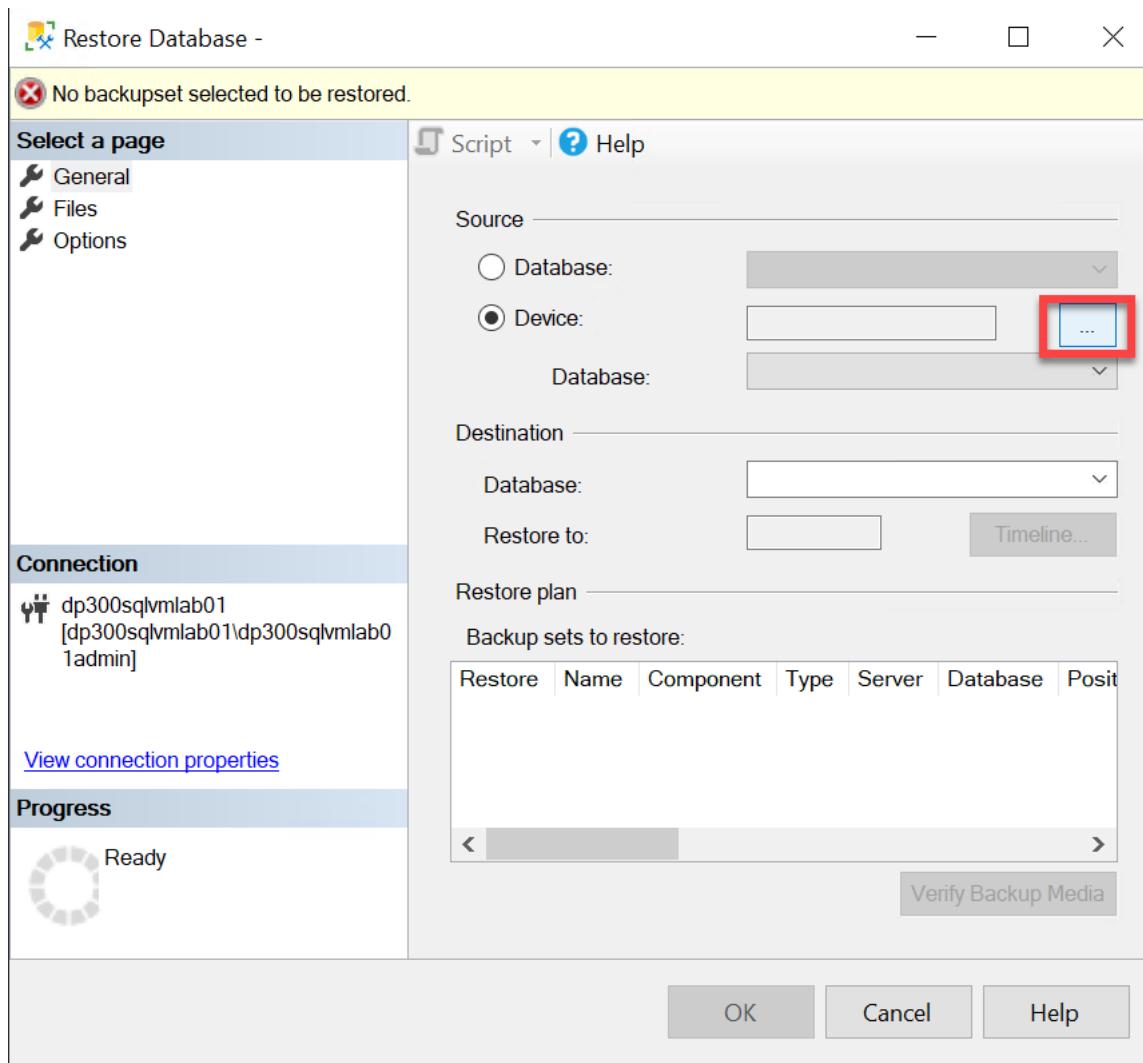
2. When Management Studio opens, the Connect to Server dialog will be pre-populated to the local machine. Select the Connect button.



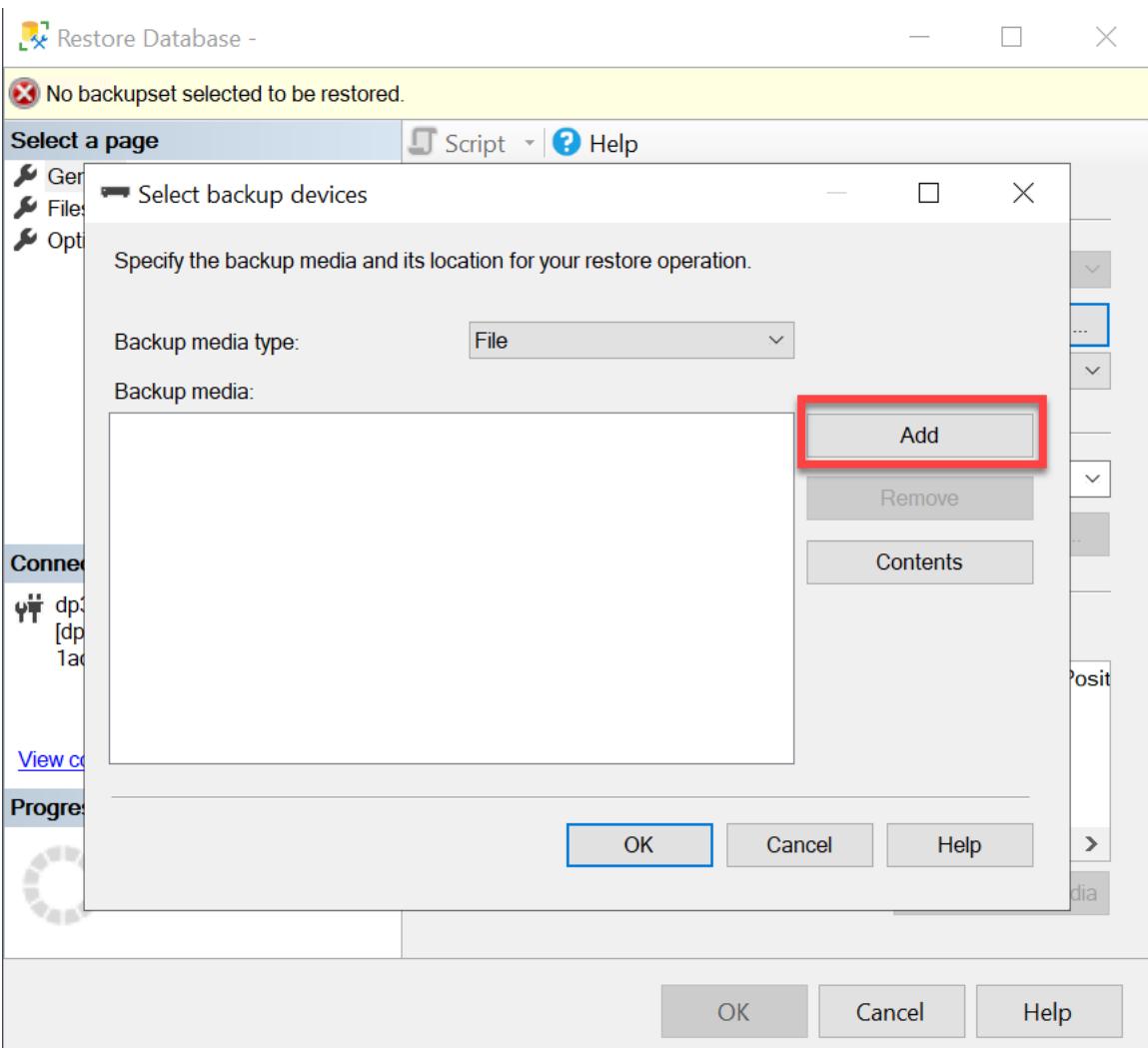
3. In the Object Explorer, right-click on the Databases node and choose Restore Database...



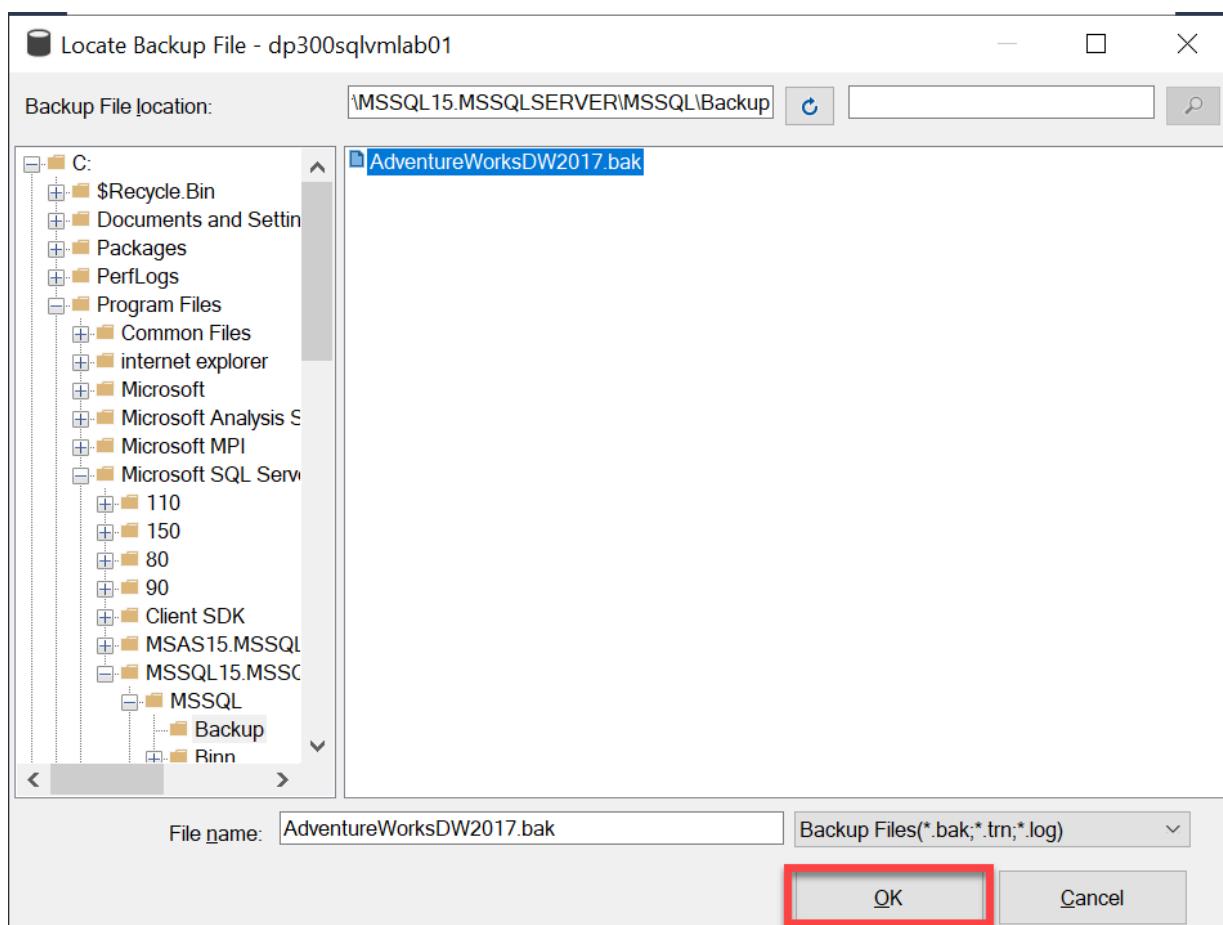
4. In the Restore Database dialog, Select Device under Source, and select the button to the right of device labeled with an ellipsis.



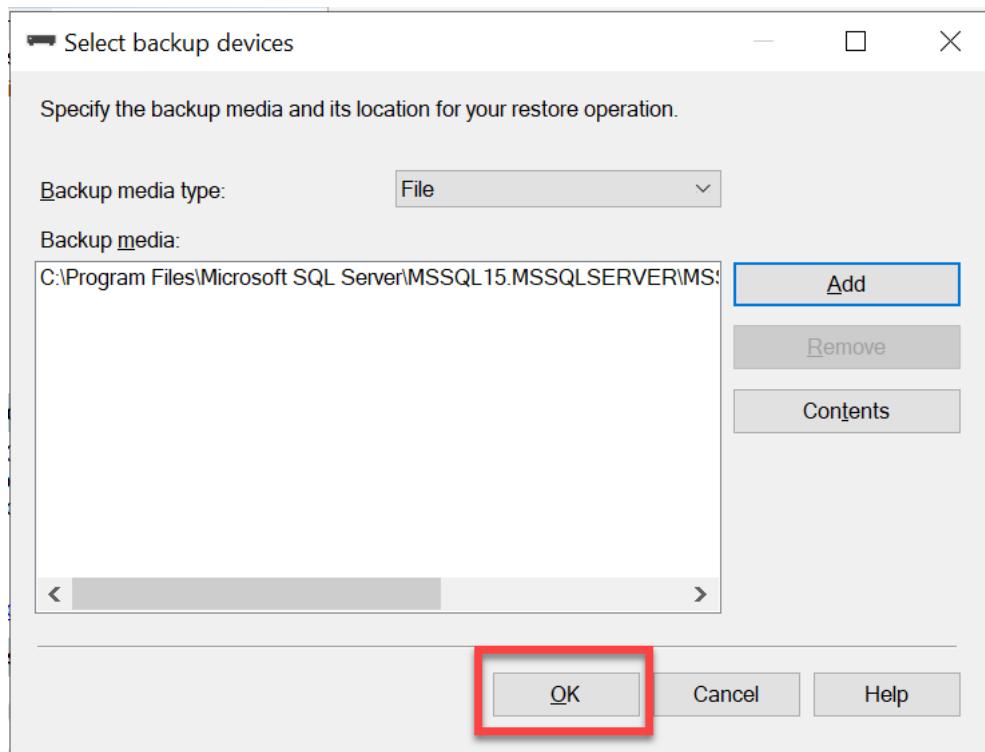
5. In the Select backup devices dialog, select Add.



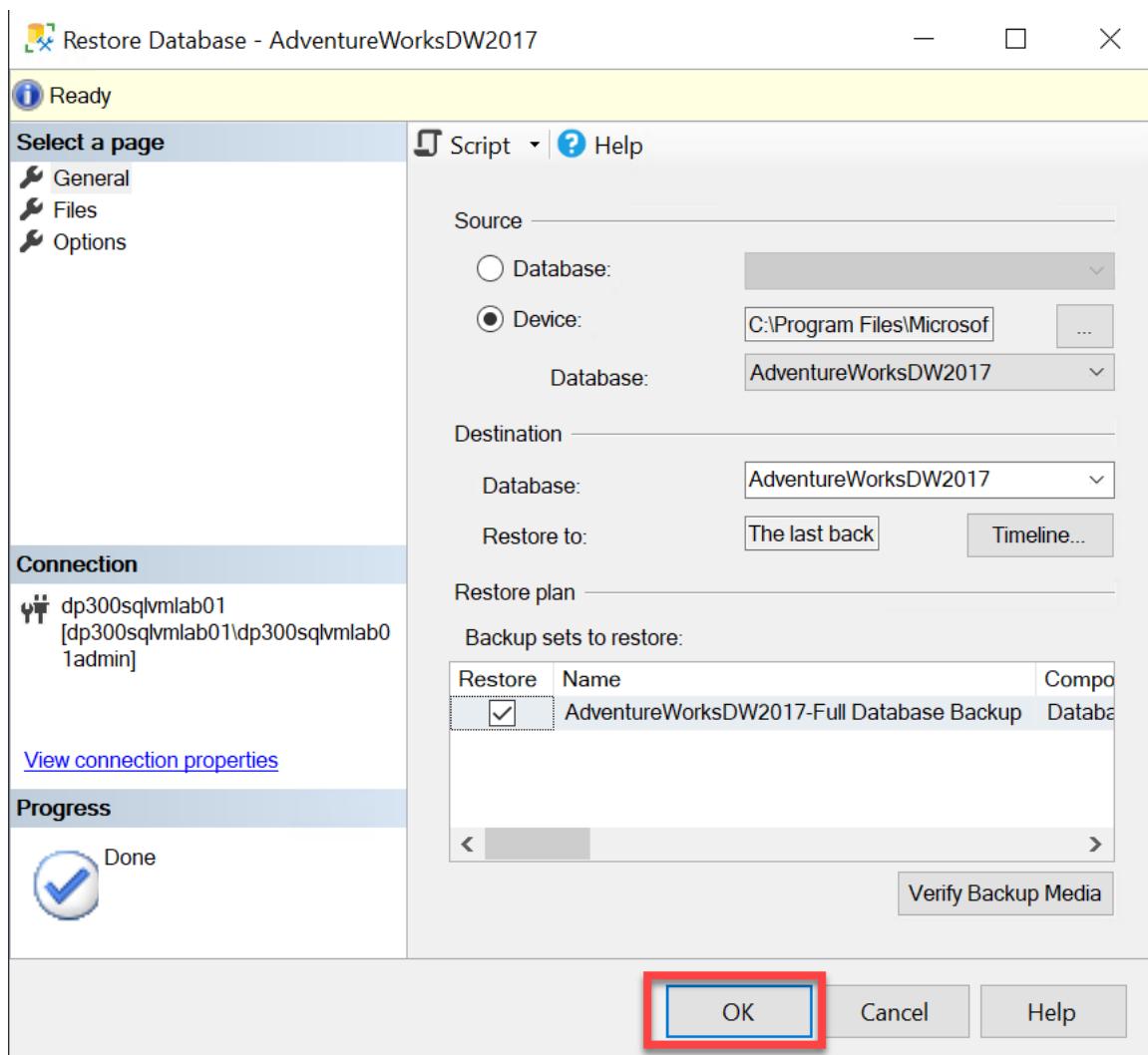
6. In the Local Backup File dialog, select AdventureWorksDW2017.bak and select the OK button.



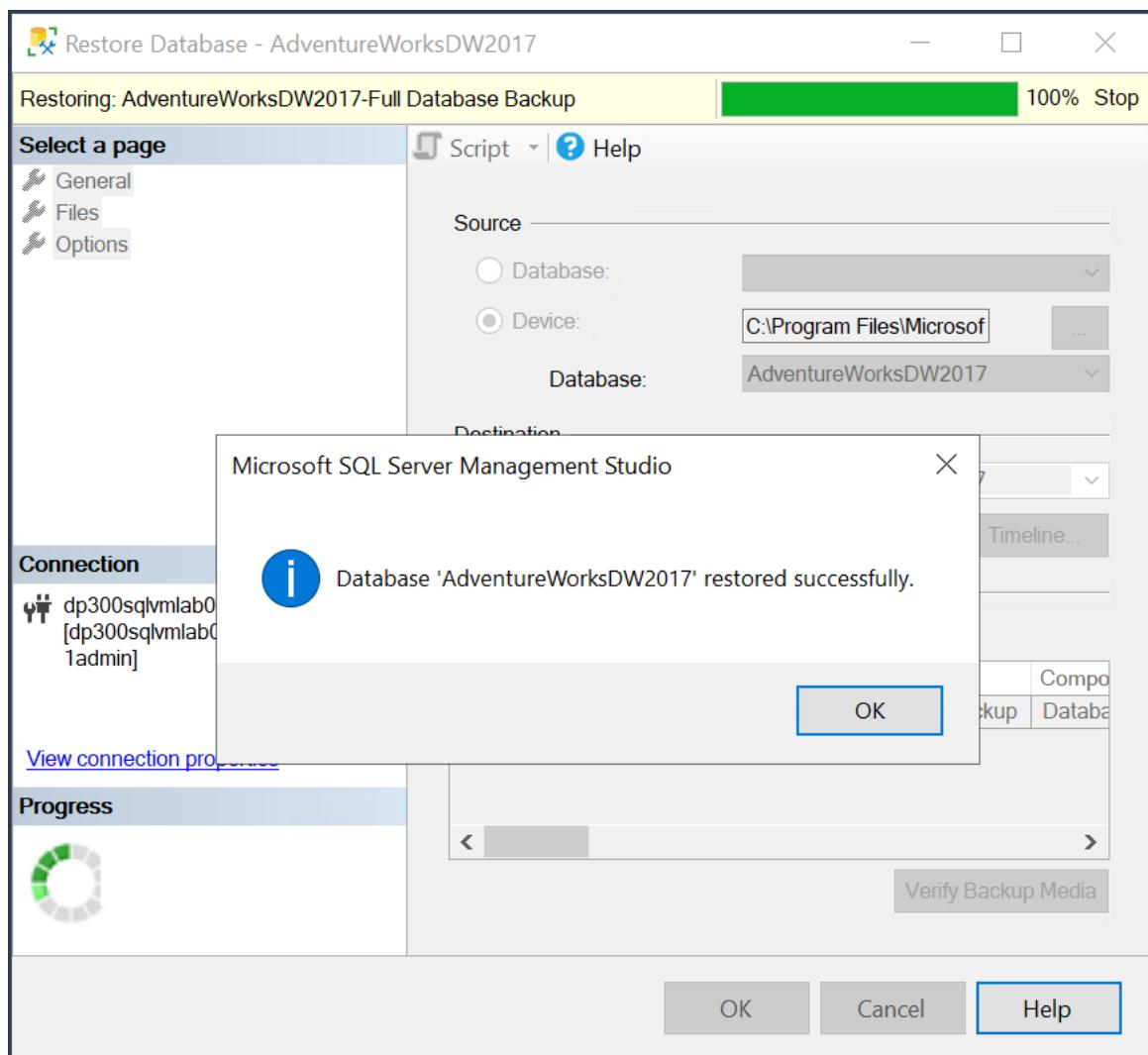
7. Verify that there is now one item listed under Backup media in the Select backup devices dialog. Then select the OK button.



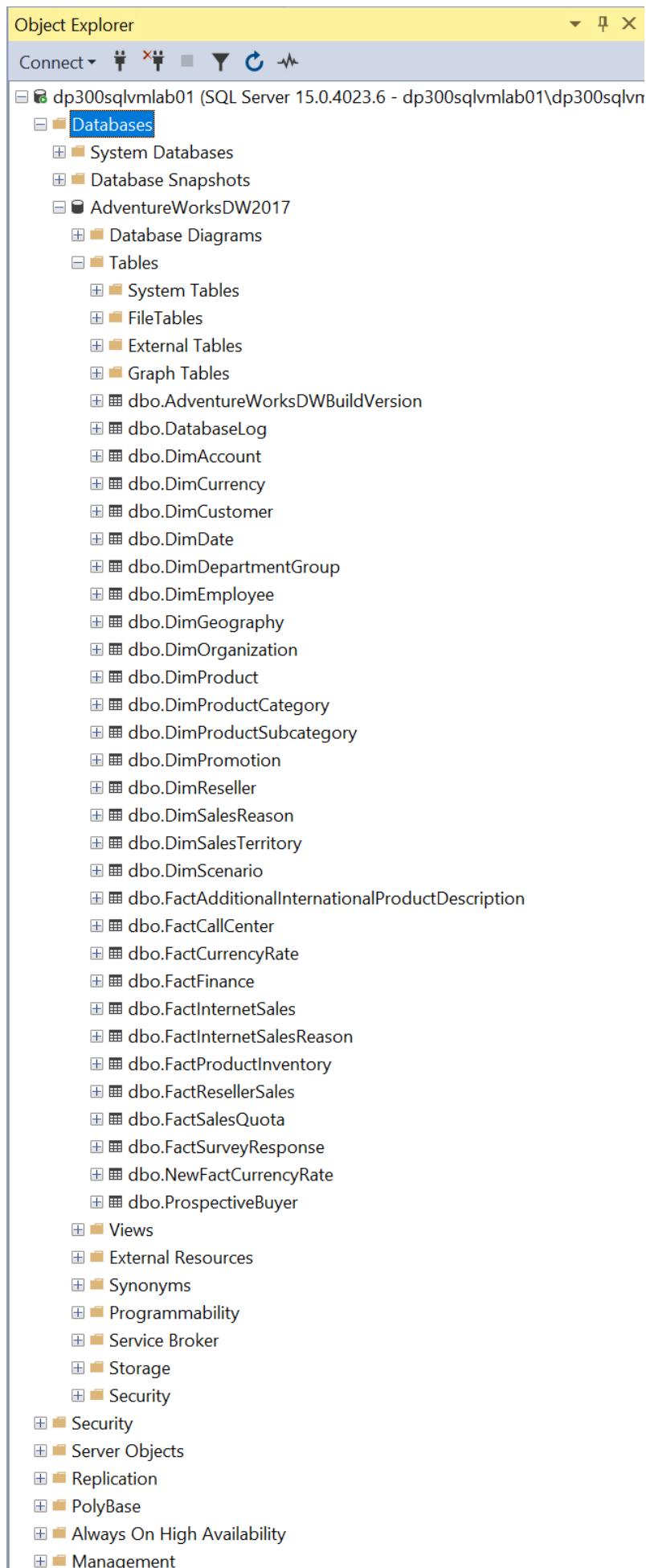
8. Verify that there is one item listed under Backup sets to restore in the Restore Database dialog. Then select the OK button.



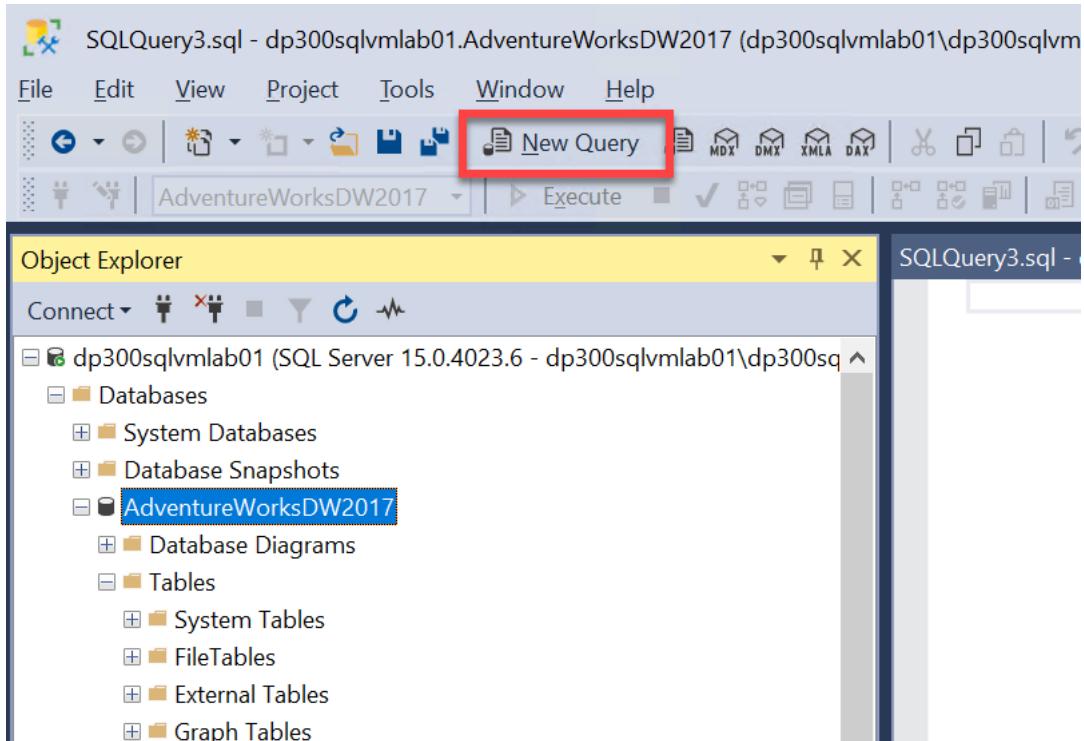
- Once the database restore completes, a dialog box will appear with the message "Database 'AdventureWorksDW2017' restored successfully". Select the OK button.



10. In the Object Explorer, expand the Databases node, then expand the AdventureWorksDW2017 node, then expand the Tables node. Verify that the database contains several tables including dbo.DimAccount.

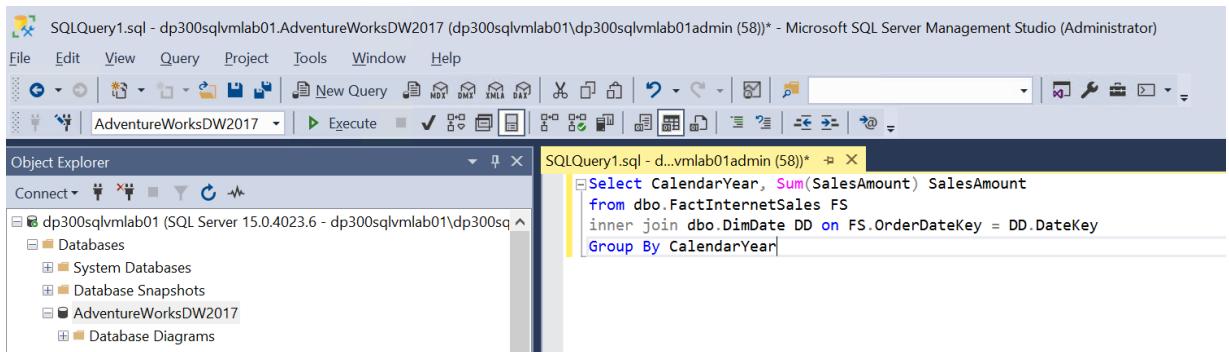


11. In Management Studio, select the AdventureWorksDW2017 node in the Object Explorer. Then select the New Query button.

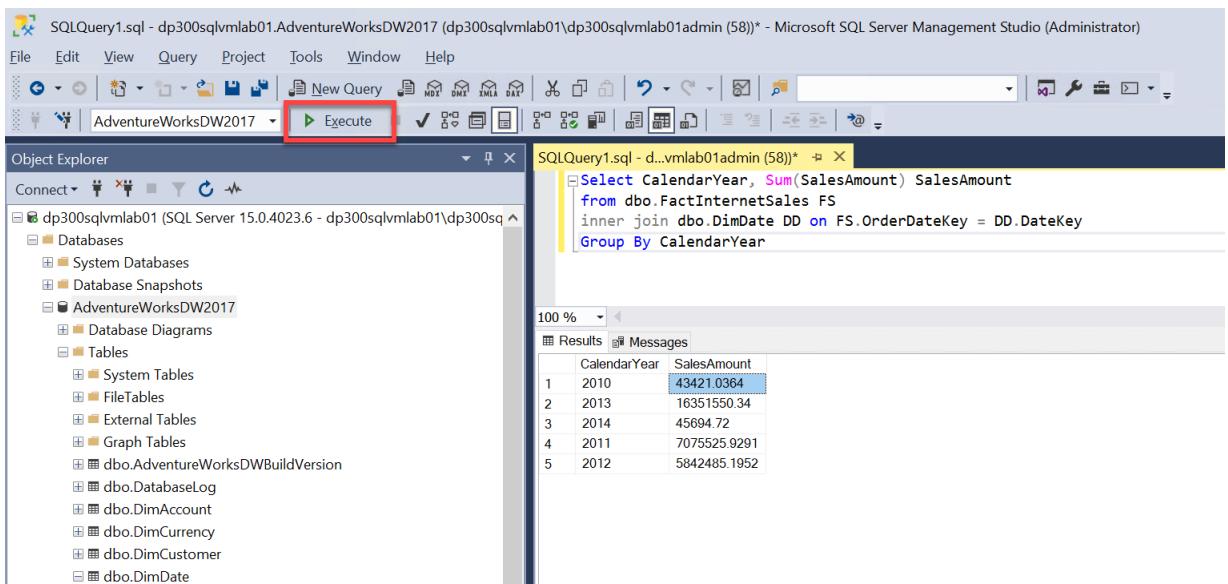


12. Copy and paste the following statement into the new query window:

```
SELECT CalendarYear, Sum(SalesAmount) SalesAmount
FROM dbo.FactInternetSales FS
INNER JOIN dbo.DimDate DD ON FS.OrderDateKey = DD.DateKey
GROUP BY CalendarYear
```



13. Select the Execute button execute the query. Verify that the sales amount for 2010 is **43421.0364**.



### 9.3 Task 3: Clean Up

1. Exit out of the Remote Desktop.
  2. Delete the VM you created in this lab to save costs. You will not be using it for subsequent labs.
  3. Navigate to the main (Overview) blade for the **dp300sqlvmlab01** Virtual Machine and click the delete button in the menu.
- 
4. Click **OK**.
- 

### 9.4 lab: title: 'Lab 2 – Deploying PaaS databases' module: 'Plan and Implement Data Platform Resources'

## 10 Lab 2 – Deploying PaaS databases

**Estimated Completion Time:** 35-40 minutes

**Pre-requisites:** None

**Lab files:** None

## 11 Lab Overview

The student will configure basic resources needed to deploy an Azure SQL Database with a Virtual Network Endpoint. Connectivity to the SQL Database will be validated using Azure Data Studio from the lab VM. Finally, an Azure Database for PostgreSQL will be created.

## 12 Lab Objectives

The students will:

1. Configure basic resources
2. Deploy an Azure SQL Database
3. Connect to Azure SQL Database with Azure Data Studio
4. Deploy an Azure Database for PostgreSQL

## 13 Scenario

As a database administrator for AdventureWorks, you will set up a new SQL Database, including a Virtual Network Endpoint to increase and simplify the security of the deployment. Azure Data Studio will be used to evaluate the use of a SQL Notebook for data querying and results retention.

Finally, an Azure Database for PostgreSQL will be deployed to support additional data system needs.

## 14 Exercise 1: Configure Basic Resources

### 14.1 Task 1: Create a Resource Group

1. Start a browser, and open the Azure Portal at <http://portal.azure.com>, logging in with the appropriate credentials
2. From the Home screen, click on the **Resource Groups** button

3. Review your existing Resource Groups and then click on the **Add** button to create a new Resource Group.

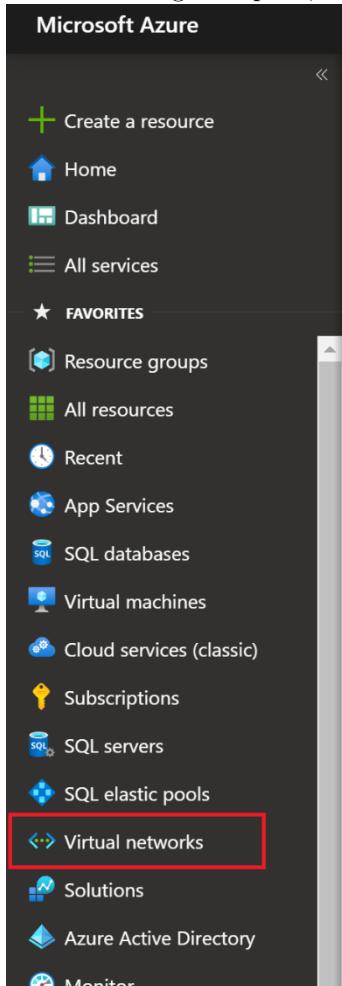
4. Complete the Create a Resource Group wizard with the required information to create the RG.

- Ensure Subscription is set to the desired subscription
- Enter **DP-300-Lab02** for the name of the Resource Group
- For the purposes of this lab, select the Region nearest to your physical location
- Click the **Review + create** button

- Click the **Create** button

## 14.2 Task 2: Create a Virtual Network

1. In the left navigation pane, click **Virtual Networks**



2. Click **+ New** to open the **Create Virtual Network** page. On the **Basics** tab, complete the following information:

- Subscription: **Select the lab subscription**
- Resource Group: Select the **DP-300-Lab02** Resource Group
- Name: **Lab02-vnet**
- Region: Select the same region where the Resource Group was created (the region nearest to your location)

**Create virtual network**

Basics IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

**Project details**

Subscription \* ⓘ AdventureWorks Production

Resource group \* ⓘ DP-300-Lab02

Create new

**Instance details**

Name \* Lab02-vnet

Region \* (US) East US

- Click the **Next: IP Addresses** button

**Review + create**

< Previous

**Next : IP Addresses >**

### 3. Configure the virtual network's IP range for the Azure SQL database endpoint

- On the IP Addresses page, leave the defaults for the IPv4 address space.
- Click on the **default** subnet. (Note that the Subnet address range you see might be different.)

Home > Virtual networks >

### Create virtual network

Basics IP Addresses Security Tags Review + create

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

IPv4 address space

10.0.0.0/16 10.0.0.0 - 10.0.255.255 (65536 addresses)



Add IPv6 address space ⓘ

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

+ Add subnet Remove subnet

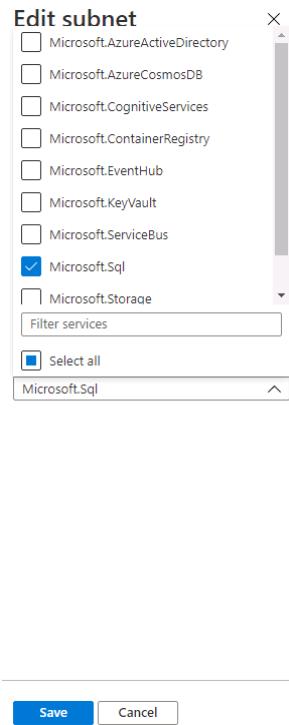
Subnet name

Subnet address range

default

10.0.2.0/24

- In the Edit subnet flyout on the right, expand the Services drop-down, and tick **Microsoft.Sql**

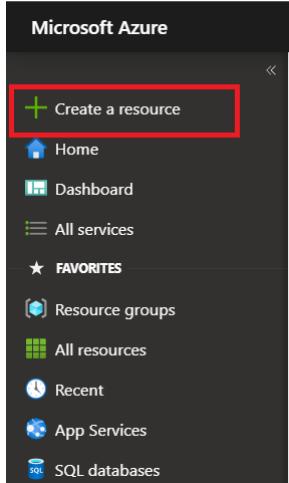


- Click **Save**
- Click the **Review + Create** button, review the settings for the new virtual network, and then click **Create**

## 15 Exercise 2: Deploy an Azure SQL Database

### 15.1 Task 1: Deploy an Azure SQL Database

1. From the Azure Portal, click on **+ Create a Resource** at the top of the left side navigation bar



2. Search for “SQL database” in the search box at the top, then click **SQL Database** from the list of options

The screenshot shows the Azure Marketplace interface. At the top, there is a navigation bar with 'Home > New > Marketplace'. Below this is a search bar containing 'SQL Database'. To the left, there is a sidebar titled 'Categories' with various service categories listed: Get Started, AI + Machine Learning, Analytics, Blockchain, Compute, Containers, Databases, and Developer Tools. The 'Get Started' category is currently selected. The main area displays search results under the heading 'Showing All Results'. The first result is 'SQL Database' by Microsoft, which is highlighted with a red box. This result includes a brief description: 'Scalable and managed relational database service for modern business-class apps.' To the right of this result is another item: 'SQL Database Reserved vCores' by Microsoft, with a description about Azure SQL Database Reserved vCores. There are also other items partially visible below them.

3. Click the **Create** button
4. Complete the Create SQL Database Basics screen with the following inputs and then click **Next: Networking**
  - Subscription: Select the lab subscription
  - Resource Group: **DP-300-Lab02** (the RG created in Exercise 1)
  - Database Name: **AdventureWorksLT**
  - Server: click **Create new**. In the New Server sidebar, complete the form as follows:
    - Server name: **dp300-lab-<your initials (lower case)>** (server name must be globally unique)
    - Server admin login: **dp300admin**
    - Password: **dp300P@ssword!**
    - Confirm password: **dp300P@ssword!**
    - Location: Select the Region nearest to you (same as in Exercise 1)
    - Your New server sidebar should look similar to the one below. Click **OK**

# New server

Microsoft

Server name \*

 ✓  
.database.windows.net

Server admin login \*

 ✓

Password \*

 ✓

Confirm password \*

 ✓

Location \*

 ✓

- Want to use Elastic Pool: **No**
- Compute + Storage: Click **Configure database**
  - On the Configure screen, click **Looking for basic, standard, premium?**

Configure

Feedback

Looking for basic, standard, premium?  

**General Purpose**  
Scalable compute and storage options  
500 - 20,000 IOPS  
2-10 ms latency

Compute tier

**Provisioned** ✓  
Compute resources are pre-allocated  
Billed per hour based on vCores configured

**Serverless**  
Compute resources are provisioned  
Billed per second based

Compute Hardware

– Click **Basic**  
– Click the **Apply** button

**Note:** Make note of this server name, and your login information. You will use it in subsequent labs.

- Review settings and then click **Next: Networking**

# Create SQL Database

Microsoft

Basics Networking Additional settings Tags Review + create

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

## Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ Microsoft Azure Internal Consumption

Resource group \* ⓘ DP-300-Lab02

[Create new](#)

## Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \* AdventureWorksLT ✓

Server \* ⓘ (new) dp300-lab-adw (East US)

[Create new](#)

Want to use SQL elastic pool? \* ⓘ  Yes  No

Compute + storage \* ⓘ

**Basic**  
2 GB storage  
[Configure database](#)

6. On the Networking screen, for Connectivity method, click the **Private endpoint** radio button

Connectivity method \* ⓘ

No access  
 Public endpoint  
 Private endpoint

7. Then click the **Add private endpoint** link under Private Endpoints

Private endpoints

Private endpoint connections are associated with the private endpoint connections for the database and they provide access to all databases.

[+ Add private endpoint](#)

8. Complete the Create private endpoint flyout as follows:

- Subscription: Ensure the lab subscription is selected
- Resource group: **DP-300-Lab02**
- Location: The same Region that was selected for previous parts of this lab
- Name: **DP-300-SQL-Endpoint**

- Target sub-resource: **SqlServer**
- Virtual network: **Lab02-vnet**
- Subnet: **default (10.x.0.0/24)**
- The Private DNS integration options can remain at the default
- Review settings before clicking **OK**

### Create private endpoint

Subscription * ⓘ	AdventureWorks Production
Resource group * ⓘ	DP-300-Lab02 <a href="#">Create new</a>
Location *	(US) East US
Name * ⓘ	DP-300-SQL-Endpoint
Target sub-resource *	SqlServer

### Networking

To deploy the private endpoint, select a virtual network subnet. [Learn more about private endpoint networking ↗](#)

Virtual network * ⓘ	Lab02-vnet
Subnet * ⓘ	default (10.2.0.0/24)

**ⓘ** If you have a network security group (NSG) enabled for the subnet above, disabled for private endpoints on this subnet only. Other resources on the subnet have NSG enforcement.

### Private DNS integration

To connect privately with your private endpoint, you need a DNS record. We recommend that you integrate your private endpoint with a private DNS zone. You can also utilize your own DNS servers or create DNS records using the host files on your virtual machine. [Learn more about private DNS integration ↗](#)

Integrate with private DNS zone ⓘ	<input type="radio"/> Yes <input type="radio"/> No
Private DNS Zone * ⓘ	(New) privatelink.database.windows.net

9. Confirm the endpoint appears on the Networking page, and then click the **Next: Additional settings** button.

Basics Networking Additional settings Tags Review + create

Configure network access and connectivity for your server. The configuration selected below will apply to the selected server 'dp300-lab02-lab' and all databases it manages. [Learn more](#)

**Network connectivity**

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can configure connection method after server creation. [Learn more](#)

Connectivity method \* ⓘ

- No access
- Public endpoint
- Private endpoint

**Private endpoints**

Private endpoint connections are associated with a private IP address within a Virtual Network. The list below shows all the private endpoint connections for this server. Note that private endpoint connections are defined at the server level and they provide access to all databases in the server. [Learn more](#)

+ Add private endpoint

Name	Subscription	Resource group	Region	Subnet
DP-300-SQL-Endpoint	AdventureWorks Production	DP-300-Lab02	eastus	Lab02-vnet / default

10. On the Additional Settings page, select the following options:

- Set Use existing data to **Sample**
- Ensure enable advanced data security is set to **Not now**

Basics Networking **Additional settings** Tags Review + create

Customize additional configuration parameters including collation & sample data.

**Data source**

Start with a blank database, restore from a backup or select sample data to populate your new database.

Use existing data \* ⓘ

- None
- Backup
- Sample

AdventureWorksLT will be created as the sample database.

**Database collation**

Database collation defines the rules that sort and compare data, and cannot be changed after database creation. The default database collation is SQL\_Latin1\_General\_CI\_AS. [Learn more](#)

Collation ⓘ

SQL\_Latin1\_General\_CI\_AS

**Advanced data security**

Protect your data using advanced data security, a unified security package including data classification, vulnerability assessment and advanced threat protection for your server. [Learn more](#)

Get started with a 30 day free trial period, and then 15 USD/server/month.

Enable advanced data security \* ⓘ

- Start free trial
- Not now

11. Click **Review + Create**
12. Review the settings before clicking **Create**
13. Once the deployment is complete, click the **Go to resource** button

## 15.2 Task 2: Enable All Azure Services access to new SQL Server

- From the SQL Database blade, click on the link for the Server name in the top section

AdventureWorksLT (dp300-lab-adw/AdventureWorksLT) ✖

**Overview**

Resource group (change) : DP-300-Lab02

Status : Online

Location : East US

Subscription (change) : Microsoft Azure Internal Consumption

Subscription ID : 6642bdd7-b570-4c10-bbf5-363ee6982e61

Tags (change) : Click here to add tags

Server name : dp300-lab-adw.database.windows.net

Elastic pool : No elastic pool

Connection strings : Show database connection strings

Pricing tier : Basic

Earliest restore point : No restore point available

2. On the SQL Server object's navigation blade, click **Firewalls and virtual networks** under Security

Security

- Advanced data security
- Auditing
- Firewalls and virtual networks**
- Private endpoint connections
- Transparent data encryption

3. Set Allow Azure services and resources to access this server to **Yes**

### dp300-lab-adw | Firewalls and virtual networks

SQL server

Save Discard + Add client IP

Deny public network access Yes **No**

**Setting to Yes** allows connections via approved private endpoint only and disables any existing firewall rules. [Learn more.](#)

Minimal TLS Version > 1.0 >1.1 >1.2

**You are setting the Minimal TLS Version property for all SQL Database and SQL Data Warehouse databases associated with the server. Any login attempts from clients using TLS version less than the Minimal TLS Version shall be rejected.**

Connection Policy Default Proxy Redirect

Allow Azure services and resources to access this server **Yes** No

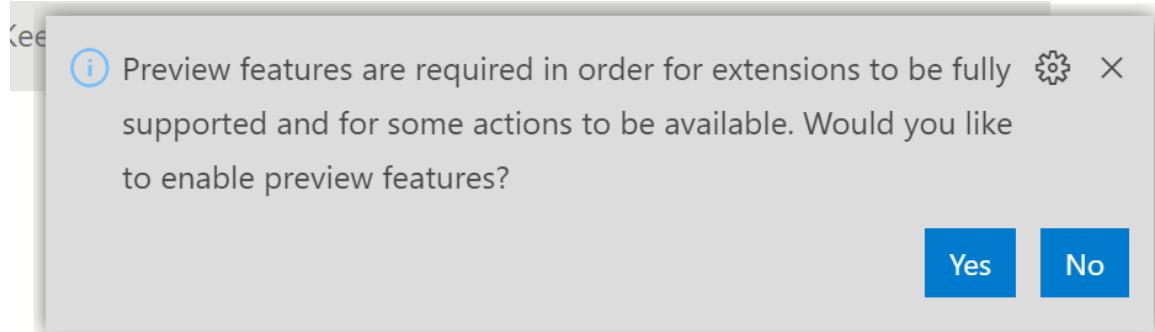
4. Click **Save**

## 16 Exercise 3: Connect to Azure SQL Database

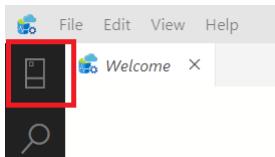
### 16.1 Task 1: Register Azure SQL Database Instance in Azure Data Studio

1. Launch Azure Data Studio (ADS) from the lab VM

- You may see this pop-up at initial launch of Azure Data Studio. If you receive it, click **Yes**



2. When Azure Data Studio opens, click the **Connections** button in Azure Data Studio's left sidebar, then the **Add a Connection** button



3. In the Connections sidebar, fill out the Connection Details section with connection information to connect to the SQL database created in the previous Exercise

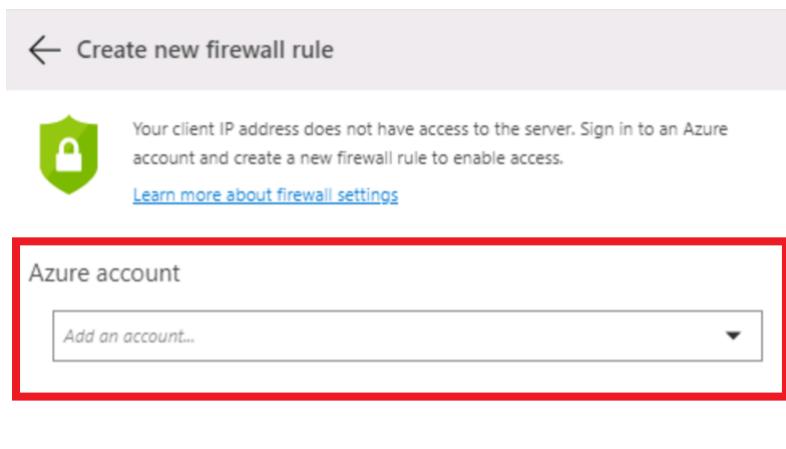
- Connection Type: **Microsoft SQL Server**
- Server: Enter the name of the SQL Server created in Exercise 2, Task 1. For example: **dp300-lab-xx.database.windows.net**  
[Note that you were asked to create a server name with your initials, instead of 'xx']
- Authentication Type: **SQL Login**
- User name: **dp300admin**
- Password: **dp300P@ssword!**
- Expand the Database drop-down to select **AdventureWorksLT**.

**NOTE:** You may be asked to add a firewall rule that allows your client IP access to this server.

If you do **not** receive a message about a firewall rule, just continue filling out the connection details.

If you are asked to add a firewall rule, do the following:

Click on Add account and login to your Azure account. On Create new firewall screen rule screen, click **OK**.



- Server group will remain on <**default**>
- Name (optional) can be populated with a friendly name of the database, if desired
- Review settings and click **Connect**

## Connection

[Recent Connections](#) [Saved Connections](#)

No recent connection

## Connection Details

Connection type

Microsoft SQL Server

Server

dp300-lab-adw.database.windows.net

Authentication type

SQL Login

User name

dp300admin

Password

\*\*\*\*\*

Remember password

Database

AdventureWorksLT

Server group

<Default>

Name (optional)

[Advanced...](#)

[Connect](#)

[Cancel](#)

4. Azure Data Studio will connect to the database and show some basic information about the database, plus a partial list of objects

Name	Schema	Type
BuildVersion	dbo	Table
ErrorLog	dbo	Table
Address	SalesLT	Table
Customer	SalesLT	Table
CustomerAddress	SalesLT	Table
Product	SalesLT	Table
ProductCategory	SalesLT	Table
ProductDescription	SalesLT	Table
ProductModel	SalesLT	Table
ProductModelProductDescription	SalesLT	Table
SalesOrderDetail	SalesLT	Table
SalesOrderHeader	SalesLT	Table

## 16.2 Task 2: Query Azure SQL Database with a SQL Notebook

1. In Azure Data Studio, connected to this lab's AdventureWorksLT database, click the **New Notebook** button

2. Click the **+Text** button to add a new text box in the notebook

Click on **+ Code** or **+ Text** to add a code or text cell

**Note:** Within the notebook you can embed plain text to explain queries or result sets.

3. Enter the text **Top Ten Customers by Order SubTotal**, making it Bold if desired

**Top Ten Customers by Order SubTotal**

4. Click the **+ Code** button to add a new cell at the end of the notebook to put a query in

The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with several buttons: 'Welcome' (with a cloud icon), a database connection icon labeled 'dp300-lab-adw.database.windows.net:AdventureWorksLT', a 'Notebook-0' button, and other options like 'Kernel: SQL', 'Attach To: dp300-lab-adw.database.w...', 'Trusted', and 'Run Cells'. A red box highlights the '+ Code' button. Below the toolbar, a code cell contains the text 'Top Ten Customers by Order SubTotal'. Inside the cell, there's a small blue play button icon followed by the number '1'. A red box highlights the entire code cell area.

5. Paste the following SQL statement into the new cell

```
select top 10 cust.[CustomerID], cust.[CompanyName], sum(sohead.[SubTotal]) as OverallOrderSubTotal  
from [SalesLT].[Customer] cust  
inner join [SalesLT].[SalesOrderHeader] sohead  
on sohead.[CustomerID] = cust.[CustomerID]  
group by cust.[CustomerID], cust.[CompanyName]  
order by [OverallOrderSubTotal] desc
```

6. Click on the blue circle with the arrow to execute the query. Note how the results are included within the cell with the query.

7. Click the **+ Text** button to add a new text cell.

8. Enter the text **Top Ten Ordered Product Categories**, making it Bold if desired

9. Click the **+ Code** button again to add a new cell, and paste the following SQL statement into the cell

```
select top 10 cat.[Name] as ProductCategory, sum(detail.[OrderQty]) as OrderedQuantity
```

```
from salesLT.[ProductCategory] cat  
inner join salesLT.[Product] prod  
on prod.[ProductCategoryID] = cat.[ProductCategoryID]  
inner join salesLT.[SalesOrderDetail] detail  
on detail.[ProductID] = prod.[ProductID]  
group by cat.[name]  
order by [OrderedQuantity] desc
```

10. Click on the blue circle with the arrow to execute the query

11. To run all cells in the notebook and present results, click the **Run Cells** button in the toolbar

```

1 select top 10 cat.[Name] as ProductCategory, sum(detail.[OrderQty]) as OrderedQuantity
2     from salesLT.[ProductCategory] cat
3         inner join salesLT.[Product] prod
4             on prod.[ProductCategoryID] = cat.[ProductCategoryID]
5                 inner join salesLT.[SalesOrderDetail] detail
6                     on detail.[ProductID] = prod.[ProductID]
7                         group by cat.[name]
8                             order by [OrderedQuantity] desc
9

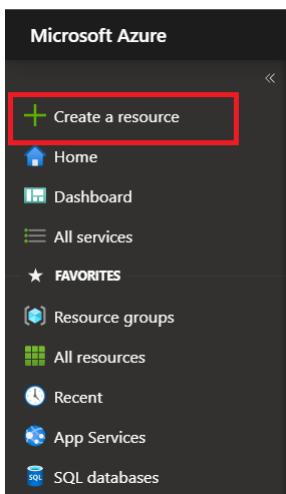
```

12. Within Azure Data Studio save the notebook from File menu (either Save or Save As) to the D:\Labfiles\Deploy Azure SQL Database (this folder already exists on the VM) directory. Close the tab for the Notebook from inside of Azure Data Studio. From the File Menu, select Open File, and open the notebook you just saved. Observe that query results were saved along with the queries in the notebook.

## 17 Exercise 4: Deploy an Azure Database for PostgreSQL Database

### 17.1 Task 1: Deploy a PostgreSQL Database

1. From the Azure Portal, click on **+ Create a Resource** at the top of the left side navigation bar



2. Search for “postgresql” in the search box at the top, then click **Azure Database for PostgreSQL** in the results

Home > New > Marketplace

## Marketplace

Showing All Results

**Categories**

- Get Started
- AI + Machine Learning
- Analytics
- Blockchain
- Compute
- Containers
- Databases

**Azure Database for PostgreSQL**  
Microsoft  
Managed PostgreSQL database service for app developers.

**PostgreSQL Container Image**  
Bitnami  
Up-to-date, customizable, and secure.

**PostgreSQL HA Container Image**  
Bitnami  
Up-to-date, customizable, and secure.

3. Click the **Create** button
4. Click **Create** for the Single Server option

Select **Azure Database for PostgreSQL deployment option**

Microsoft

#### How do you plan to use the service?



##### Single server

Best for broad range of traditional transactional workloads.

Enterprise ready, fully managed community PostgreSQL server with up to 64 vCores, optional geospatial support, full-text search and more.

**Create**

[Learn more](#)



##### Hyperscale (Citus) server group

Best for ultra-high performance and data needs beyond 100GB.

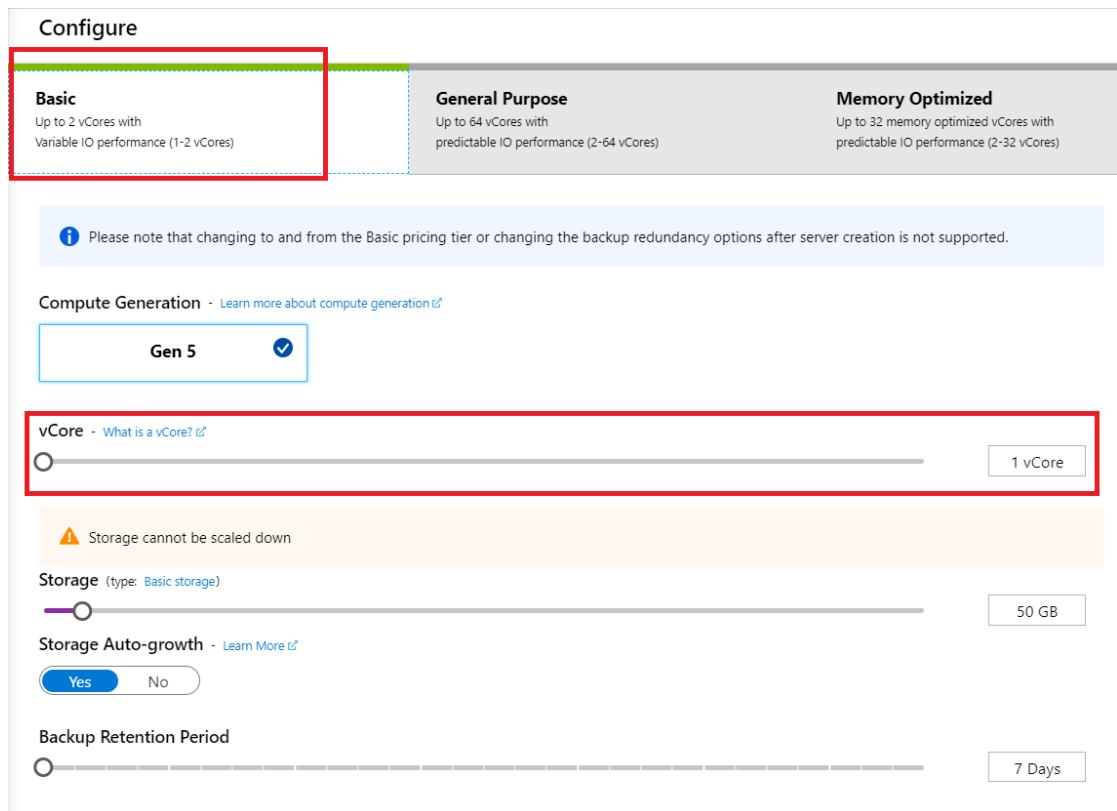
Ideal for multi-tenant applications and real-time analytical workloads that need sub-second response. Supports both transactional/operational workloads and hybrid transactional analytics workloads.

**Create**

[Learn more](#)

5. Complete the Single Server Basics screen with the following inputs:

- Subscription: Select the lab subscription
- Resource Group: **DP-300-Lab02** (the RG created in Exercise 1)
- Server Name: **dp300-lab02-\*\*\*\*<your initials>** (server name must be unique)
- Data source: **None**
- Location: Select the Region closest to you, which has been used throughout this lab
- Version: **10 or 11**
- Click the **Configure Server** link under Compute + Storage
  - Click the **Basic** tab
  - Slide the vCore slider all the way to the left to select **1 vCore**



– Click **OK**

- Admin username: **dp300admin**
- Password: **dp300P@ssword!**
- Confirm password: **dp300P@ssword!**

**Single server**

Microsoft

**Basics** Tags Review + create

Create an Azure Database for PostgreSQL server. [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ AdventureWorks Production

Resource group \* ⓘ DP-300-Lab02

[Create new](#)

**Server details**

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name \* ⓘ dp300-lab02-doc ✓

Data source \* ⓘ  None  Backup

Location \* ⓘ (US) East US ✓

Version \* ⓘ 10 ✓

Compute + storage ⓘ

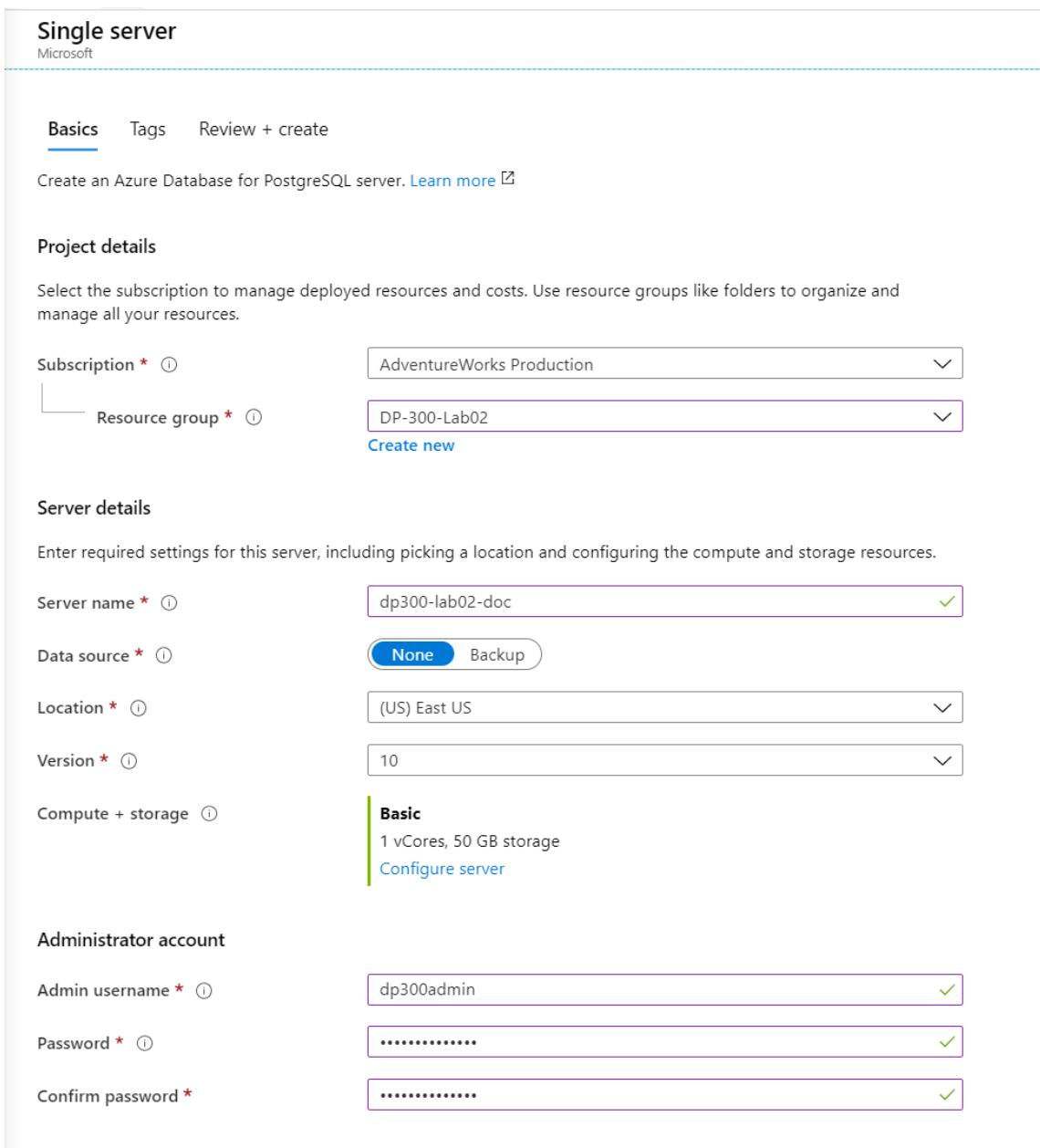
**Basic**  
1 vCores, 50 GB storage  
[Configure server](#)

**Administrator account**

Admin username \* ⓘ dp300admin ✓

Password \* ⓘ ..... ✓

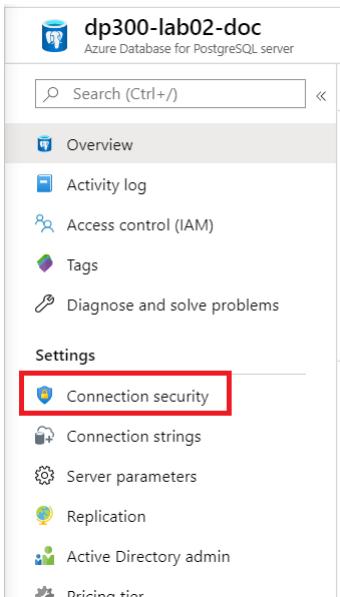
Confirm password \* ⓘ ..... ✓



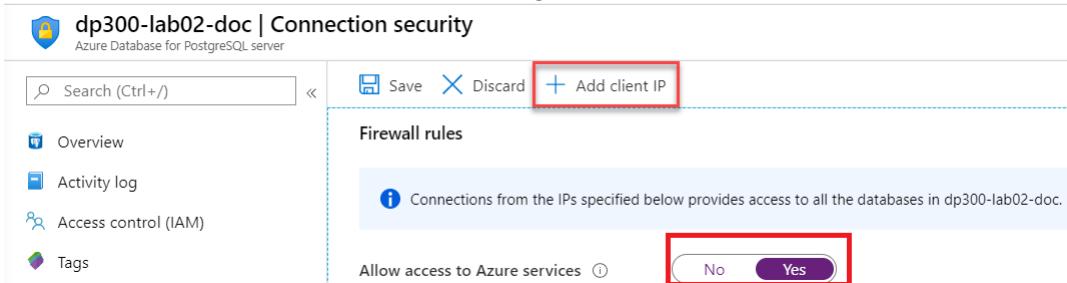
6. Click **Review + Create**
7. Review the settings and click **Create**
8. When the deployment is complete, click **Go to resource**

## 17.2 Task 2: Enable All Azure Services Access to PostgreSQL Database

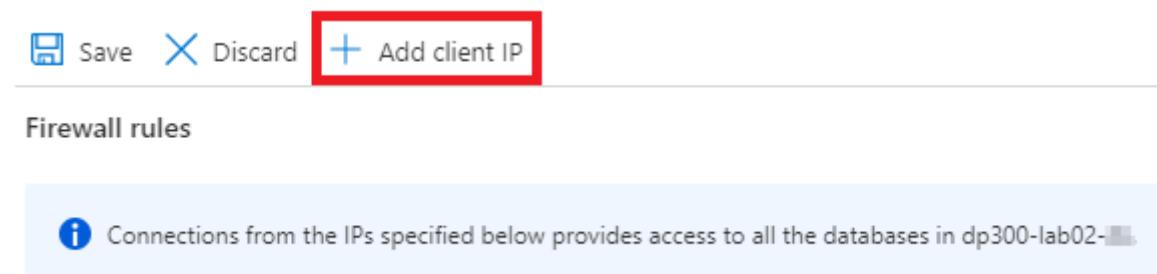
1. Click on the **Connection security** button under Settings in the database's sidebar



2. Slide the Allow access to Azure services setting to **Yes**. Click **+ Add client IP**



3. Click on the **Add client IP** button at the top right to add your client IP to the firewall.

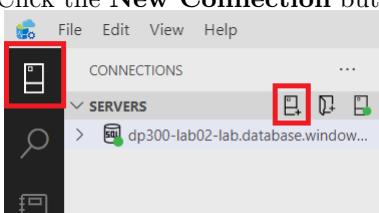


4. Click **Save** at the top left.

## 17.3 Task 3: Connect to the PostgreSQL Database with Azure Data Studio

**Note:** With the PostgreSQL extension installed to Azure Data Studio, PostgreSQL can be queried. This extension is pre-installed on the lab VM.

1. In Azure Data Studio on the lab VM, ensure the Connections sidebar is expanded—Click the **Connections** button in the left nav bar, if not.
2. Click the **New Connection** button



3. In the Connection Details sidebar, enter the following information to connect to the previous Task's PostgreSQL database:

- Connection Type: **PostgreSQL**
- Server: Enter the name of the PostgreSQL Server you created in Task 1. For example: **dp300-lab02-xx.postgres.database.azure.com** (You can find the full name of the server in the Overview pane in the portal. You'll probably need to replace xx with the specific letters you added. )
- Authentication Type: **Password**
- User name: **dp300admin@dp300-lab02-xx** (note that the user name must include the first part of the hostname that you specified above; use your suffix instead of 'xx'. The admin username can also be found on the Overview pane in the portal.)
- Password: **dp300P@ssword!**
- Database Name will remain on <b><default></b>
- Server group will remain on <b><default></b>
- Name (optional) can be populated with a friendly name of the database, if desired

4. Review settings

Connection Details

Connection type: PostgreSQL

Server name: dp300-lab02-xx.postgres.database.azure.com

Authentication type: Password

User name: dp300admin@dp300-lab02-xx

Password:

Remember password

Database name: <Default>

Server group: <Default>

Name (optional):

Advanced...

Connect Cancel

5. Click **Connect**

6. Note Azure Data Studio's connection to the PostgreSQL database

Home > dp300-lab02-doc.database.windows.net

Home

New Query New Notebook Refresh Learn More

Search

Search databases

postgres  
azure\_maintenance  
azure\_sys

**17.4 lab: title: 'Lab 3 – Implement a Secure Environment' module: 'Implement a Secure Environment'**

## 18 Lab 3 – Implement a Secure Environment

**Estimated Time:** 60 minutes

**Prerequisites:** An Azure SQL server you created in the lab for Module 2. Azure Active Directory access in the subscription.

**Lab files:** The files for this lab are in the D:\Labfiles\Secure Environment folder.

## 19 Lab overview

The students will take the information gained in the lessons to configure and subsequently implement security in the Azure Portal and within the AdventureWorks database.

## 20 Lab objectives

After completing this lab, you will be able to:

1. Configure an Azure SQL Database Firewall
2. Authorize Access to Azure SQL Database with Azure Active Directory
3. Enable Advanced Threat Protection for Azure SQL Database
4. Configure Data Classification for Azure SQL Database
5. Manage access to database objects

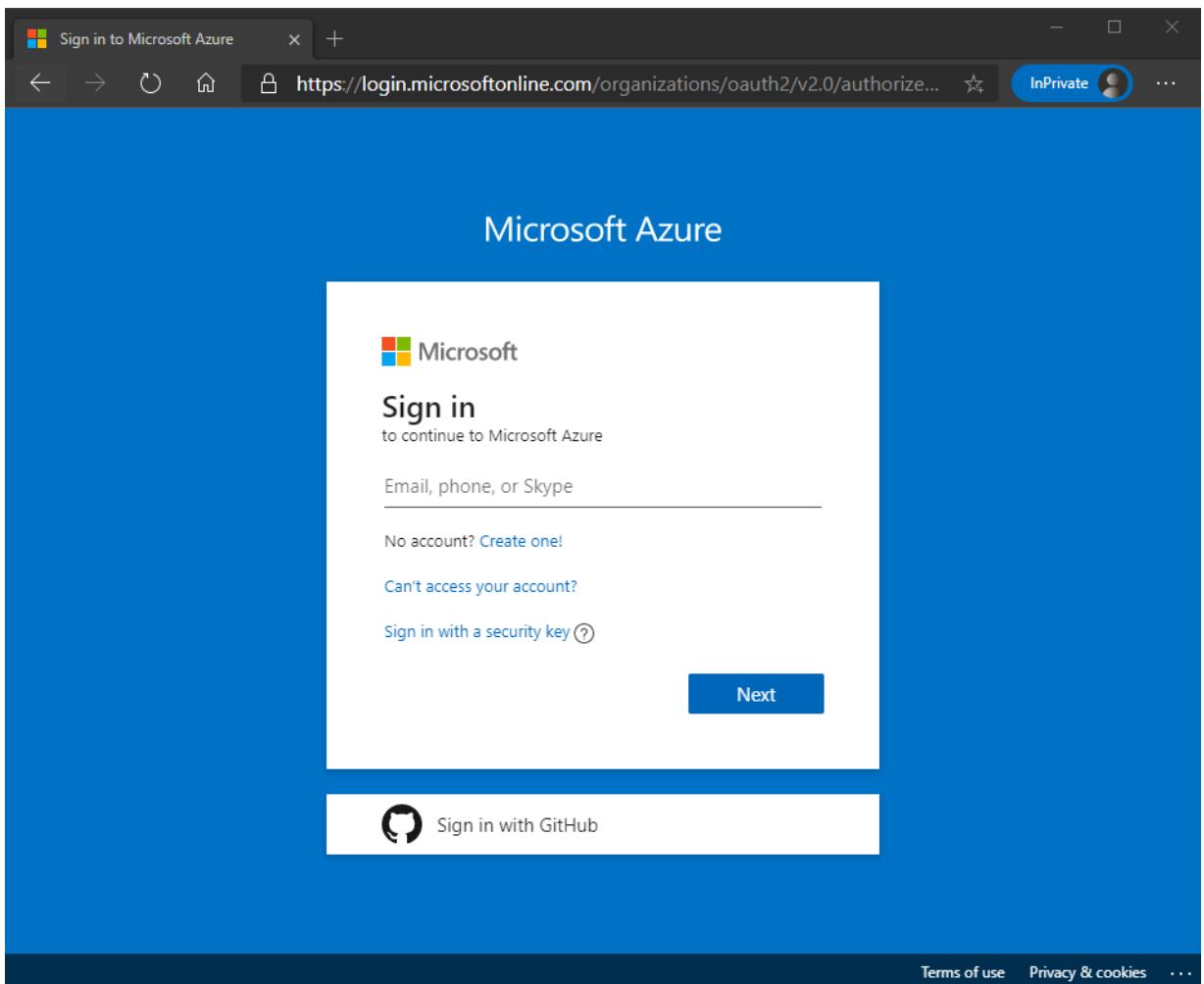
## 21 Scenario

You have been hired as a Senior Database Administrator help ensure the security of the database environment. These tasks will focus on Azure SQL Database.

**Note:** The exercises ask you to copy and paste T-SQL code. Please verify that the code has been copied correctly, with the proper line breaks, before executing the code.

### 21.1 Exercise 1: Configure an Azure SQL Database Firewall and connect to a new database

1. From the lab virtual machine, start a browser session and navigate to <https://portal.azure.com>. Provide appropriate credentials.



2. In the search bar at the top of the Azure Portal, type SQL. The SQL servers icon will appear. Click on SQL servers. Click on the server name to be taken to the detail page for the server you created in Lab 2

A screenshot of the Microsoft Azure portal showing the "SQL servers" list. The search bar at the top contains "Search resources, services, and docs (G+/-)". The main heading is "SQL servers". Below the heading are filter options: "Subscription == 2 of 11 selected", "Resource group == all", "Location == all", and "Add filter". A table lists one record: "dp300-lab-adw" (Status: Available, Resource group: DP-300-Lab02, Location: East US). The table has columns for Name, Status, Resource group, and Location.

3. In the detail screen for your SQL server, move your mouse to right of the server name, and click copy to clipboard button as shown below.

A screenshot of the Microsoft Azure portal showing the detail screen for the SQL server "dp300-lab-adw". The left sidebar includes "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", and "Settings". The main area shows "Essentials" details: Resource group: DP-300-Lab02, Status: Available, Location: West US 2, Subscription: Azure Pass - Sponsorship, Subscription ID: 12345678901234567890123456789012, and Tags: Click here to add tags. On the far right, there is a "Copy to clipboard" button with a tooltip. The status bar at the bottom says "dp300-lab-adw.database.windows.net".

4. Click on Show firewall settings (above the server name that you just copied). Click on + Add client IP as highlighted below and then click Save.

Deny public network access ⓘ Yes No

Minimal TLS Version ⓘ > 1.0 > 1.1 > 1.2

Connection Policy ⓘ Default Proxy Redirect

Allow Azure services and resources to access this server Yes No

**i** Connections from the IPs specified below provides access to all the databases in dp300-lab-adw.

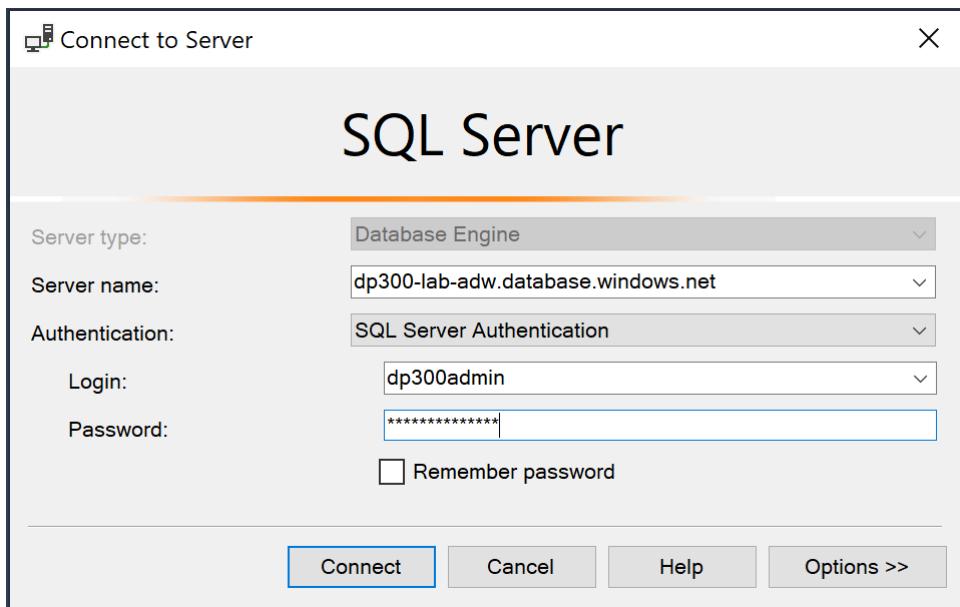
Rule name	Start IP	End IP
ClientIPAddress_2020-6...	116.88.72.234	116.88.72.234

**i** Connections from the VNET/Subnet specified below provides access to all databases in dp300-lab-adw.

This will allow you to connect to your Azure SQL Database server using SQL Server Management Studio or any other client tools. Make note of your client IP address, you will use it later in this task.

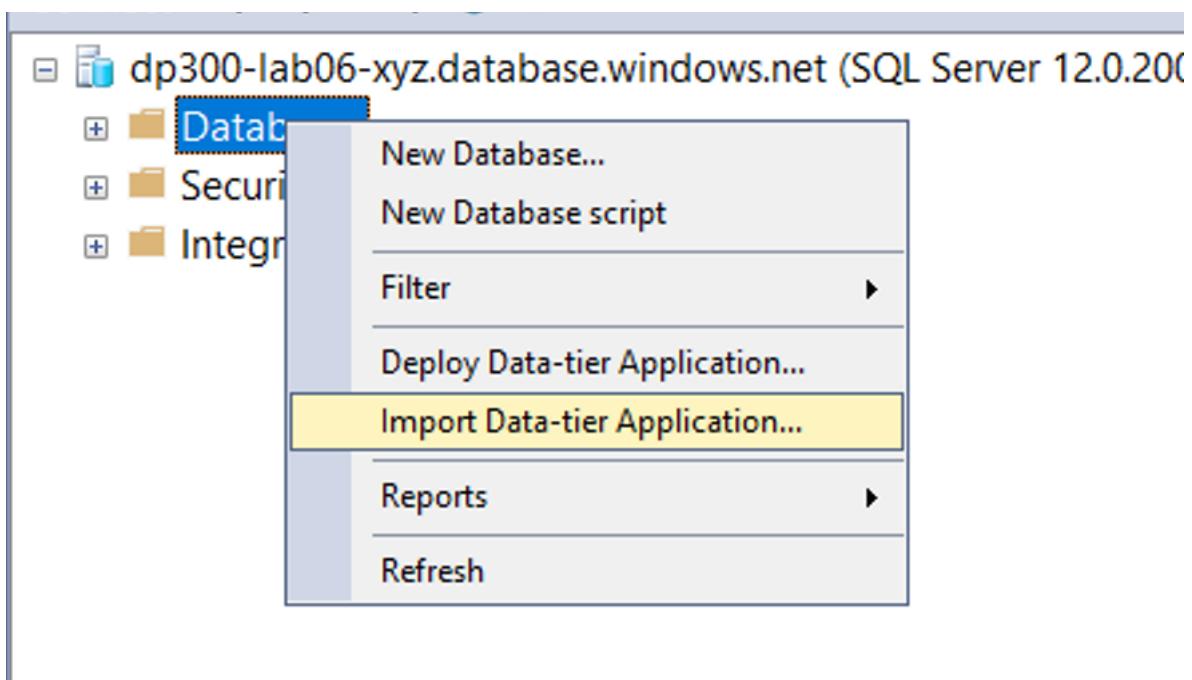
5. Open SQL Server Management Studio on the lab VM, by navigating to Microsoft SQL Server Tools 18 > SQL Server Management Studio from the Start menu. Paste in the name of your Azure SQL database server and login with the credentials you created in Lab 2:

- Server name:
- Authentication: SQL Server Authentication
- Server admin login: **dp300admin**
- Password: **dp300P@ssword!**

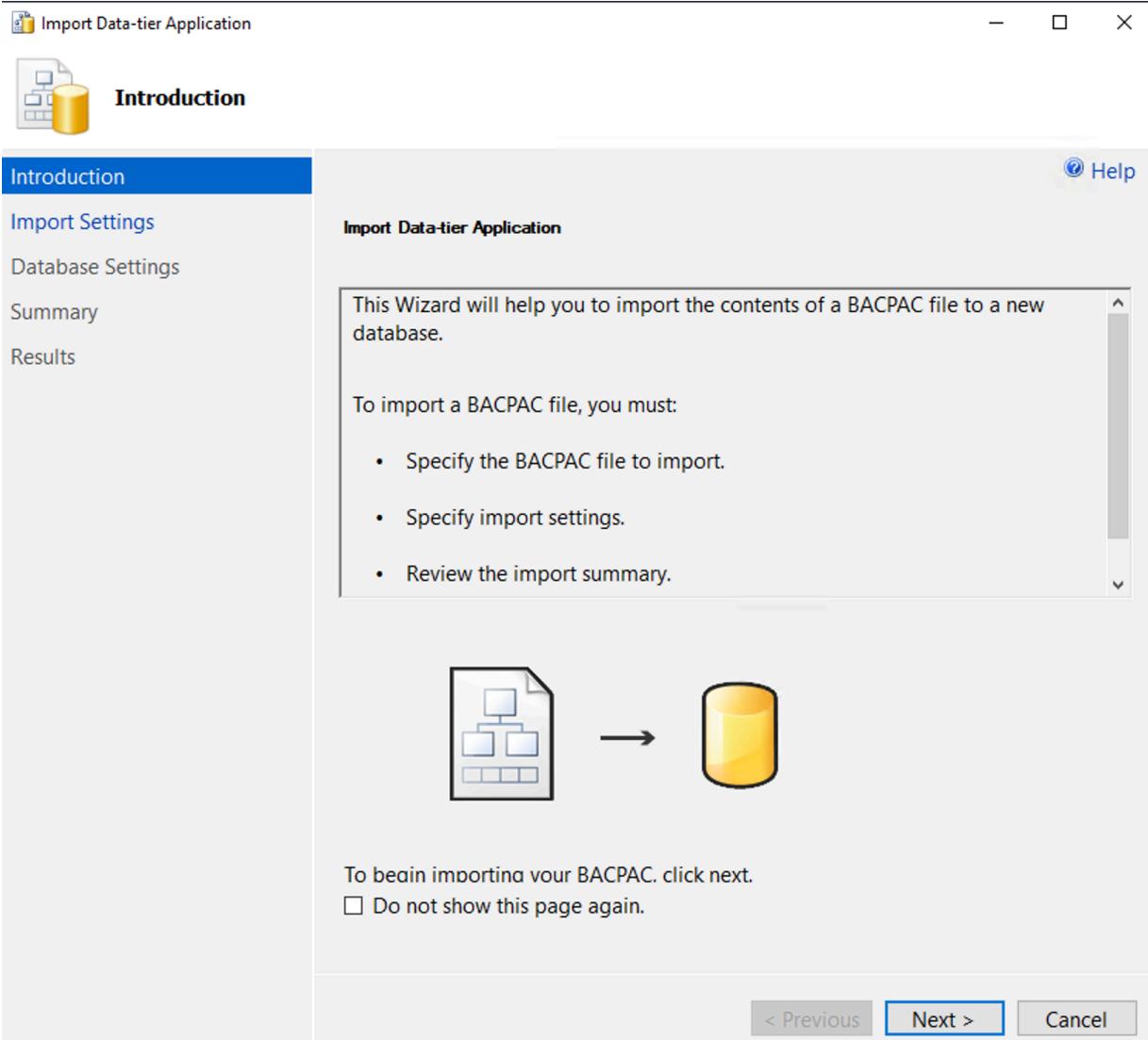


Click Connect.

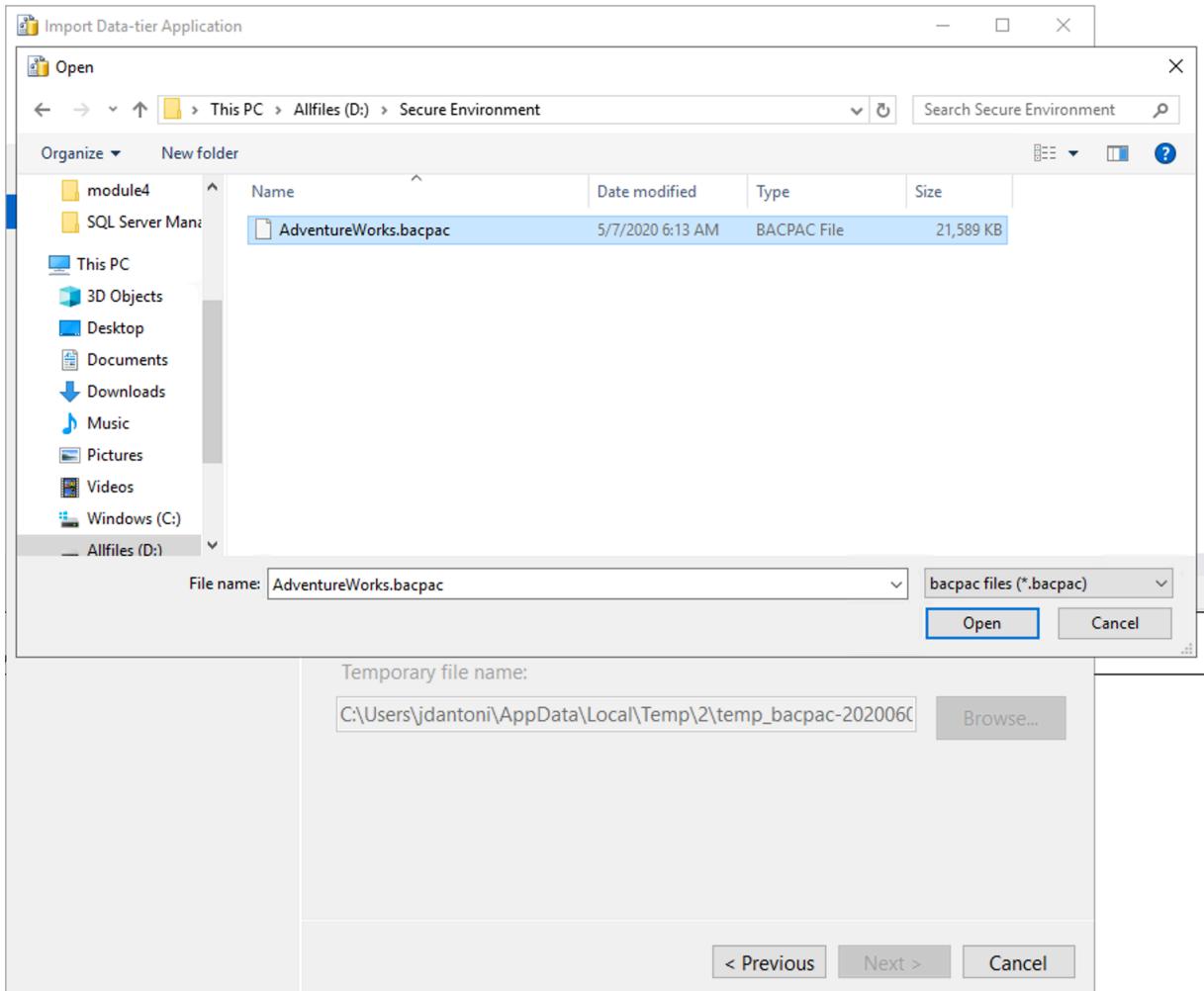
6. In Object Explorer expand the server node, and right click on databases. Click Import a Data-tier Application.

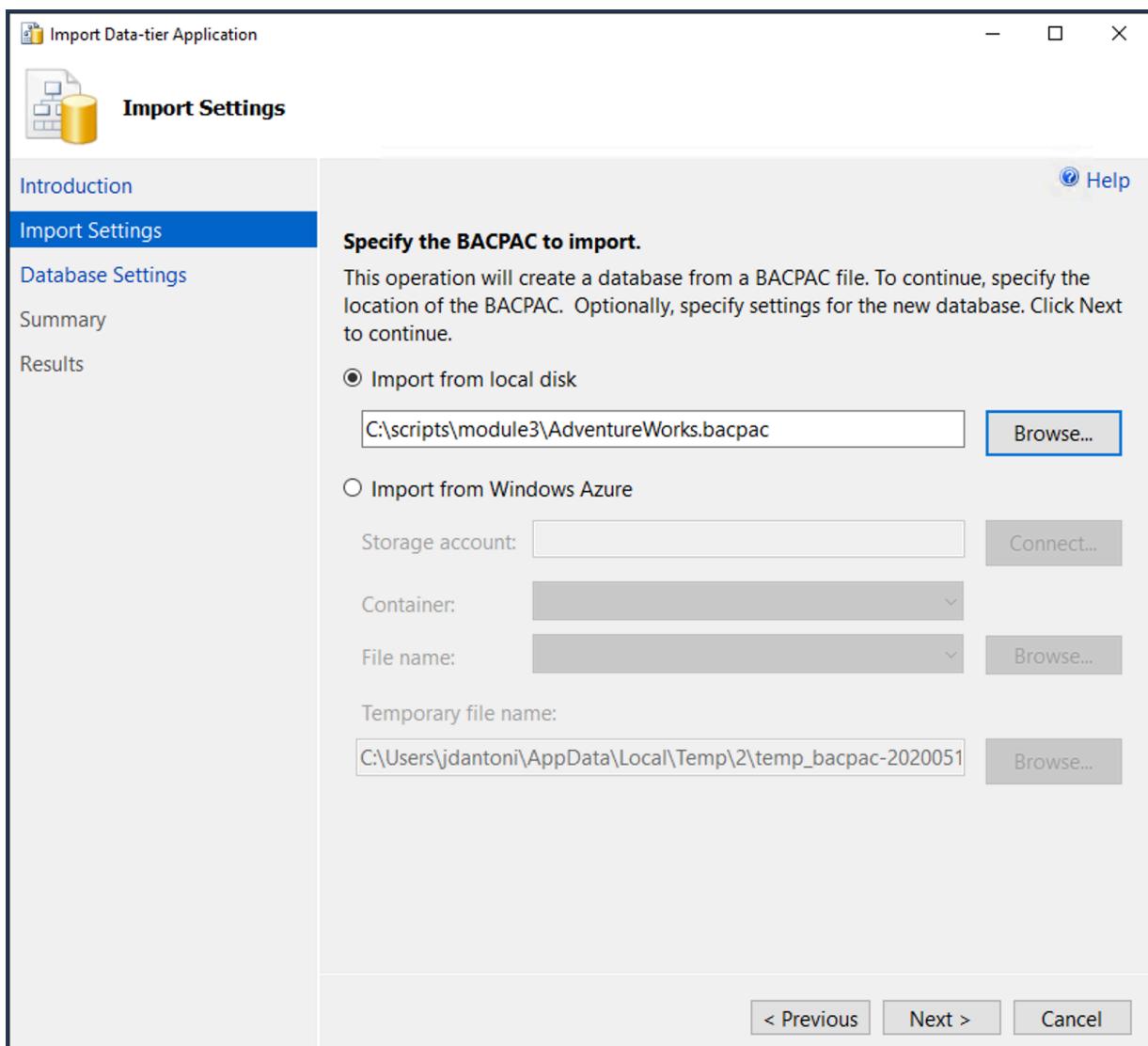


7. In the Import Data Tier Application dialog, click Next on the first screen.

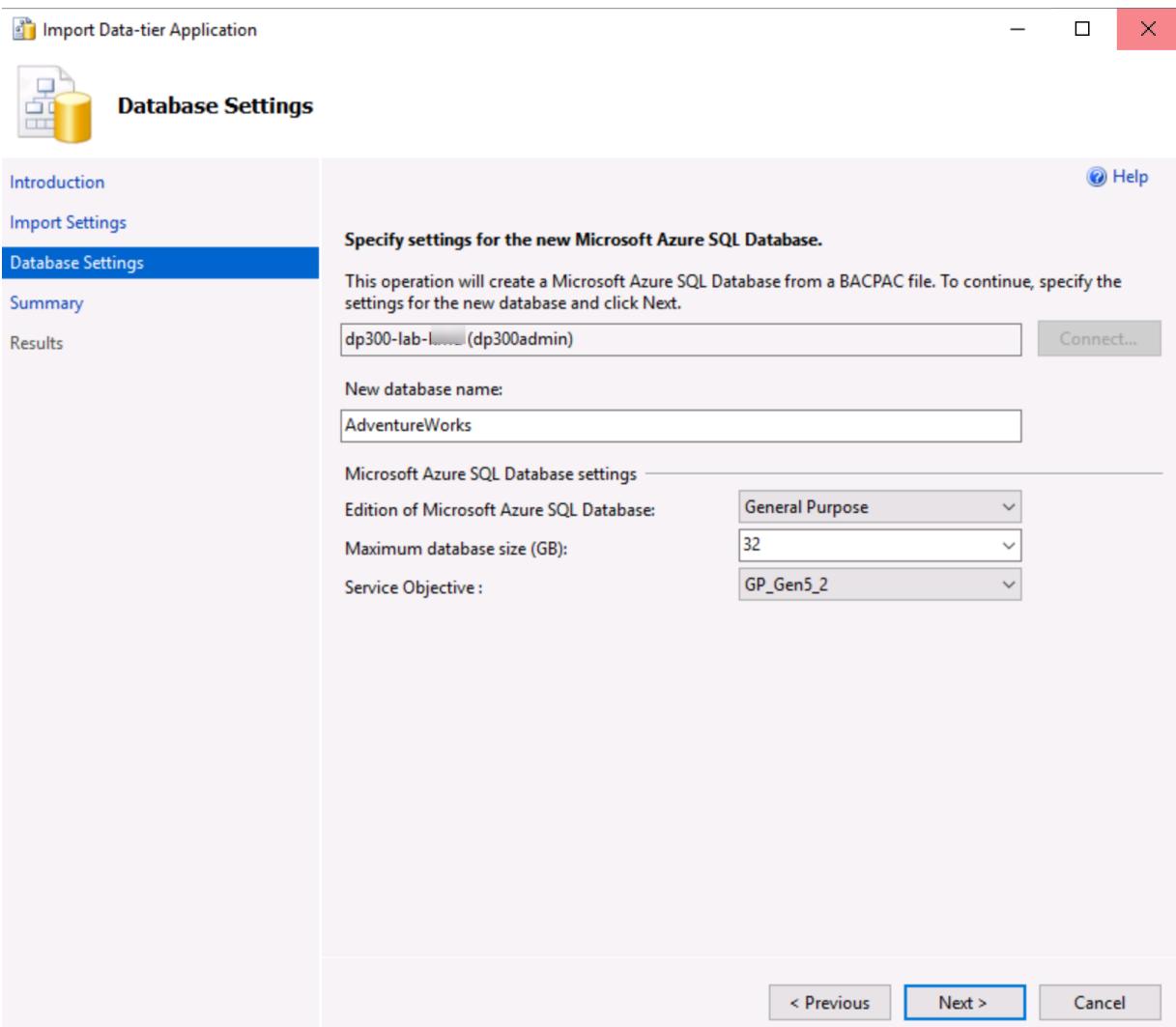


8. In the Import Settings screen, click Browse and navigate to D:\Labfiles\Secure Environment folder and click on the AdventureWorks.bacpac file and click open. Then in the Import Data-tier application screen click Next.





9. On the database settings screen, change the edition of Azure SQL Database to General Purpose. Change the Service Objective to **GP\_Gen5\_2** and click next.



10. Click **Next** and then on the Summary screen click **Finish**. When your import completes you will see the re-

Import Data-tier Application

Results

Introduction  
Import Settings  
Database Settings  
Summary  
**Results**

Help

**Operation Complete**

Summary:

Name	Result
Importing database	Success
Creating deployment plan	Success
Initializing deployment	Success
Verifying deployment plan	Success
Analyzing deployment plan	Success
Importing package schema and data into database	Success
Updating database	Success
Importing data	Success
Processing Import.	Success
Disabling indexes.	Success
Disabling index 'PK_DatabaseLog_DatabaseLogID'.	Success
Disabling index 'AK_Department_Name'.	Success
Disabling index 'AK_Employee_LoginID'.	Success
Disabling index 'AK_Employee_NationalIDNumber'.	Success
Disabling index 'AK_Employee_rowguid'.	Success
Disabling index 'IX_Employee_OrganizationLevel_OrganizationNo...'.	Success
Disabling index 'IX_Employee_OrganizationNode'.	Success
Disabling index 'IX_EmployeeDepartmentHistory_DepartmentID'.	Success
Disabling index 'IX_EmployeeDepartmentHistory_ShiftID'.	Success
Disabling index 'IX_JobCandidate_BusinessEntityID'	Success

< Previous    Next >    Close

sults below.

11. In Object Explorer right-click on AdventureWorks and then click on new query.

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

New Query MDX DMX XMLA DAX

AdventureWorks

Object Explorer

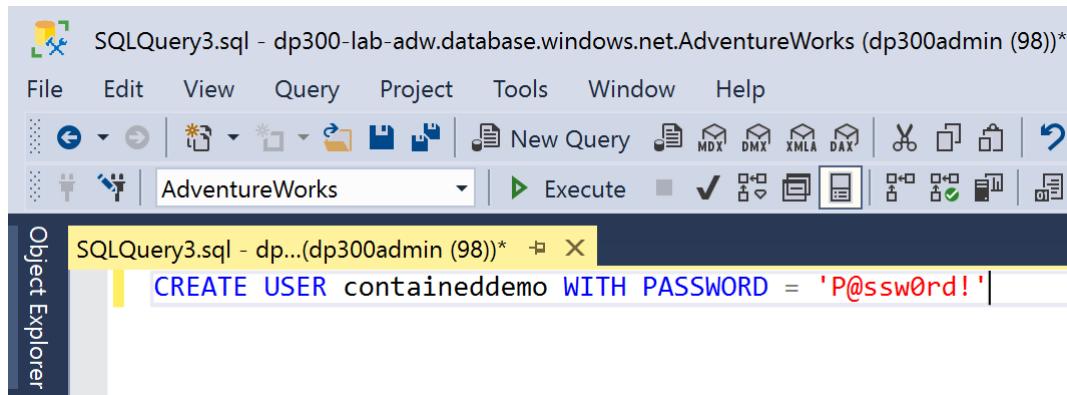
dp300-lab-adw.database.windows.net (SQL Server 12.0.2000.8 - dp300adm)

- Databases
  - System Databases
    - AdventureWorks
    - AdventureWorksLT
  - Security
  - Integration Services Catalogs

12. Execute the following T-SQL query by pasting the text into your query window and replacing 192.168.1.1. with your client IP address from Step 4. Click execute or press F5.

```
EXECUTE sp_set_database_firewall_rule @name = N'ContosoFirewallRule',
@start_ip_address = '192.168.1.1', @end_ip_address = '192.168.1.1'
```

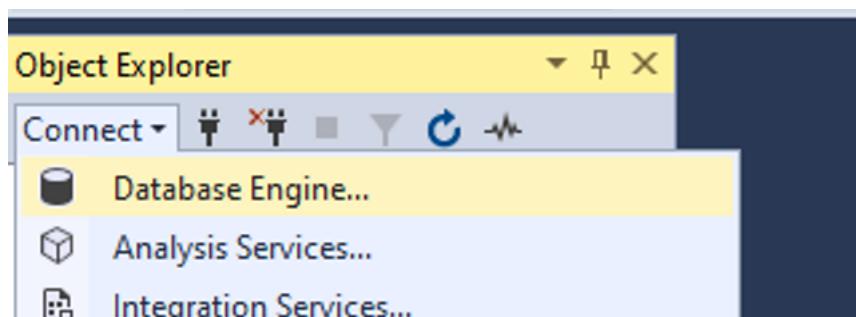
13. Next you will create a contained user in the AdventureWorks database. Click New Query and execute the following T-SQL. Ensure that you are still using the AdventureWorks database. If you see master in the database name box below, you can pull down and switch to AdventureWorks.



Click execute to run this command. This command creates a contained user within the AdventureWorks

```
CREATE USER containeddemo WITH PASSWORD = 'P@ssw0rd!'
```

14. Navigate to Object Explorer in SSMS and click on connect and then Database Engine

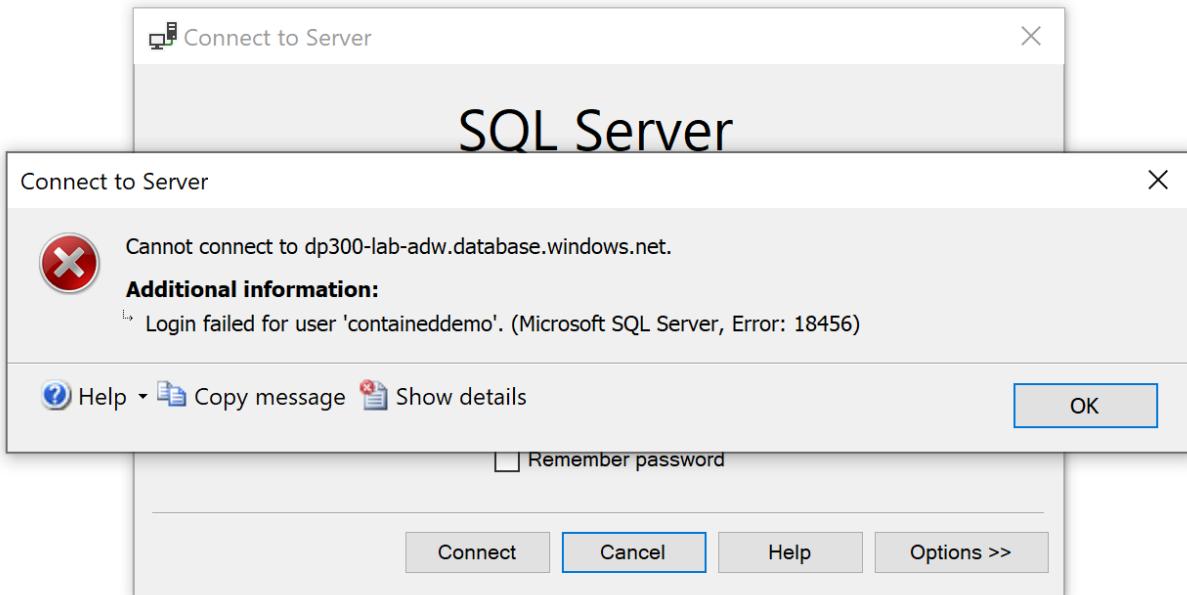


15. Attempt to connect with the credentials you created in step 13. You will need to use the following information:

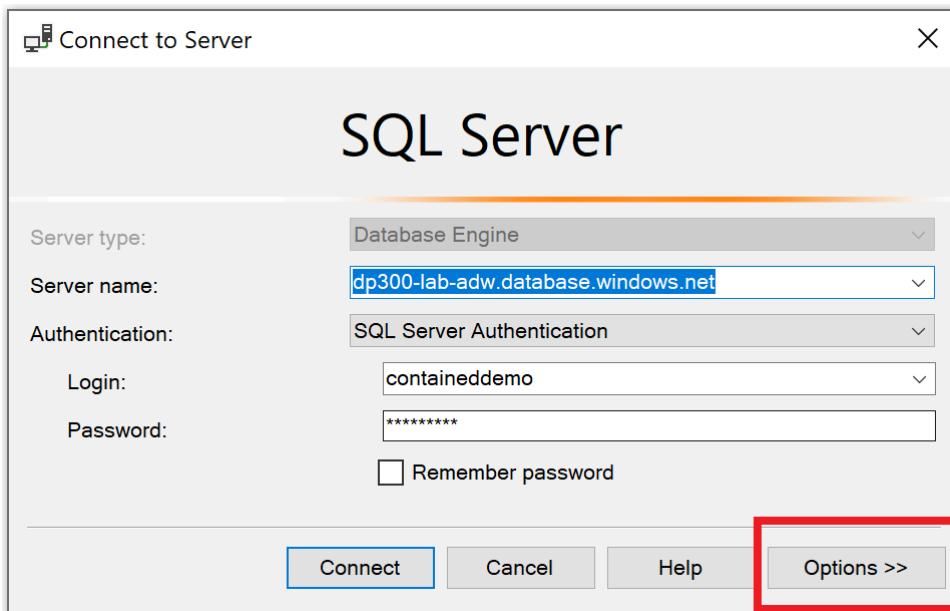
- **Login:** containeddemo
- **Password:** P@ssw0rd!

Click Connect.

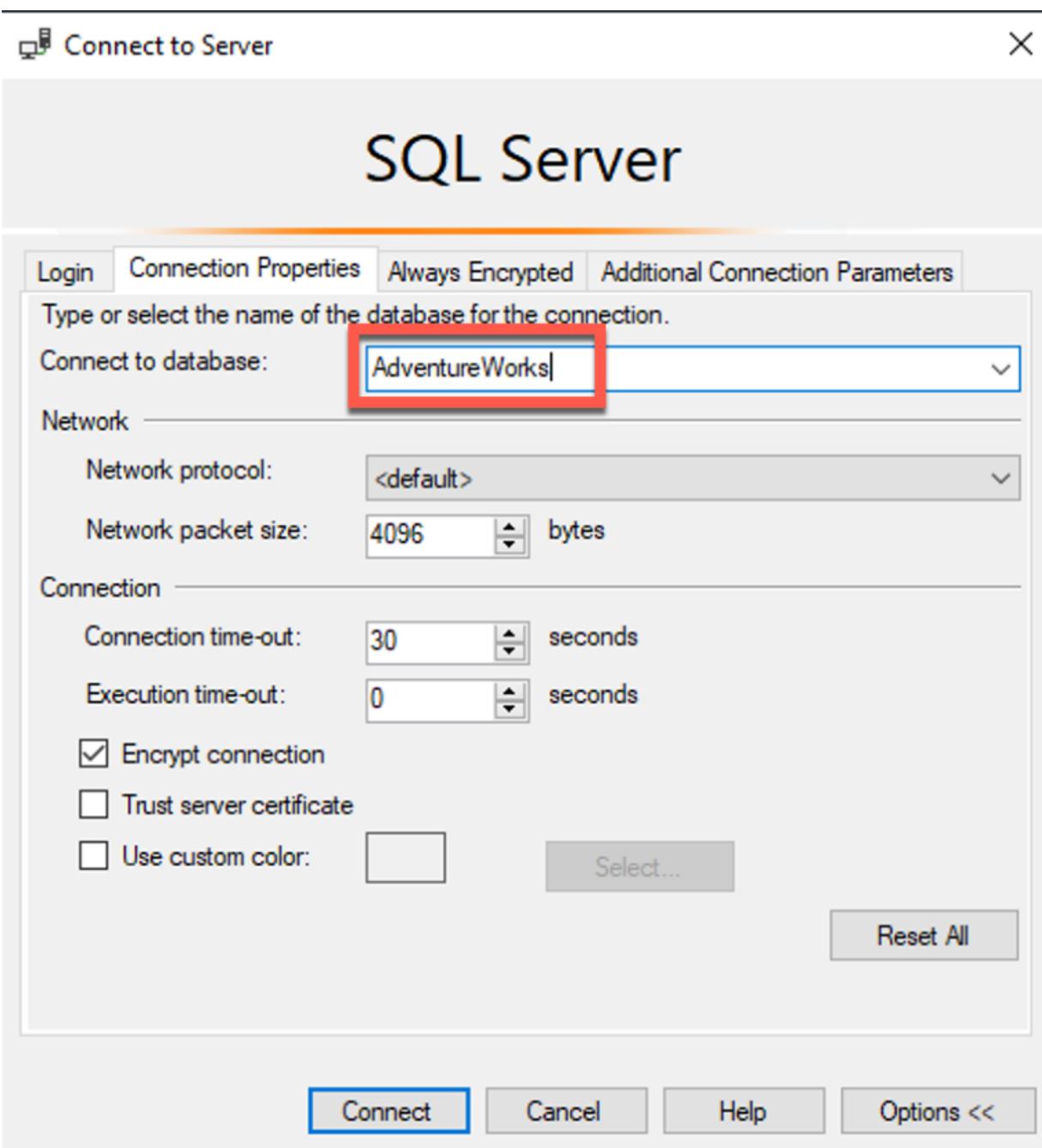
You will see the following error.



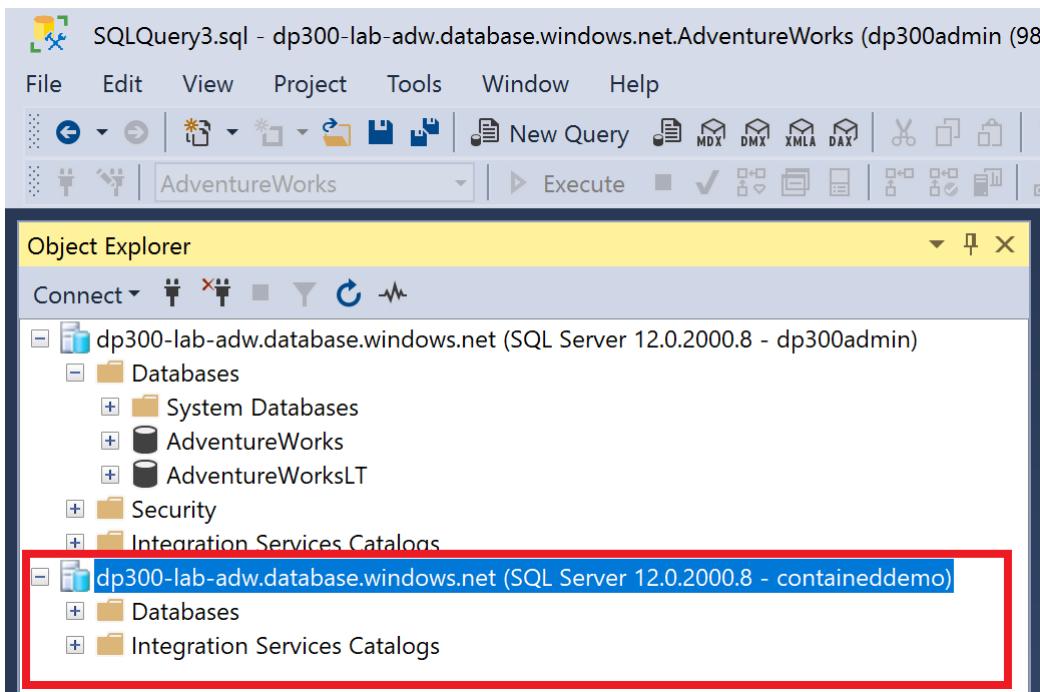
This error is generated because the connection attempted to login to the master database and not AdventureWorks where the user was created. Change the connection context by clicking OK to exit the error message and then clicking on Options >> in the Connect to Server dialog box as shown below.



16. On the connection options tab, type the database name AdventureWorks.



17. Click connect from that screen. This connection bypasses the master database and logs you directly into AdventureWorks, which is the only database to which the newly created user has access.



## 21.2 Exercise 2: Authorize Access to Azure SQL Database with Azure Active Directory

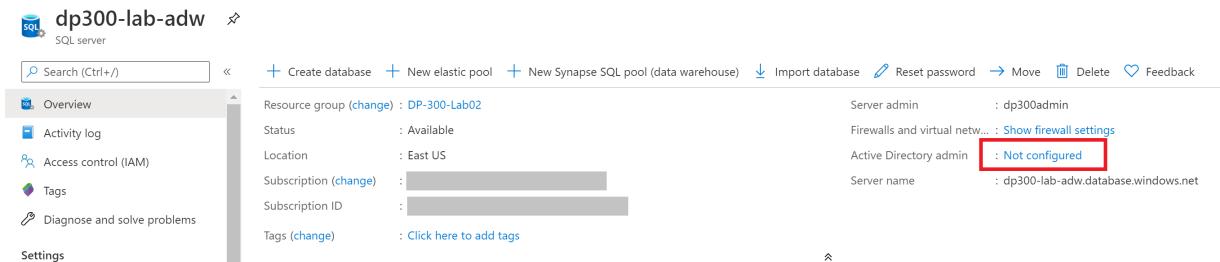
1. Navigate to the Azure Portal, and click on your user name in the top right corner of the screen.



Make note of the user name.

**Note:** A Microsoft account (a user account from Outlook, Gmail, Hotmail or Yahoo, for example) is not supported for the Azure Active Directory administrator for Azure SQL Database. As a workaround, you can create an Azure Active Directory group named DBA and add your user account to it. Alternatively, you can skip Exercise 2.

2. In the Azure Portal navigate to your Azure SQL Database server dp300-lab-xx and click on Not Configured next to Active Directory Admin.



On the next screen, click set admin.

Azure Active Directory authentication allows you to centrally manage identity and access to your Azure SQL Database V12.  
[Learn more](#)

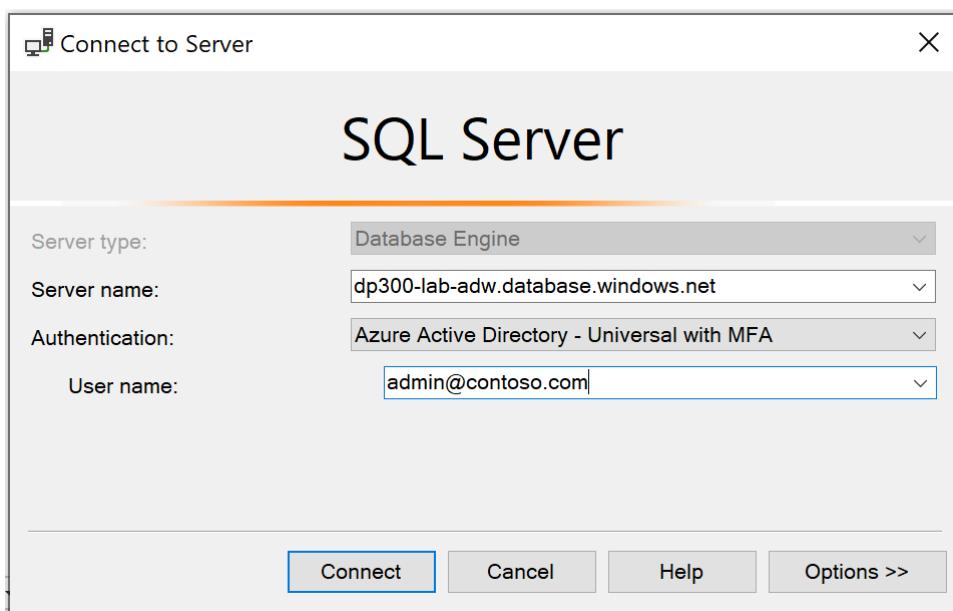
Active Directory admin ⓘ  
No Active Directory admin

In the Set admin screen, search for your username. When you have found it, click on it to highlight the username, and then click Select. You will be returned to the above Active Directory Admin screen. Click save to complete the process. This will make your username the Azure Active Directory admin for the server as shown below

Azure Active Directory authentication allows you to centrally manage identity and access to your Azure SQL Database V12.  
[Learn more](#)

Active Directory admin ⓘ  
d...@contoso.com

3. Open SQL Server Management Studio and click Connect > Database Engine. In the server name enter the name of your server. Change the authentication type to Azure Active Directory Universal with MFA.



You will be prompted to enter your Azure Active Directory password, and will you click **Connect**, you'll be logged in to your database.

### 21.3 Exercise 3: Enable Azure Defender for SQL and Data Classification

- From the main blade of your Azure SQL server navigate to the Security section and click Security Center.  
[Note that Azure Defender for SQL was formerly called Advanced data security, and the screenshots still reflect the former name.]

The screenshot shows the Azure portal interface for a SQL server named 'dp300-lab-adw'. The left sidebar lists various management options: Manage Backups, Active Directory admin, SQL databases, SQL elastic pools, Deleted databases, Import/Export history, DTU quota, Properties, Locks, and Export template. Below this, a 'Security' section is visible, containing 'Advanced data security'. This last item is highlighted with a red rectangular box.

Click the radio button next to **NOT CONFIGURED** under AZURE DEFENDER FOR SQL, then select Enable Azure Defender for SQL. Then select **Configure** next to Azure Defnader for SQL: Enabled at the subscription-level: Click on Storage account.

## ADVANCED DATA SECURITY

 ON  OFF



Advanced Data Security costs 15 USD/server/month. It includes Data Discovery & Classification, Vulnerability Assessment and Advanced Threat Protection. We invite you to a trial period for the first 30 days, without charge.

## VULNERABILITY ASSESSMENT SETTINGS

Subscription >

Contoso Ltd

Storage account >

dp300module7sacct

Periodic recurring scans 

 ON  OFF

Send scan reports to 

Email addresses 

Also send email notification to admins and subscription owners 

## ADVANCED THREAT PROTECTION SETTINGS

Send alerts to 

Email addresses 

Also send email notification to admins and subscription owners 

Advanced Threat Protection types >

All

 [Enable Auditing for better threats investigation experience](#)

You may be prompted to create a storage account, unless there already are storage accounts for your subscription and region. You can choose an existing storage account or click + Create new. Enter a name for your storage account and click OK.

## Create storage account

X

Name \*

 .core.windows.net

Account kind ⓘ

Storage (general purpose v1)

Performance ⓘ

Standard  Premium

Replication ⓘ

Locally-redundant storage (LRS)

OK

Toggle the switch for Periodic recurring scans to ON. Choose an email account to send scan reports and alerts to. Uncheck the box that says “Also send email notification to admins and subscription owners.”

## ADVANCED DATA SECURITY

ON

OFF



Advanced Data Security costs 15 USD/server/month. It includes Data Discovery & Classification, Vulnerability Assessment and Advanced Threat Protection. We invite you to a trial period for the first 30 days, without charge.

## VULNERABILITY ASSESSMENT SETTINGS

Subscription >

Microsoft Azure Internal Consumption

Storage account >

sqldbdiag197

### Periodic recurring scans

ON

OFF

### Send scan reports to

admin@contoso.com 

Also send email notification to admins and subscription owners 

## ADVANCED THREAT PROTECTION SETTINGS

### Send alerts to

admin@contoso.com 

Also send email notification to admins and subscription owners 

### Advanced Threat Protection types >

All

 [Enable Auditing for better threats investigation experience](#)

Click on Advanced Threat Protection types and review the selections. Leave all of the boxes checked and click **Save** at the top of the screen.

2. Navigate to the AdventureWorks database in the Azure portal by scrolling down in the Overview screen for Azure SQL server and clicking on the database name

dp300-lab-adw

SQL server

Search (Ctrl+ /)

+ Create database + New elastic pool + New Synapse SQL pool (data warehouse) ↴ Import database ⚙ Reset password →

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Quick start

Failover groups

Manage Backups

Active Directory admin

SQL databases

SQL elastic pools

Deleted databases

Import/Export history

DTU quota

**Advanced Data Security Free Trial**

Your free trial will expire in 30 days

Available resources

Filter by name

All types

2 databases

Name	Type	Status
AdventureWorks	SQL database	Online
AdventureWorksLT	SQL database	Online

3. Navigate to the Security section of the main blade for your Azure SQL Database and click on Data Discovery and Classification.

AdventureWorks (dp300-lab-adw/AdventureWorks) | Advanced data security

Power BI (preview)

Power Apps (preview)

Power Automate (preview)

Settings

Configure

Geo-Replication

Connection strings

Sync to other databases

Add Azure Search

Properties

Locks

Export template

Integrations

Stream analytics (preview)

Security

Advanced data security

Complete Advanced Data Security setup by selecting a storage account for Vulnerability Assessment →

**Data Discovery & Classification**

0 TOTAL

Recommended columns to classify

Column	Sensitivity label
PasswordHash	Confidential
PasswordSalt	Confidential
FirstName	Confidential - GDPR
...	

**Vulnerability Assessment**

0 TOTAL

HIGH RISK FAILURES  
MEDIUM RISK FAILURES  
LOW RISK FAILURES

Failed Checks  
Click to configure a storage account for storing scan SECURITY CHECK results. RISK

There are no failing security checks.

**Advanced Threat Protection**

0 TOTAL

HIGH SEVERITY ALERTS  
MEDIUM SEVERITY ALERTS

Security Alerts

Description	Date
There are no alerts or recommendations to display.	

Turn on auditing for full investigation experience

On the next screen you will see an informational message that says “We have found 32 columns with classification recommendations”. Click on that link.

## Data Discovery & Classification

[Export](#) [Configure](#) [Feedback](#)

We have found 32 columns with classification recommendations →

[Overview](#) [Classification](#)

Classified columns **0 / 513** Tables containing sensitive data **0 / 76** Unique information types **0**

Label distribution

Information type distribution

On the next screen. Click the check box next to Select all, and then click “Accepted selected recommendations”.

[Data Discovery & Classification](#)

[Save](#) [Discard](#) [+ Add classification](#) [Feedback](#)

[Learn more - Getting Started](#)

[Overview](#) [Classification](#)

**32 columns with classification recommendations** (Click to minimize)

[Accept selected recommendations](#) [Dismiss selected recommendations](#)  Show dismissed recommendations

<input checked="" type="checkbox"/> Select all	Schema: 6 selected	Table: 16 selected	Filter by column	Information type: 7 selected	Sensitivity label: 2 selected
<input checked="" type="checkbox"/>	Person	Password	PasswordHash	Credentials	Confidential
<input checked="" type="checkbox"/>	Person	Password	PasswordSalt	Credentials	Confidential
<input checked="" type="checkbox"/>	Person	Person	FirstName	Name	Confidential - GDPR
<input checked="" type="checkbox"/>	Person	Person	LastName	Name	Confidential - GDPR
<input checked="" type="checkbox"/>	Person	PersonPhone	PhoneNumber	Contact Info	Confidential
<input checked="" type="checkbox"/>	Person	PersonPhone	PhoneNumberTypeID	Contact Info	Confidential
<input checked="" type="checkbox"/>	Person	PhoneNumberType	PhoneNumberTypeID	Contact Info	Confidential
<input checked="" type="checkbox"/>	Production	ProductReview	EmailAddress	Contact Info	Confidential
<input checked="" type="checkbox"/>	Purchasing	PurchaseOrderHeader	TaxAmt	Financial	Confidential
<input checked="" type="checkbox"/>	Purchasing	Vendor	AccountNumber	Financial	Confidential
<input checked="" type="checkbox"/>	Sales	CreditCard	CreditCardID	Credit Card	Confidential
<input checked="" type="checkbox"/>	Sales	CreditCard	CardType	Credit Card	Confidential
<input checked="" type="checkbox"/>	Sales	CreditCard	CardNumber	Credit Card	Confidential
<input checked="" type="checkbox"/>	Sales	CreditCard	ExpYear	Credit Card	Confidential
<input checked="" type="checkbox"/>	Sales	CurrencyRate	CurrencyRateID	Financial	Confidential
<input checked="" type="checkbox"/>	Sales	CurrencyRate	CurrencyRateDate	Financial	Confidential
<input checked="" type="checkbox"/>	Sales	PersonCreditCard	CreditCardID	Credit Card	Confidential
<input checked="" type="checkbox"/>	Sales	SalesOrderHeader	AccountNumber	Financial	Confidential

Click **Save** to save the classifications into the database.

4. Go back to SQL Server Management Studio on your VM. Launch a new query window from the AdventureWorks database and execute the following query.

```
SELECT o.name AS [Table Name]
```

```
,ac.name AS [Column Name]
```

```
,sc.label
```

```
,sc.information_type
```

```

FROM sys.sensitivity_classifications sc
INNER JOIN sys.objects o ON o.object_id = sc.major_id
INNER JOIN sys.all_columns ac ON ac.column_id = sc.minor_id
WHERE ac.object_id = o.object_id;

```

This query will return the results of your classified columns as shown below.

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top window is titled "SQLQuery4.sql - dp...(dp300admin (94))". It contains a T-SQL query that selects columns from sys.sensitivity\_classifications, sys.objects, and sys.all\_columns. The results grid below shows 32 rows of data, each mapping a table name, column name, sensitivity label, and information type. The columns are labeled "Table Name", "Column Name", "label", and "information\_type".

	Table Name	Column Name	label	information_type
1	Password	PasswordHash	Confidential	Credentials
2	Password	PasswordSalt	Confidential	Credentials
3	Person	FirstName	Confidential - GDPR	Name
4	Person	LastName	Confidential - GDPR	Name
5	PersonPhone	PhoneNumber	Confidential	Contact Info
6	PersonPhone	PhoneNumberTypeID	Confidential	Contact Info
7	PhoneNumberType	PhoneNumberTypeID	Confidential	Contact Info
8	ProductReview	EmailAddress	Confidential	Contact Info
9	PurchaseOrderHeader	TaxAmt	Confidential	Financial
10	Vendor	AccountNumber	Confidential	Financial
11	CreditCard	CardNumber	Confidential	Credit Card
12	CreditCard	CardType	Confidential	Credit Card
13	CreditCard	CreditCardID	Confidential	Credit Card
14	CreditCard	ExpYear	Confidential	Credit Card
15	CurrencyRate	CurrencyRateDate	Confidential	Financial
16	CurrencyRate	CurrencyRateID	Confidential	Financial
17	PersonCreditCard	CreditCardID	Confidential	Credit Card
18	SalesOrderHeader	AccountNumber	Confidential	Financial
19	SalesOrderHeader	CreditCardApprovalCode	Confidential	Credit Card
20	SalesOrderHeader	CreditCardID	Confidential	Credit Card
21	SalesOrderHeader	CurrencyRateID	Confidential	Financial
22	SalesOrderHeader	TaxAmt	Confidential	Financial
23	ErrorLog	UserName	Confidential	Credentials
24	SalesTaxRate	SalesTaxRateID	Confidential - GDPR	National ID
25	SalesTaxRate	TaxRate	Confidential	Financial
26	Employee	BirthDate	Confidential - GDPR	Date Of Birth
27	Employee	NationalIDNumber	Confidential - GDPR	National ID
28	Address	AddressLine1	Confidential	Contact Info
29	Address	AddressLine2	Confidential	Contact Info
30	Address	City	Confidential	Contact Info
31	Address	PostalCode	Confidential	Contact Info
32	EmailAddress	EmailAddress	Confidential	Contact Info

## 21.4 Exercise 4: Manage access to database objects

- In this exercise, you will manage access to the database and its objects. The first thing you will do is create two users in the AdventureWorks database. Open a new query window and copy and paste the following T-SQL into it. Verify that the code has been copied correctly. Execute the query to create the two users.

```
CREATE USER [DP300User1] WITH PASSWORD = 'Azur3Pa$$'
```

```
GO
```

```
CREATE USER [DP300User2] WITH PASSWORD = 'Azur3Pa$$'
```

```
GO
```

You will note these users are created in the scope of the database. So if you were to try to login with one of these users, you would need to specify the AdventureWorks database in your connection string.

2. Next you will create a custom role and add the users to it. Execute the following T-SQL in the same query window as in step 1.

```
CREATE ROLE [SalesReader]
```

```
GO
```

```
ALTER ROLE [SalesReader] ADD MEMBER [DP300User1]
```

```
GO
```

```
ALTER ROLE [SalesReader] ADD MEMBER [DP300User2]
```

```
GO
```

3. Next you will grant permissions to the role. In this case you are assigning SELECT and EXECUTE on the Sales schema. Execute the below T-SQL to grant the permissions to the role.

```
GRANT SELECT, EXECUTE ON SCHEMA::Sales TO [SalesReader]
```

```
GO
```

4. Next you will create a new stored procedure in the Sales schema. You will note this procedure access a table in the Product schema. Execute the below T-SQL in your query window.

```
CREATE OR ALTER PROCEDURE Sales.DemoProc
```

```
AS
```

```
SELECT P.Name, Sum(SOD.LineTotal) as TotalSales ,SOH.OrderDate  
FROM Production.Product P  
INNER JOIN Sales.SalesOrderDetail SOD on SOD.ProductID = P.ProductID  
INNER JOIN Sales.SalesOrderHeader SOH on SOH.SalesOrderID = SOD.SalesOrderID  
GROUP BY P.Name, SOH.OrderDate  
ORDER BY TotalSales DESC
```

```
GO
```

5. Next you will use the EXECUTE AS USER command to test out the security you just created. This allows the database engine to execute a query in the context of your user. Execute the below query in your query window.

```
EXECUTE AS USER = 'DP300User1'
```

```
SELECT P.Name, Sum(SOD.LineTotal) as TotalSales ,SOH.OrderDate  
FROM Production.Product P
```

```

INNER JOIN Sales.SalesOrderDetail SOD on SOD.ProductID = P.ProductID
INNER JOIN Sales.SalesOrderHeader SOH on SOH.SalesOrderID = SOD.SalesOrderID
GROUP BY P.Name, SOH.OrderDate
ORDER BY TotalSales DESC

```

This query will fail, with an error message saying the SELECT permission was denied on the Production.Product table. The role that user DP300User1 is a member of has SELECT permission in the Sales schema, but not in the Production schema.

However, if you execute the stored procedure in that same context, the query will complete. Execute the following T-SQL.

```
EXECUTE AS USER = 'DP300User1'
```

```
EXECUTE Sales.DemoProc
```

This happens because stored procedures take advantage a feature called ownership chaining to provide data access to users who do not have direct permissions to access database objects. For all objects that have the same owner, the database engine only checks the EXECUTE permission on the procedure and not the underlying objects.

**21.5 Do not remove any of the resources created in this lab as they will be used in subsequent lab exercises.**

**21.6 lab: title: 'Lab 4 – Monitor and Optimize Resources' module: 'Monitor and Optimize Operational Resources'**

## 22 Lab 4 – Monitor and Optimize Resources

**Estimated Time:** 60 minutes

**Pre-requisites:** Lab VM, and Azure SQL Database created in Labs 2 and 3

## 23 Lab overview

The students will take the information gained in the lessons to scope out the deliverables for a digital transformation project within AdventureWorks. Examining the Azure portal as well as other tools, students will determine how to utilize native tools to identify and resolve performance related issues. Finally, students will be able to identify fragmentation within the database as well as learn steps to resolve the issue appropriately.

## 24 Lab objectives

After completing this lab, you will be able to:

1. Isolate performance problems through monitoring.
2. Utilize query store to locate & resolve blocking issues by creating a new index
3. Detect and correct fragmentation issues.

## 25 Scenario

You have been hired as a database administrator to identify performance related issues and provide viable solutions to resolve any issues found. AdventureWorks has been selling bicycles and bicycle parts directly to consumers and distributors for over a decade. The company utilizes a mixture of both Azure SQL Databases as well as a SQL Server in an Azure Virtual Machine. Recently the company has noticed performance degradation within both products that are used to service customer requests. You need to use the Azure portal as well as on-premises tools to identify the performance issues and suggest methods to resolve them.

**Note:** The exercises ask you to copy and paste T-SQL code. Please verify that the code has been copied correctly, with the proper line breaks, before executing the code.

## 26 Exercise 1: Isolate problems by monitoring

Estimated Time: 30 minutes

The main tasks for this exercise are as follows:

1. Review monitoring solutions to identify performance related issues for Azure SQL Database
2. Use Query Store to identify any ongoing blocking issues as well identify appropriate indexes to implement to resolve the blocking.
3. Use querying tools to identify index fragmentation as well as provide a method to resolve the fragmentation.

### 26.1 Task 1: Review CPU Utilization in Azure Portal

1. If you're not already in the Azure portal, navigate back to portal.azure.com. You may need to login again. In the search bar at the top of the Azure Portal, type SQL, and click on SQL databases. This task will use the AdventureWorks database you created in Lab 3. Click on the database name.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, the 'SQL databases' blade is open. The URL in the address bar is 'https://portal.azure.com/#blade/HubsBlade/resourceType/sqlDatabases'. The main content area shows a table with one item: 'AdventureWorks (dp300-lab-adw/AdventureWorks)'. The table columns include Name, Status, Replication role, Server, Pricing tier, and Location. The 'AdventureWorks' row is highlighted with a blue background.

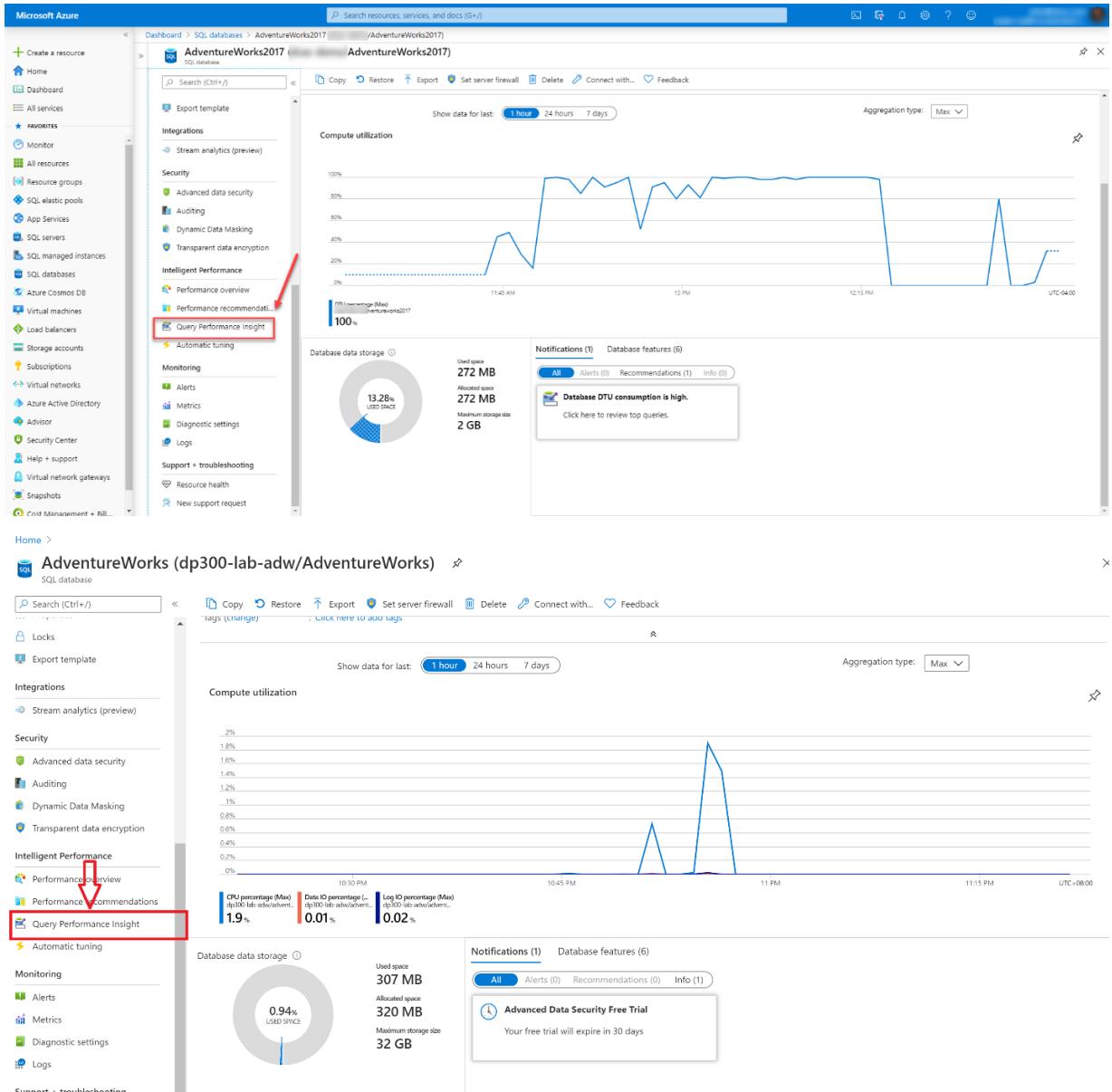
2. Locate and select Metrics on the Monitoring section of the blade for the AdventureWorks database.
3. Change the Metric menu option to reflect CPU percentage

The screenshot shows the 'AdventureWorks (dp300-lab-kmd/AdventureWorks) | Metrics' blade. The left sidebar has sections for Alerts, Metrics (which is selected and highlighted with a red box), Diagnostic settings, Logs, Tasks, Export template, Support + troubleshooting, Resource health, and New support request. The main area shows a chart title 'Chart Title' and a metric selection interface. The 'Scope' is set to 'AdventureWorks'. The 'Metric Namespace' dropdown is set to 'Sql database standard ...'. The 'Metric' dropdown is open, showing options like 'Select metric', 'BASIC', 'Blocked by Firewall', 'CPU limit', and 'CPU percentage' (which is highlighted with a red box). The 'Aggregation' dropdown is set to 'Select aggregation'. The chart area shows a line graph for CPU percentage over time, with the Y-axis ranging from 0 to 100 and the X-axis showing dates from Fri 25 to Sun 27.

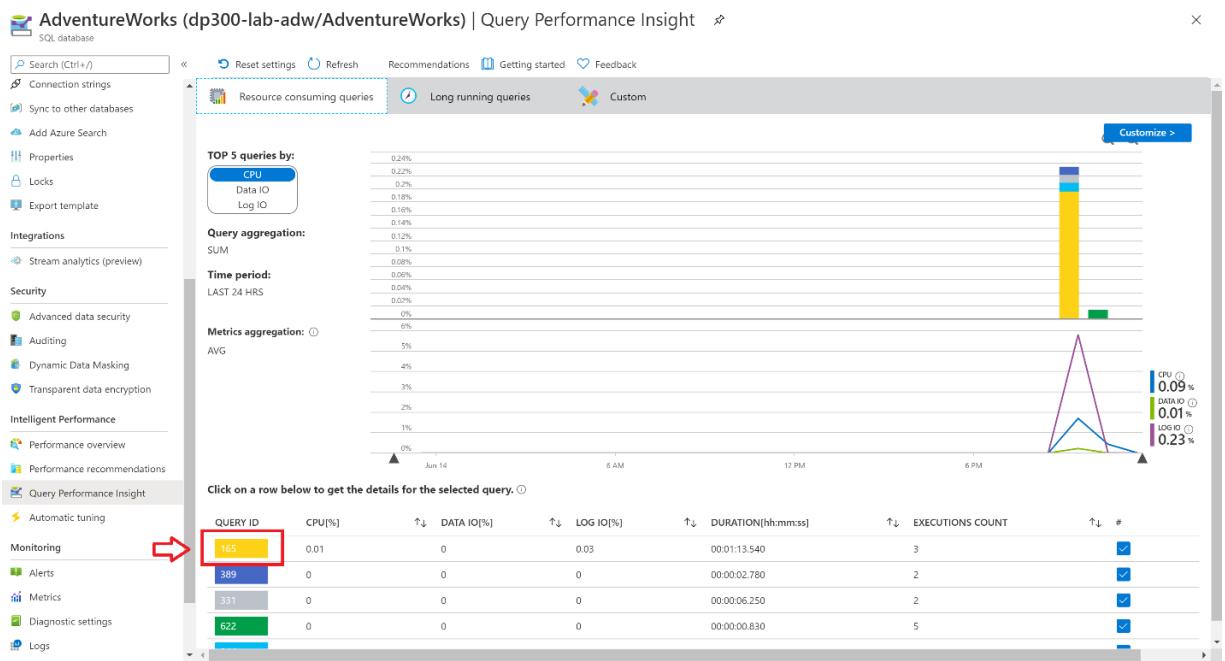
4. Select an Aggregation of "Avg". This will display the average CPU Percentage for the given time frame.
5. Now adjust the aggregation to Max to identify the maximum CPU Percentage used for the give time frame. How is it different from the average?

## 26.2 Task 2: Identify High CPU Consuming Queries

1. In the same view as shown in first task, select Query Performance Insight in the same menu under Intelligent Performance.



2. Click on the first query in the grid below the graph.



3. Make note of the query, and its resource usage as shown in the following image. Azure tracks CPU, Data IO, and Log IO for each query.

## Query details

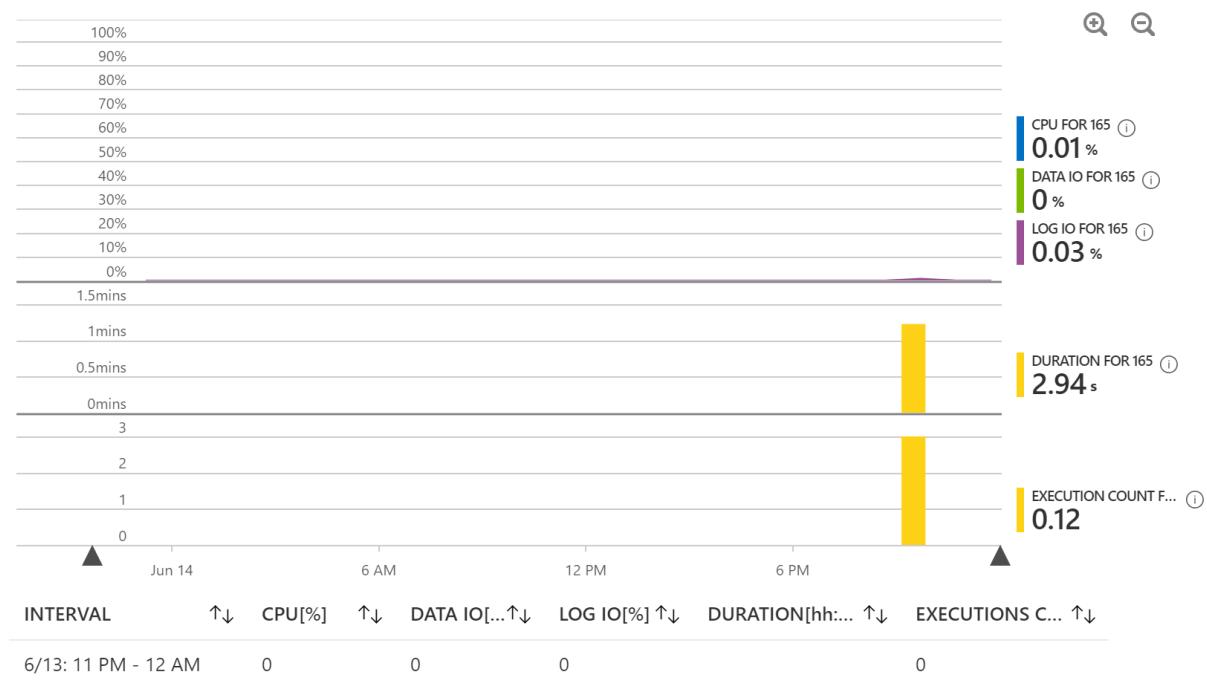
AdventureWorks - Query ID 165

Settings Refresh Recommendations Query Text

**Query ID 165:**

```
1  insert bulk [Person].[Person] ([BusinessEntityID] Int, [PersonType] NChar(2) COLLATE
SQL_Latin1_General_CI_AS, [NameStyle] Bit, [Title] NVarChar(8) COLLATE
SQL_Latin1_General_CI_AS, [FirstName] NVarChar(50) COLLATE
SQL_Latin1_General_CI_AS, [MiddleName] NVarChar(50) COLLATE
SQL_Latin1_General_CI_AS, [LastName] NVarChar(50) COLLATE
SQL_Latin1_General_CI_AS, [Suffix] NVarChar(10) COLLATE SQL_Latin1_General_CI_AS,
[EmailPromotion] Int, [AdditionalContactInfo] Xml, [Demographics] Xml, [rowguid]
UniqueIdentifier, [ModifiedDate] DateTime) with (KEEP_NULLS)
```

**Details of Query ID 165 (Query aggregation: sum) Last 24 hrs**



## 27 Exercise 2: Identify and Resolve Blocking Issues

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Determine which query is causing blocking by utilizing the blocking report with extended events
2. Identify indexes that could be implemented to resolve the blocking

### 27.1 Task 1: Run blocked queries report

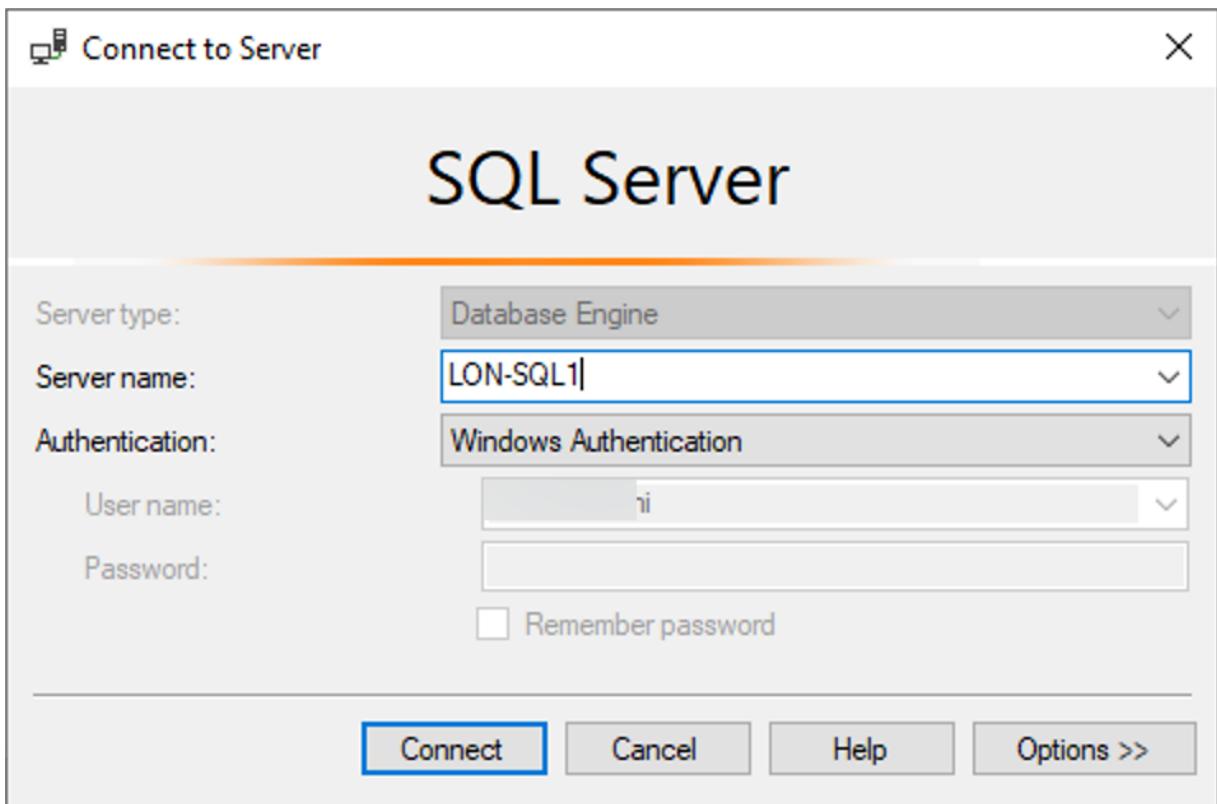
For this exercise, you will be using SQL Server Management Studio from the lab VM and connecting to an on-premises SQL Server.

1. From the lab VM, open SQL Server Management Studio, by opening the start menu and navigating to Microsoft SQL Server Tools 18> Microsoft SQL Server Management Studio. Alternatively, you can select the icon for the Management Studio in your task bar.

2. Start a new query by clicking the New Query button in Management Studio.



You will be prompted to connect to your SQL Server.



Enter LON-SQL1 for the local server name, and ensure that Windows Authentication is selected, and click connect.

3. Copy and paste the code below into your query window.

```
USE MASTER  
GO  
  
CREATE EVENT SESSION [Blocking] ON SERVER  
  
ADD EVENT sqlserver.blocked_process_report(  
ACTION(sqlserver.client_app_name,sqlserver.client_hostname,sqlserver.database_id,sqlserver.databas  
  
ADD TARGET package0.ring_buffer  
  
WITH (MAX_MEMORY=4096 KB, EVENT_RETENTION_MODE=ALLOW_SINGLE_EVENT_LOSS, MAX_DISPATCH_LATENCY=30 SE  
  
GO
```

```
-- Start the event session  
  
ALTER EVENT SESSION [Blocking] ON SERVER  
  
STATE = start;  
  
GO
```

Click the execute button to execute this query.

The above T-SQL code will create an Extended Event session that will capture blocking events. The data will contain the following elements:

- Client application name
- Client host name
- Database ID
- Database name
- NT Username
- Session ID
- T-SQL Text
- Username

4. Click New Query from SQL Server Management Studio. Copy and paste the following T-SQL code into the query window. Click the execute button to execute this query.

```
USE AdventureWorks2017
```

```
GO
```

```
BEGIN TRANSACTION
```

```
UPDATE Person.Person SET LastName = LastName;
```

```
GO
```

5. Open another query window by clicking the New Query button. Copy and paste the following T-SQL code into the query window. Click the execute button to execute this query.

```
USE AdventureWorks2017
```

```
GO
```

```
SELECT TOP (1000) [LastName]
```

```
, [FirstName]
```

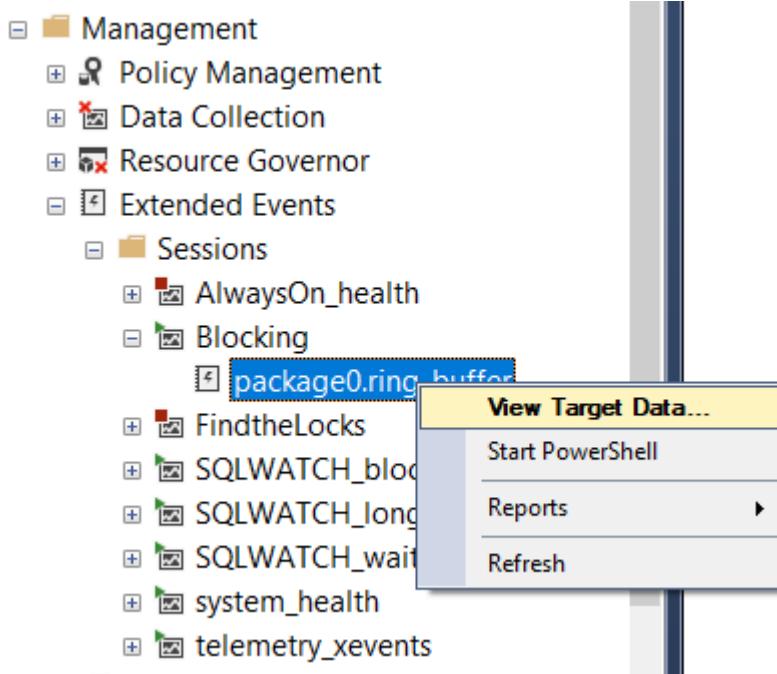
```
, [Title]
```

```
FROM Person.Person
```

```
WHERE FirstName = 'David'
```

You should notice that this query does not return results immediately and appears to be still running.

6. In Object Explorer, navigate to Management, and expand the hive by clicking the plus sign. Expand the Extended Events hive and then expand the Sessions Hive. Expand the Blocking session that was created in step 3. Right click on package0.ring\_buffer and select View target data.



- Click on the hyperlink.

The target data will be refreshed every 5 seconds. Right click the table to manually refresh or set the automatic refresh interval.

Data
<pre>&lt;RingBufferTarget truncated="0" processingTime="0" totalEventsProcessed="17" eventCount="17" droppedCount="0" memoryUsed="68614"&gt;&lt;event name="blocked"&gt; &lt;process id="process2a58407d848" taskpriority="0" logused="0" waitresource="&lt;resource type="key" database="master" object="1" index="1" id="1" handle="0x02000000ab2c1b2044addc4a871e41"/&gt; &lt;executionStack&gt; &lt;frame line="1" stmtend="228" sqlhandle="0x02000000ab2c1b2044addc4a871e41"/&gt; &lt;/executionStack&gt; &lt;inputbuf&gt; SELECT TOP (1000) [LastName] ,[FirstName] ,[Title] FROM Person.Person where firstname = 'David'    &lt;/inputbuf&gt; &lt;/process&gt; &lt;/blocked-process&gt; &lt;blocking-process&gt; &lt;process status="sleeping" spid="76" sbid="0" ecid="0" priority="0" trancount="0" transactionname=""&gt;&lt;executionStack /&gt; &lt;inputbuf&gt; BEGIN TRANSACTION UPDATE Person.Person SET LastName = LastName; &lt;/inputbuf&gt; &lt;/process&gt; &lt;/blocking-process&gt; &lt;/blocked-process-report&gt; &lt;/value&gt; &lt;/data&gt;</pre>

- The XML will show you which processes are being blocked and which process is causing the blocking.

```

<blocked-process>
<process id="process2a58407d848" taskpriority="0" logused="0" waitresource="<resource type="key" database="master" object="1" index="1" id="1" handle="0x02000000ab2c1b2044addc4a871e41"/>
<executionStack>
<frame line="1" stmtend="228" sqlhandle="0x02000000ab2c1b2044addc4a871e41"/>
</executionStack>
<inputbuf>
SELECT TOP (1000) [LastName]
,[FirstName]
,[Title]
FROM Person.Person
where firstname = 'David'    </inputbuf>
</process>
</blocked-process>
<blocking-process>
<process status="sleeping" spid="76" sbid="0" ecid="0" priority="0" trancount="0" transactionname=""><executionStack />
<inputbuf>
BEGIN TRANSACTION
UPDATE Person.Person SET LastName = LastName;
</inputbuf>
</process>
</blocking-process>
</blocked-process-report>
</value>
</data>
```

- Right click the Extended Events session (the one called Blocking) and select Stop Session.

10. Navigate back to the query tab you opened in step 4, and type ROLLBACK TRANSACTION on the line below the query. Highlight ROLLBACK TRANSACTION and execute the command.

```
SQLQuery3.sql - LO...SQL1\Student (64)*      SQLQuery2.sql - LO...SQL1\Student (54)*  ✎ X
USE AdventureWorks2017
GO
BEGIN TRANSACTION
UPDATE Person.Person SET LastName = LastName;
GO
ROLLBACK TRANSACTION
```

11. Navigate back to the query tab you opened in Step 5. You will notice that the query has now completed.
12. Close all your query windows.

## 27.2 Task 2: Enabling Read Committed Snapshot Isolation

1. Open a new query window. Copy and paste the following T-SQL code into the query window. Click the execute button to execute this query.

```
USE master
GO
ALTER DATABASE AdventureWorks2017 SET READ_COMMITTED_SNAPSHOT ON WITH ROLLBACK IMMEDIATE;
```

2. Copy and paste the following T-SQL code into a new query window. Click the execute button to execute this query.

```
USE AdventureWorks2017
GO
BEGIN TRANSACTION
UPDATE Person.Person SET LastName = LastName;
```

3. Copy and paste the following T-SQL code into a new query window. Click the execute button to execute this query.

```
USE AdventureWorks2017
GO
SELECT TOP (1000) [LastName]
,[FirstName]
,[Title]
FROM Person.Person
WHERE firstname = 'David';
```

4. Discuss with the instructor why the query in step 3 now completes whereas in the previous exercise it was blocked by the UPDATE. Keep SQL Server Management Studio open as you will need it in the next exercise, but close all your query windows.

## 28 Exercise 3: Detect/Correct Fragmentation Issues

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Identify fragmentation
2. Remove fragmentation
3. The instructor will discuss the findings with the group.

### 28.1 Task 1: Identify fragmentation

1. In SQL Server Management Studio, click the New Query button. Copy and paste the following T-SQL code into the query window. Click the execute button to execute this query.

```
USE AdventureWorks2017  
GO
```

```
INSERT INTO [Person].[Address]
```

```
([AddressLine1]  
, [AddressLine2]  
, [City]  
, [StateProvinceID]  
, [PostalCode]  
, [SpatialLocation]  
, [rowguid]  
, [ModifiedDate])
```

```
SELECT AddressLine1,  
       AddressLine2,  
       'Amsterdam',  
       StateProvinceID,  
       PostalCode,  
       SpatialLocation,  
       newid(),  
       getdate()  
FROM Person.Address;
```

```
GO
```

This query will increase the fragmentation level of the Person.Address table and its indexes.

2. Copy and paste the following T-SQL code into a new query window. Click the execute button to execute this query.

```

USE AdventureWorks2017
GO
SELECT i.name Index_Name
    , avg_fragmentation_in_percent
    , db_name(database_id)
    , i.object_id
    , i.index_id
    , index_type_desc
FROM sys.dm_db_index_physical_stats(db_id('AdventureWorks2017'),object_id('person.address'),NULL,NULL,'')
INNER JOIN sys.indexes i ON ps.object_id = i.object_id
AND ps.index_id = i.index_id
WHERE avg_fragmentation_in_percent > 50 -- find indexes where fragmentation is greater than 50%

```

This query will report any indexes that have a fragmentation over 50%. You should see four indexes with fragmentation.

3. Copy and paste the following T-SQL code into a new query window. Click the execute button to execute this query.

```

SET STATISTICS IO,TIME ON
GO
USE AdventureWorks2017
GO
SELECT DISTINCT (StateProvinceID)
    ,count(StateProvinceID) AS CustomerCount
FROM person.Address
GROUP BY StateProvinceID
ORDER BY count(StateProvinceID) DESC;
GO

```

Click on the Messages tab in the result pane of SQL Server Management Studio. Make note of the count of logical reads performed by the query.

Results Messages

```

SQL Server parse and compile time:
    CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
    CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time:
    CPU time = 93 ms, elapsed time = 93 ms.

(74 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0,
Table 'Address'. Scan count 1, logical reads 94, physical reads 0, I
SQL Server Execution Times:
    CPU time = 15 ms, elapsed time = 6 ms.

```

## 28.2 Task 2: Rebuild indexes

1. Click the new query button.
2. Copy and paste the following T-SQL code into the query window. Click the execute button to execute this query.

```
USE AdventureWorks2017
```

```
GO
```

```
ALTER INDEX [IX_Address_StateProvinceID] ON [Person].[Address] REBUILD PARTITION = ALL WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF)
```

3. Re-execute the query from step 2 in Task 1. Confirm that the AK\_Address\_StateProvinceID index no longer has fragmentation greater than 50%.
  4. Re-execute the query from step 3 in Task 1. Make note of the logical reads in the Messages tab of the Results pane in Management Studio. Was there a change from the number of logical reads encountered before you rebuilt the index?
- 

## 28.3 lab: title: 'Lab 5 -Query Performance Troubleshooting' module: 'Optimize Query Performance'

# 29 Lab 5 -Query Performance Troubleshooting

**Estimated Time:** 75 minutes

**Lab files:** The files for this lab are located in the D:\Labfiles\Query Performance folder.

# 30 Lab overview

The students will evaluate a database design for problems with normalization, data type selection and index design. They will run queries with suboptimal performance, examine the query plans, and attempt to make improvements within the AdventureWorks2017 database.

# 31 Lab objectives

After completing this lab, you will be able to:

1. Identify issues with database design
  - Evaluate queries against database design
  - Examine existing design for potential bad patterns such as over/under normalization or incorrect data types

2. Isolate problem areas in poorly performing queries
  - Run query to generate actual execution plan not using the GUI
  - Evaluate given execution plans (such as key lookup)
3. Use Query Store to detect and handle regressions
  - Run a workload to generate query statistics for Query Store
  - Examine Top Resource Consuming Queries to identify poor performance
  - Force a better execution plan
4. Use query hints to impact performance
  - Run workload
  - Change the query to use a Parameter value
  - Apply query hint to query to optimize for a value

## 32 Scenario

You have been hired as a Senior Database Administrator to help with performance issues currently happening when users query the AdventureWorks2017 database. Your job is to identify issues in query performance and remedy them using techniques learned in this module.

The first step is to review the queries the users are having issues with and make recommendations:

1. Identify issues with database design within AdventureWorks2017
2. Isolate problem areas in poorly performing queries in AdventureWorks2017
3. Use Query Store to detect and handle regressions in AdventureWorks2017
4. Use Query Hints to impact performance in AdventureWorks2017

**Note:** The exercises ask you to copy and paste T-SQL code. Please verify that the code has been copied correctly, with the proper line breaks, before executing the code.

## 33 Exercise 1: Identify issues with database design in Adventure-Works2017

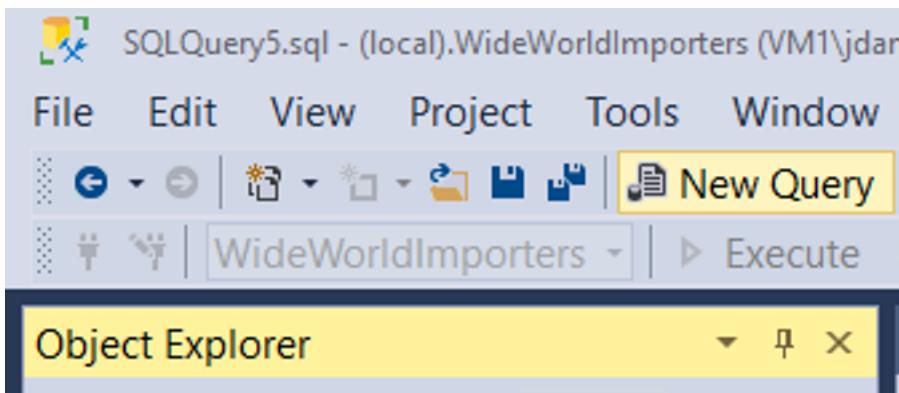
Estimated Time: 15 minutes

The main task for this exercise is as follows:

1. Examine the query and identify why you are seeing a warning and what that warning is.
2. Come up with two ways to fix the issue.
  - Change the query to resolve the issue.
  - Suggest a database design change to fix the issue.

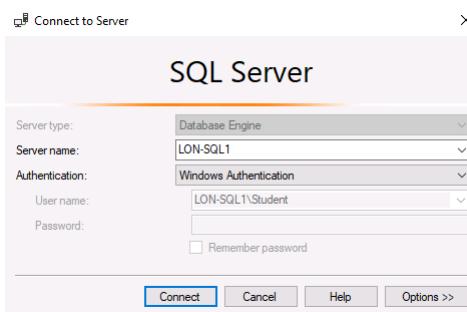
### 33.1 Task 1: Examine the query and identify the problem.

1. From the lab virtual machine, start **SQL Server Management Studio (SSMS)**.



2. You will be prompted to connect to your SQL Server.

Enter the server name LON-SQL1, and ensure that Windows Authentication is selected, and click connect.



3. Open a new query window. Copy and paste the code below into your query window.

```
USE AdventureWorks2017;

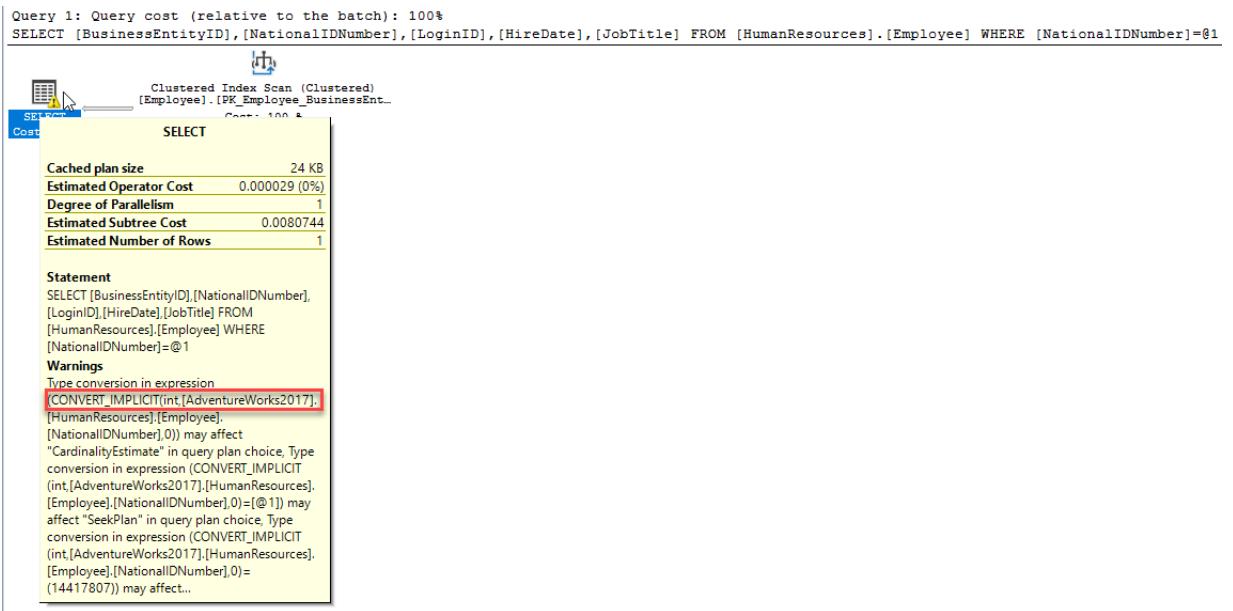
SELECT BusinessEntityID, NationalIDNumber, LoginID, HireDate, JobTitle
FROM HumanResources.Employee
WHERE NationalIDNumber = 14417807;
```

4. Click on Include Actual Execution Plan icon as shown below before running the query or type CTRL+M. This will cause the execution plan to be displayed when you execute the query.



5. Click the execute button to execute this query.

6. Navigate to the execution plan, by clicking on Execution plan tab in the results panel in SSMS. In the execution plan, move your mouse over the SELECT operator. You will note a warning message identified by an exclamation point in a yellow triangle as shown below. Identify what the Warning Message tells you.



An Implicit Conversion is causing a performance issue.

### 33.2 Task 2: Identify two ways to fix the warning issue

The structure for the table is shown in the follow data definition language (DDL) statement.

```
CREATE TABLE [HumanResources].[Employee] (
    [BusinessEntityID] [int] NOT NULL,
    [NationalIDNumber] [nvarchar](15) NOT NULL,
    [LoginID] [nvarchar](256) NOT NULL,
    [OrganizationNode] [hierarchyid] NULL,
    [OrganizationLevel] AS ([OrganizationNode].[GetLevel]()),
    [JobTitle] [nvarchar](50) NOT NULL,
    [BirthDate] [date] NOT NULL,
    [MaritalStatus] [nchar](1) NOT NULL,
    [Gender] [nchar](1) NOT NULL,
    [HireDate] [date] NOT NULL,
    [SalariedFlag] [dbo].[Flag] NOT NULL,
    [VacationHours] [smallint] NOT NULL,
    [SickLeaveHours] [smallint] NOT NULL,
    [CurrentFlag] [dbo].[Flag] NOT NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

- Fix the query using code as a solution.

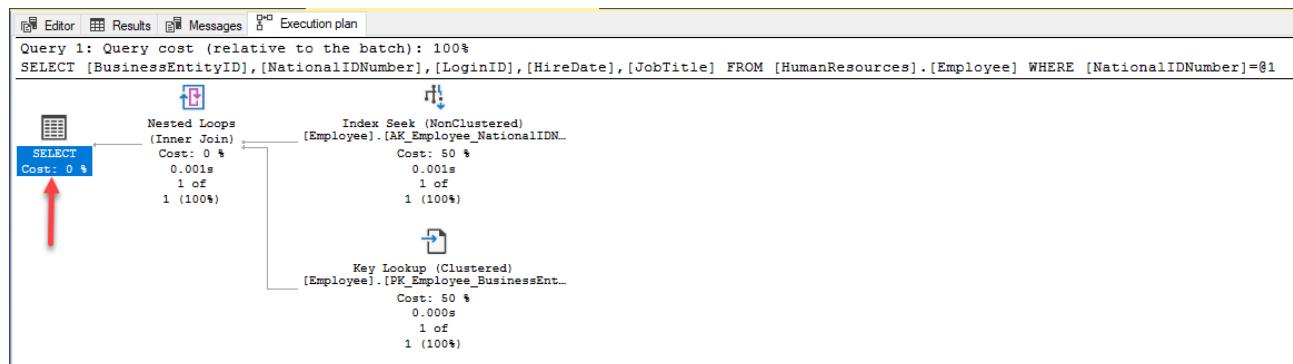
Identify what column is causing the implicit conversion and why.

If you review the query from Task 1, you will note the value compared to the NationalIDNumber column in the WHERE clause is entered as a number, since it is not a quoted string. After examining the table structure you will find this column in the table is using the nvarchar(15) datatype and not the int or integer data type. This data type inconsistency causes the optimizer to implicitly convert the constant to a nvarchar upon execution causing additional overhead to the query performance with a suboptimal plan.

- Change the code to resolve the implicit conversion and rerun the query. Remember to turn on the Include Actual Execution Plan (Cntl+M) if it is not already on from the exercise above. Note the warning is now gone.

By changing the WHERE clause so that the value compared to the NationalIDNumber column matches the column's data type in the table, you can get rid of the implicit conversion. In this scenario just adding a single quote on each side of the value changes it from a number to a character string. Keep the query window open for this query.

```
SELECT BusinessEntityID, NationalIDNumber, LoginID, HireDate, JobTitle
FROM HumanResources.Employee
WHERE NationalIDNumber = '14417807'
```



- Fix the query using database design changes.

To attempt to fix the index, open a new query window, and copy the query below to change the column's data type. Attempt to execute the query by clicking Execute.

```
ALTER TABLE [HumanResources].[Employee] ALTER COLUMN [NationalIDNumber] INT NOT NULL;
```

The changes to the table would solve the conversion issue. However this change introduces another issue that as a database administrator you need to resolve. Since this column is part of an already existing nonclustered index, the index has to be rebuilt/recreated in order to execute the data type change. This could lead to extended downtime in production, which highlights the importance of choosing the right data types in your design.

Msg 5074, Level 16, State 1, Line 1The index 'AK\_Employee\_NationalIDNumber' is dependent on column 'NationalIDNumber'.

Msg 4922, Level 16, State 9, Line 1

ALTER TABLE ALTER COLUMN NationalIDNumber failed because one or more objects access this column.

- In order to resolve this issue, copy and paste the code below into your query window and execute it by clicking Execute.

```
USE AdventureWorks2017
GO
```

```
DROP INDEX [AK_Employee_NationalIDNumber] ON [HumanResources].[Employee]
GO
```

```
ALTER TABLE [HumanResources].[Employee] ALTER COLUMN [NationalIDNumber] INT NOT NULL;
GO
```

```

CREATE UNIQUE NONCLUSTERED INDEX [AK_Employee_NationalIDNumber] ON [HumanResources].[Employee]
( [NationalIDNumber] ASC
);
GO

```

5. Rerun the original query without the quotes.

```

USE AdventureWorks2017;

SELECT BusinessEntityID, NationalIDNumber, LoginID, HireDate, JobTitle
FROM HumanResources.Employee
WHERE NationalIDNumber = 14417807;

```

Note that the execution plan no longer has a warning message.

## 34 Exercise 2: Isolate problem areas in poorly performing queries in AdventureWorks2017

Estimated Time: 30 minutes

The tasks for this exercise is as follows:

1. Run query to generate the actual execution plan.
2. Evaluate given execution plans (such as key lookup).

### 34.1 Task 1: Run a query to generate the actual execution plan

There are several ways to generate an execution plan in SQL Server Management Studio. You will use the same query from Exercise 1. Copy and paste the code below into a new query window and execute it by clicking Execute.

Using the SHOWPLAN\_ALL setting we can get the same information as we did in the last exercise but in the results pane instead of the graphical result.

```

USE AdventureWorks2017;

GO

SET SHOWPLAN_ALL ON;

SELECT BusinessEntityID
FROM HumanResources.Employee
WHERE NationalIDNumber = '14417807';

GO

SET SHOWPLAN_ALL OFF;

```

This shows you a text version of the execution plan.

StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument	DefinedValues	EstimateRows	EstimateIO	EstimateCPU	AvgRowSize	TotalSubtreeCost
1   SELECT BusinessEntityID FROM HumanResources.Emplo...	1	1	0	NULL	NULL	1	NULL	1	NULL	NULL	NULL	0.0032831
2    I-Index Seek([OBJECT @AdventureWorks2017].[HumanRes...	1	2	1	Index Seek	Index Seek	OBJECT:[AdventureWorks2017].[HumanResources].[E...	[AdventureWorks2017].[HumanResources].[Employee...	1	0.003125	0.0001581	11	0.0032831

## 34.2 Task 2: Resolve a Performance Problem from an Execution Plan

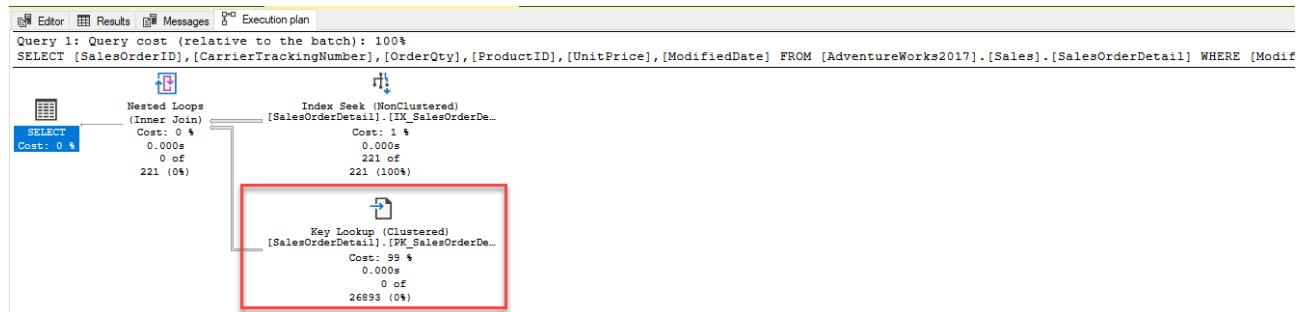
1. Copy and paste the code below into a new query window. Click on Include Actual Execution Plan icon before running the query, or type CTRL+M. Execute the query by clicking Execute. Make note of the execution plan and the logical reads in the messages tab.

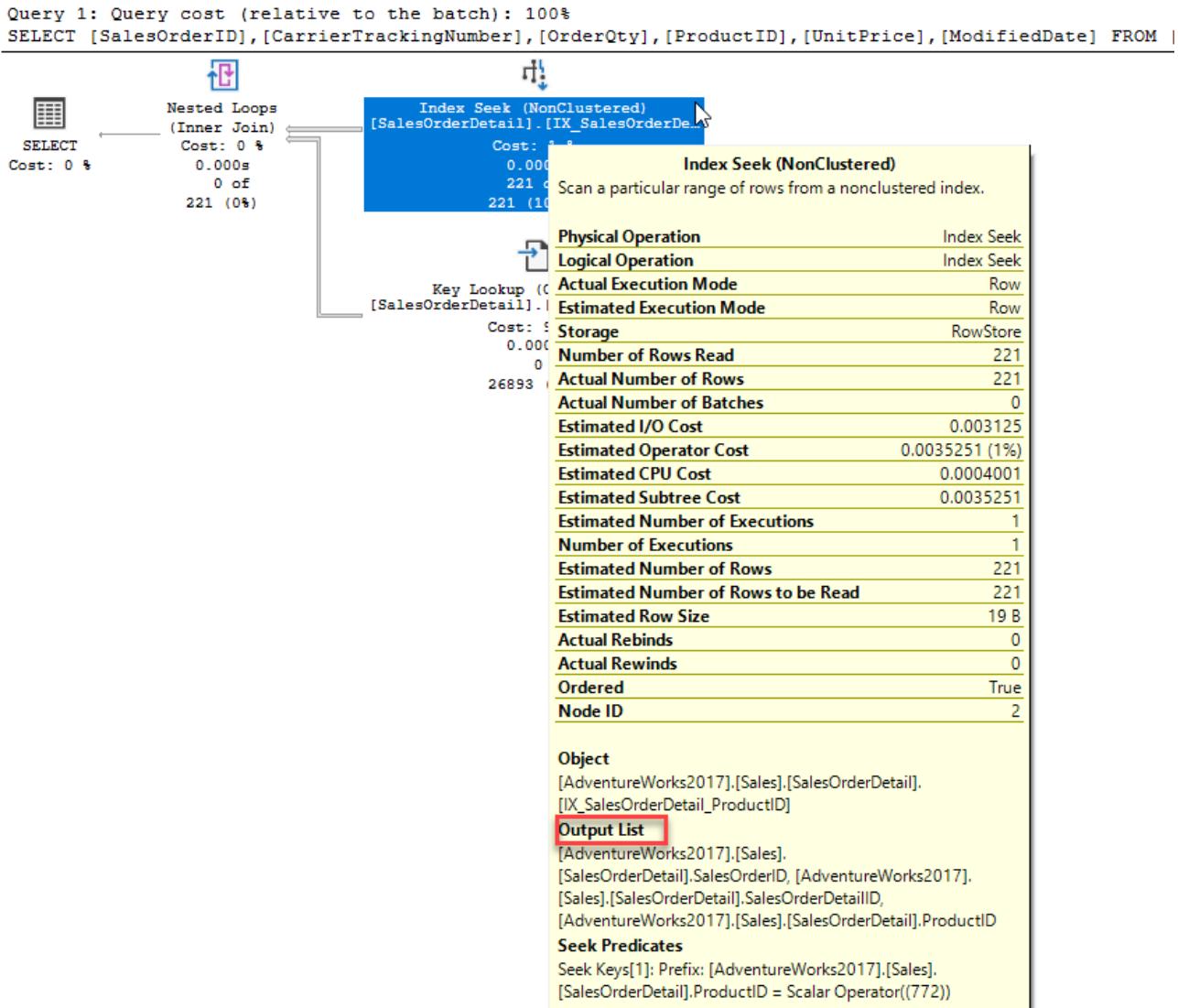
```
SET STATISTICS IO, TIME ON;
```

```
SELECT SalesOrderID, CarrierTrackingNumber, OrderQty, ProductID, UnitPrice, ModifiedDate  
FROM AdventureWorks2017.Sales.SalesOrderDetail WHERE ModifiedDate > '2012/05/01' AND ProductID = 772;
```

When reviewing the execution plan you will note there is a key lookup. If you move your mouse over the icon, you will see that the properties indicate it is performed for each row retrieved by the query. You can see the execution plan is performing a Key Lookup operation.

To identify what index needs to be altered in order to remove the key lookup, you need to examine the index seek above it. Hover over the index seek operator with your mouse and the properties of the operator will appear. Make note of the output list as shown below.





2. Fix the Key Lookup and rerun the query to see the new plan. Key Lookups are fixed by adding a COVERING index that INCLUDES all columns being returned or searched in the query. In this example the index only had **ProductID**. If we add the output list columns to the index as included columns, and add the other search column (**ModifiedDate**) as a key column, then the Key Lookup will be removed. Since the index already exists you either have to DROP the index and recreate it or set **DROP\_EXISTING=ON** in order to add the columns. Note **ProductID** is already part of the index and does not need to be added as an included column.

```
CREATE NONCLUSTERED INDEX IX_SalesOrderDetail_ProductID
ON [Sales].[SalesOrderDetail] (ProductID, ModifiedDate)
INCLUDE (CarrierTrackingNumber, OrderQty, UnitPrice)
WITH (DROP_EXISTING = ON);

GO
```

3. Rerun the query from Step 1. Make note of the changes to the logical reads and execution plan changes

## 35 Exercise 3: Use Query Store to detect and handle regression in AdventureWorks2017.

Estimated Time: 15 minutes

The tasks for this exercise are as follows:

1. Run a workload to generate query statistics for QS
2. Examine Top Resource Consuming Queries to identify poor performance
3. Force a better execution plan.

### 35.1 Task 1: Run a workload to generate query stats for Query Store

1. Copy and paste the code below into a new query window and execute it by clicking Execute. Make note of the execution plan and the logical reads in the messages tab. This script will enable the Query Store for AdventureWorks2017 and sets the database to Compatibility Level 100

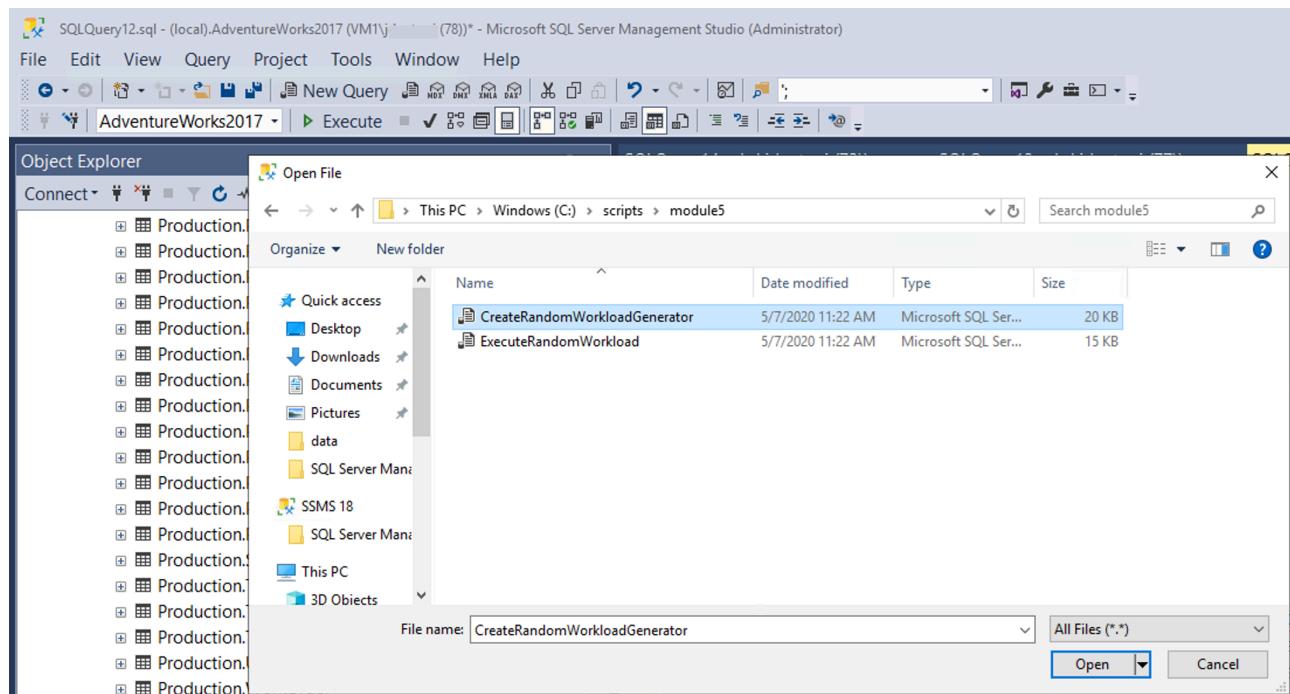
```
USE master;
GO

ALTER DATABASE AdventureWorks2017 SET QUERY_STORE = ON;
GO

ALTER DATABASE AdventureWorks2017 SET QUERY_STORE (OPERATION_MODE = READ_WRITE);
GO

ALTER DATABASE AdventureWorks2017 SET COMPATIBILITY_LEVEL = 100;
GO
```

2. From the menu in SQL Server Management Studio, select File > Open > File. Navigate to the D:\Labfiles\Query Performance\CreateRandomWorkloadGenerator.sql file. Click on the file to load it into Management Studio and then click Execute to execute the query.



3. Run a workload to generate statistics for Query Store. Navigate to the D:\Labfiles\Query Performance\ExecuteRandomWorkload.sql script to execute a workload. Click execute to run the script. After execution completes, run the script a second time to generate more data. Leave the query tab open for this query.
4. Copy and paste the code below into a new query window and execute it by clicking Execute. This script changes the database compatibility mode using the below script to SQL Server 2019 (150).

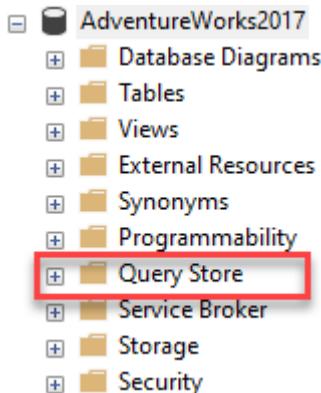
```
USE master;
GO
```

```
ALTER DATABASE AdventureWorks2017 SET COMPATIBILITY_LEVEL = 150;
GO
```

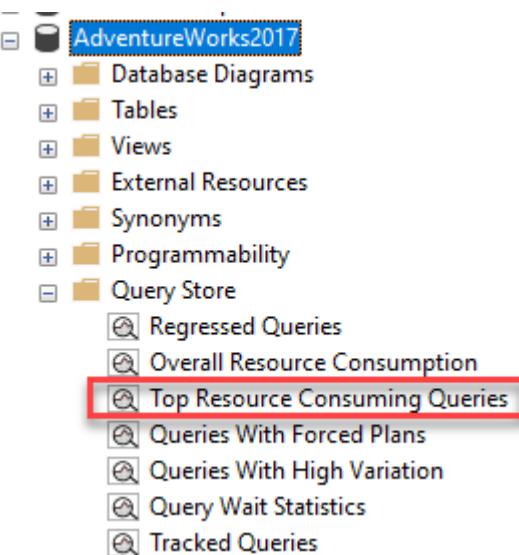
5. Navigate back to the query tab from step 3 (ExecuteRandomWorkload.sql), and re-execute.

## 35.2 Task 2: Examine Top Resource Consuming Queries to identify poor performance

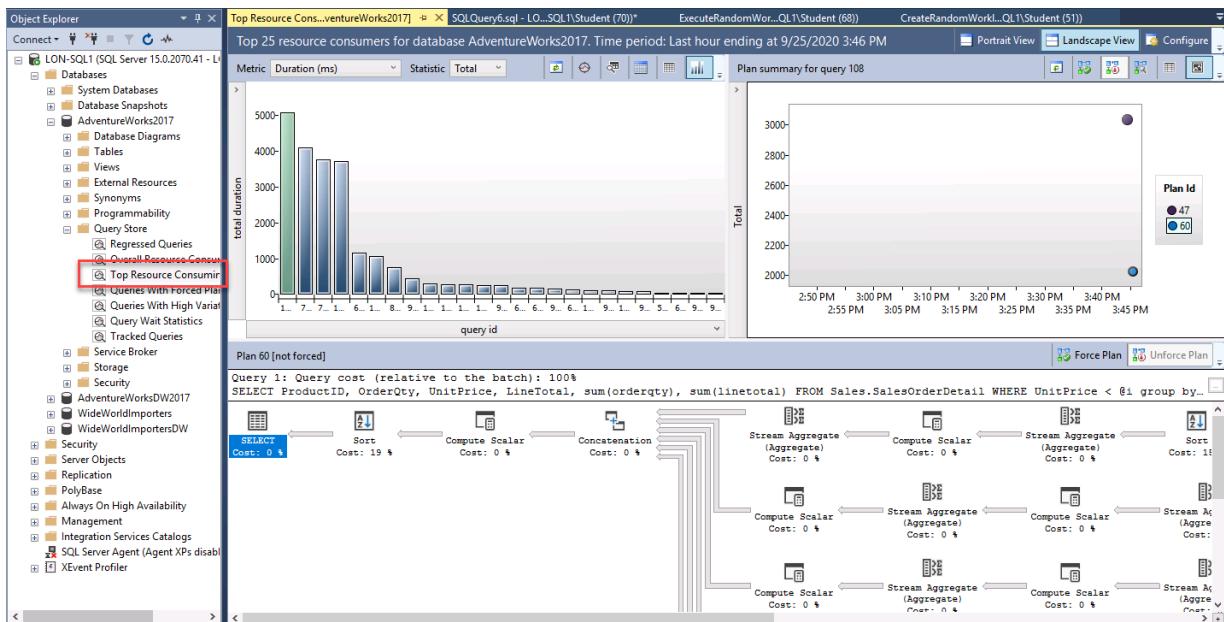
1. In order to view the Query Store you will need to refresh the AdventureWorks2017 database in Management Studio. Make sure that Management Studio is connected to the local SQL Server instance. Right click on database name and choose click refresh. You will then see the Query Store option under the database.



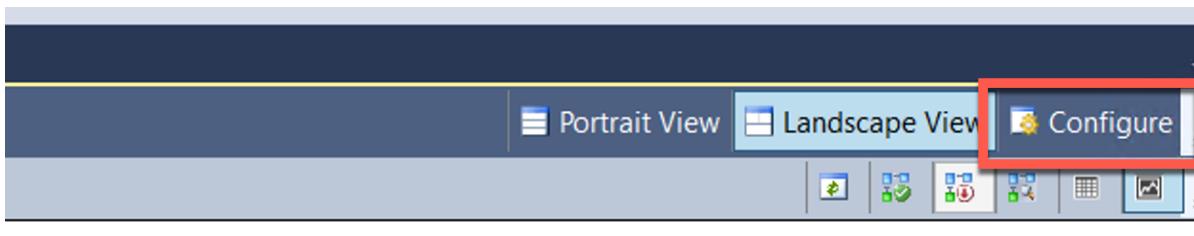
2. Expand Query Store node to view all available reports. Click on plus sign to expand Query Store reports. Select the Top Resource Consuming Queries Report by double-clicking on it.



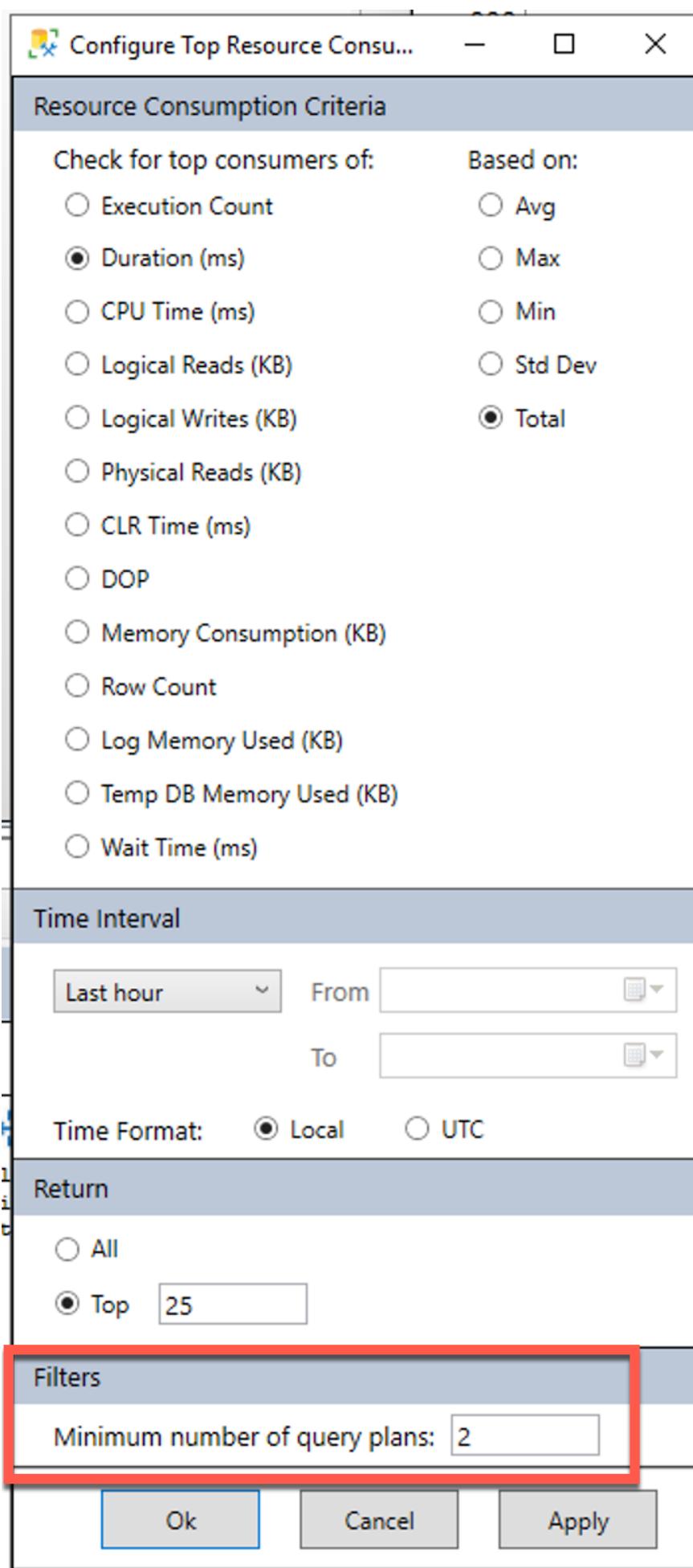
The report will open as shown below.



Max sure that SQL Server Management Studio is maximized in size. Click configure in the top right.



In the configuration screen, change the filter for the minimum number of query plans to 2. Click Apply and close the configuration window.

Configure Top Resource Consum... X

**Resource Consumption Criteria**

Check for top consumers of:  Execution Count  Duration (ms)  CPU Time (ms)  Logical Reads (KB)  Logical Writes (KB)  Physical Reads (KB)  CLR Time (ms)  DOP  Memory Consumption (KB)  Row Count  Log Memory Used (KB)  Temp DB Memory Used (KB)  Wait Time (ms)

Based on:  Avg  Max  Min  Std Dev  Total

**Time Interval**

Last hour

Time Format:  Local  UTC

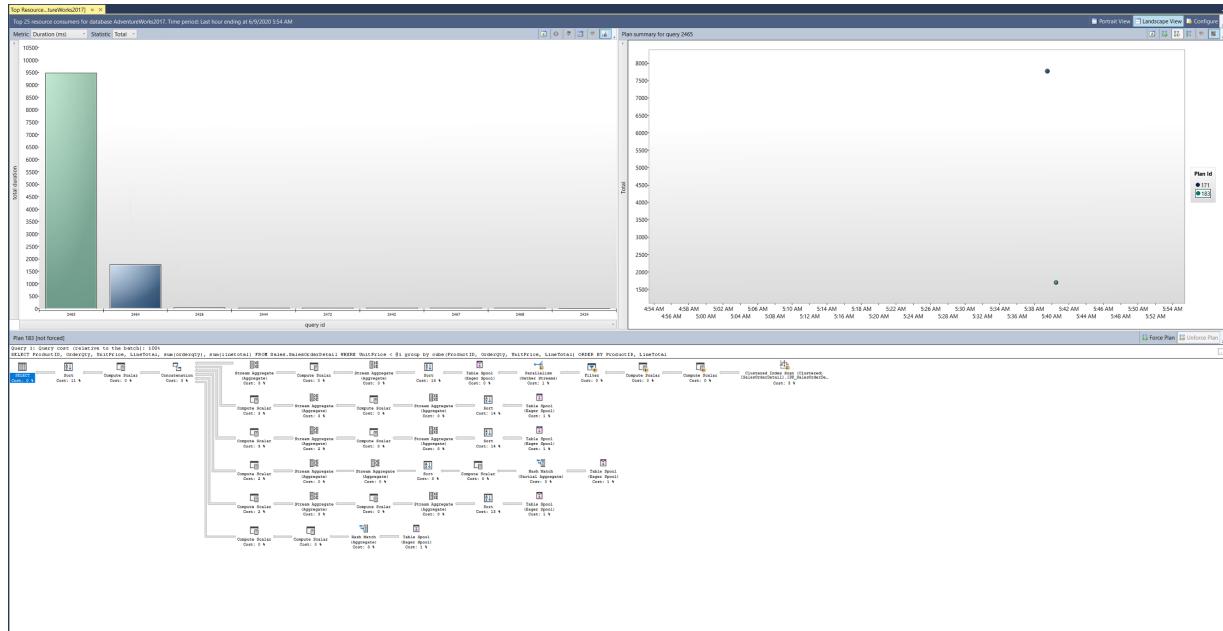
**Return**

All  Top

**Filters**

Minimum number of query plans:

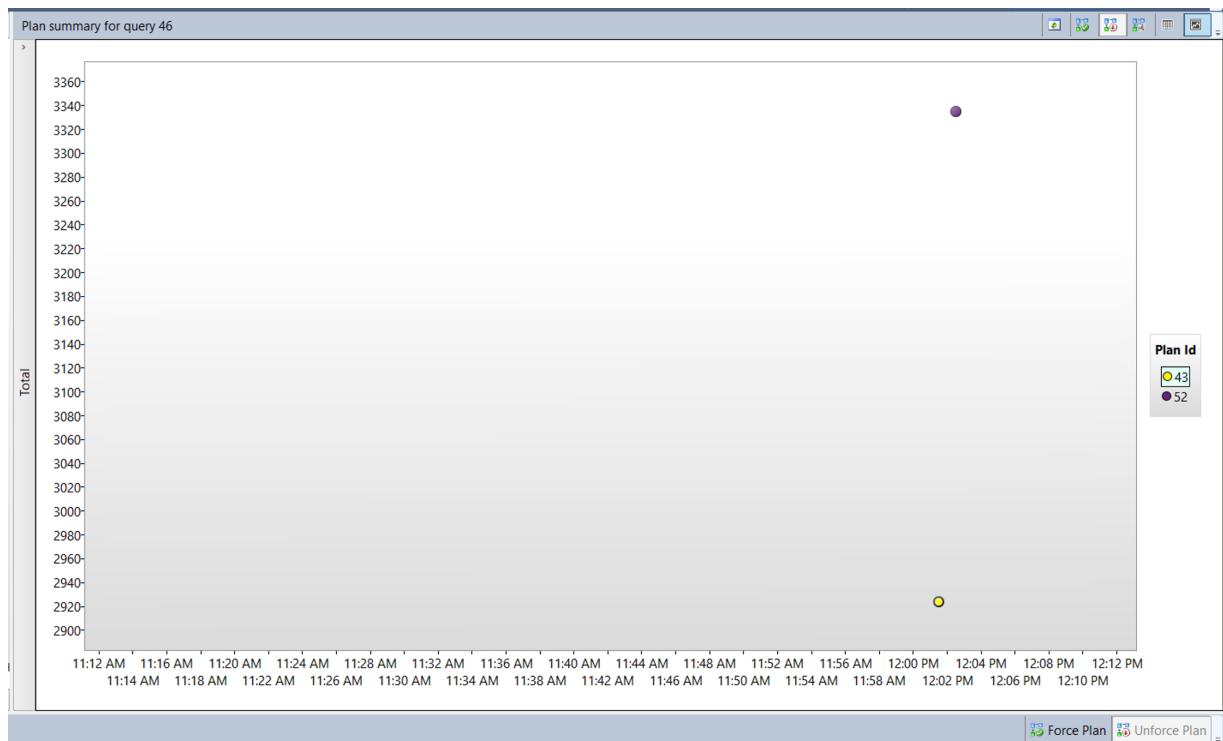
- Choose the query with the longest duration by clicking on the left most bar in the bar chart in the top left portion of the report.



This will show you the query and plan summary for your longest duration query in your query store.

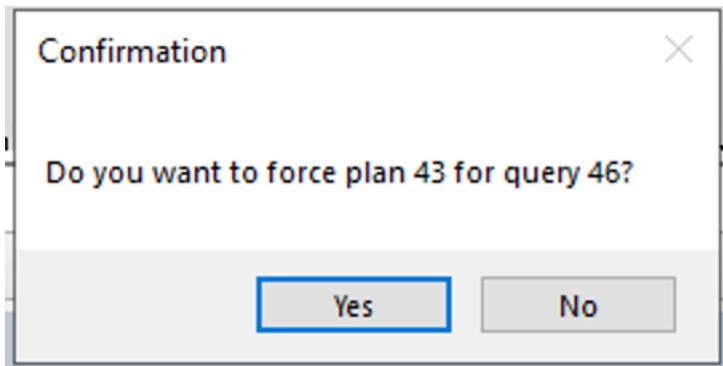
### 35.3 Task 3: Force a better execution plan

- Navigate to the plan summary portion of the report as shown below. You will note there are two execution plans with widely different durations.

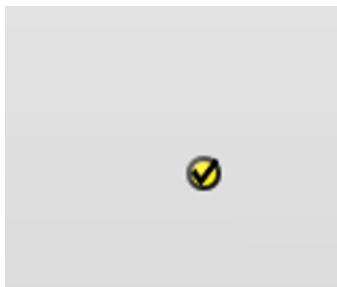


- Click on the Plan ID with the lowest duration (this is indicated by a lower position on the Y-axis of the chart) in the top right window of the report. In the graphic above, it's PlanID 43, but you'll probably have a different PlanID. Click on the plan ID next to the Plan Summary chart (it should be highlighted like in the above screenshot).
- Click on **Force Plan** under the summary chart. A confirmation window will popup, choose Yes to force

the plan.



Once forced you will see that the Forced Plan is now greyed out and the plan in the plan summary window now has a check mark indicating it is forced.



4. Close all your query windows, and the Query Store report.

## 36 Exercise 4: Use query hints to impact performance in AdventureWorks2017

Estimated Time: 15 minutes

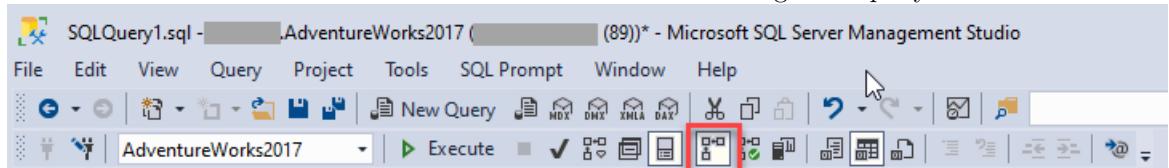
The main task for this exercise is as follows:

1. Run a workload.
2. Change query to use a parameter
3. Apply query hint to query to optimize for a value and re-execute.

### 36.1 Task 1: Run a workload

1. Open a New Query window in SQL Server Management Studio.

- Click on Include Actual Execution Plan icon before running the query or use CTRL+M.



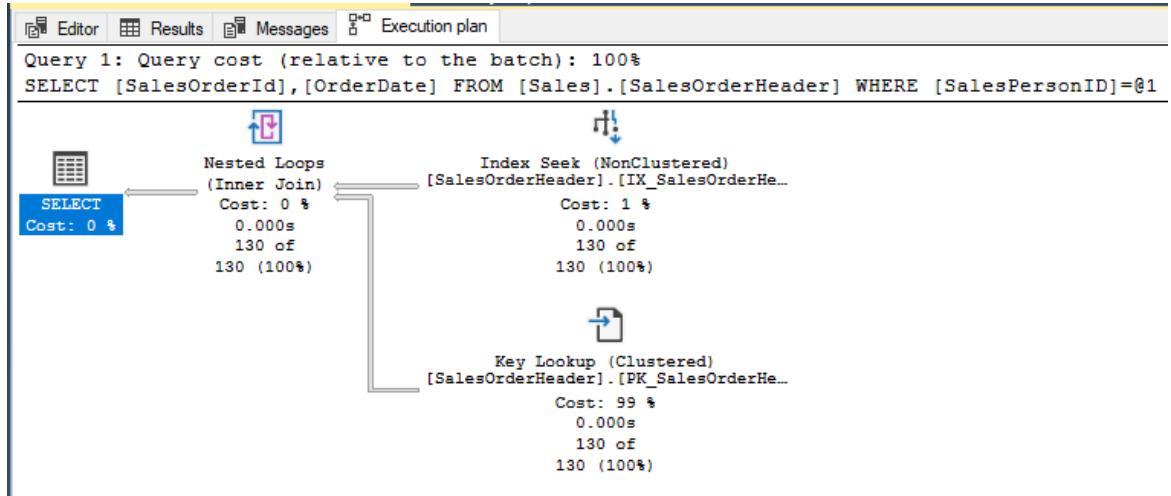
- Execute the query below. Note that the execution plan shows an index seek operator.

```
USE AdventureWorks2017
```

```
GO
```

```
SELECT SalesOrderID, OrderDate  
FROM Sales.SalesOrderHeader
```

```
WHERE SalesPersonID=288;
```

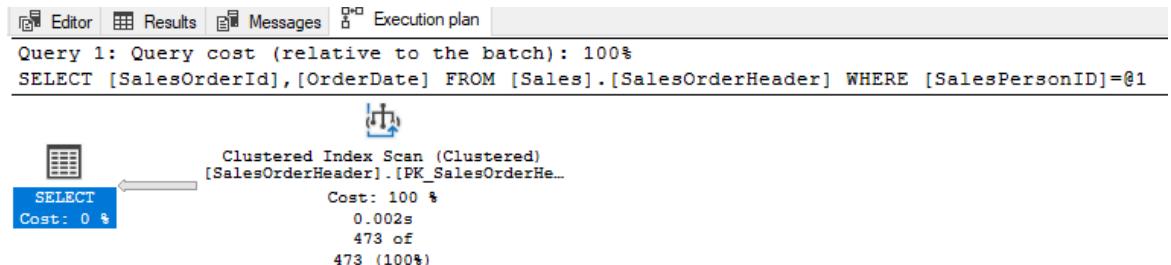


- Now run the next query. The only change is that the SalesPersonID value being searched for is 277. Note the Clustered Index Scan operation in the execution plan.

```
USE [AdventureWorks2017]
```

```
GO
```

```
SELECT SalesOrderId, OrderDate  
FROM Sales.SalesOrderHeader  
WHERE SalesPersonID=277;
```



Based on the column statistics the query optimizer has chosen a different execution plan because of the different value used in the WHERE clause. Because this query uses a constant in its WHERE clause, the optimizer sees each of these queries as unique and generates a different execution plan for each one.

### 36.2 Task 2: Change the query to use a parameterized stored procedure and use a Query Hint

- Open a New Query window. Click on Include Actual Execution Plan or use CTRL+M.
- Create a parameterized stored procedure so that the value to be searched for can be passed as a parameter instead of a hard-coded value in the WHERE clause. You should ensure that the data type of your parameter matches the data type of the column in the target table. Copy and execute the code below.

```
USE AdventureWorks2017
```

```
GO
```

```
CREATE OR ALTER PROCEDURE getSalesOrder  
@PersonID INT  
AS  
SELECT SalesOrderId, OrderDate  
FROM Sales.SalesOrderHeader
```

```
WHERE SalesPersonID = @PersonID  
GO
```

3. Call the procedure with a parameter value of 277. Copy and execute the following code:

```
EXEC getSalesOrder 277  
GO
```

As we saw before, when running the non-parameterized SELECT statement looking for a value of 277, the plan shows the use of a Clustered Index Scan.

4. Run the procedure again with a parameter value of 288. Copy and execute the following code:

```
EXEC getSalesOrder 288  
GO
```

If you examine the execution plan, you will note it is the same as it was for the value of 277. This is because SQL Server has cached the execution plan and is reusing it for the second execution of the procedure. Note that although the same plan is used for both queries, it is not necessarily the best plan.

5. Execute the following command to clear the plan cache for the AdventureWorks2017 database

```
USE AdventureWorks2017  
GO  
ALTER DATABASE SCOPED CONFIGURATION CLEAR PROCEDURE_CACHE;  
GO
```

5. Run the procedure again with a parameter value of 288. Copy and execute the following code:

```
EXEC getSalesOrder 288  
GO
```

You should notice the plan is now using the Nonclustered Index Seek operation. This is because the cached plan was removed and a new plan was created based on the new initial parameter value of 288.

6. Now recreate the stored procedure with a Query Hint. Because of the OPTION hint, the optimizer will create a plan based on a value of 288 and that plan will be used no matter what parameter value is passed to the procedure. Execute the procedure multiple times and note that it now always uses the plan with the Nonclustered Index Seek. Try calling with the procedure with parameter values we haven't seen yet, and you'll notice that no matter how many rows are returned (or no rows are returned!) the plan will always use the Nonclustered Index Seek.

```
USE AdventureWorks2017  
GO  
  
CREATE OR ALTER PROCEDURE getSalesOrder  
@PersonID INT  
AS  
SELECT SalesOrderID, OrderDate  
FROM Sales.SalesOrderHeader  
WHERE SalesPersonID = @PersonID  
OPTION (OPTIMIZE FOR (@PersonID = 288));  
GO
```

```
EXEC getSalesOrder 288;  
GO
```

```
EXEC getSalesOrder 277;  
GO
```

```
EXEC getSalesOrder 200;  
GO
```

7. Close all your query windows.

### 36.3 lab: title: 'Lab 6 – Automate Resources' module: 'Automation of Tasks'

## 37 Lab 6 – Automate Resources

**Estimated Time:** 90 minutes

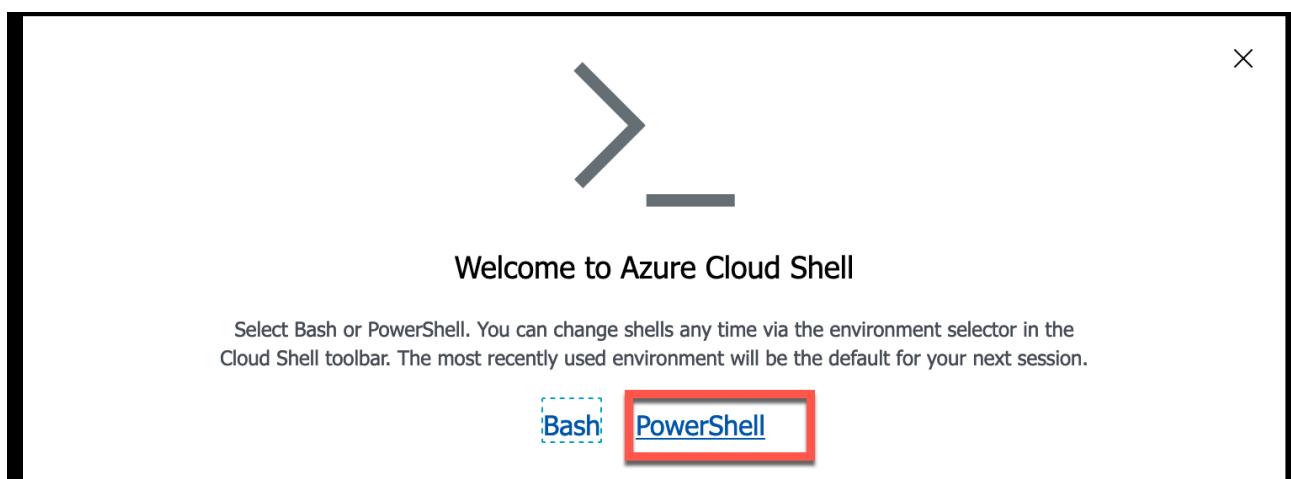
This lab can be performed from a web browser with access to the Azure portal.

Note: the Microsoft.Insights module needs to be added to your subscription in order to complete this lab. You can register by completing the following steps.

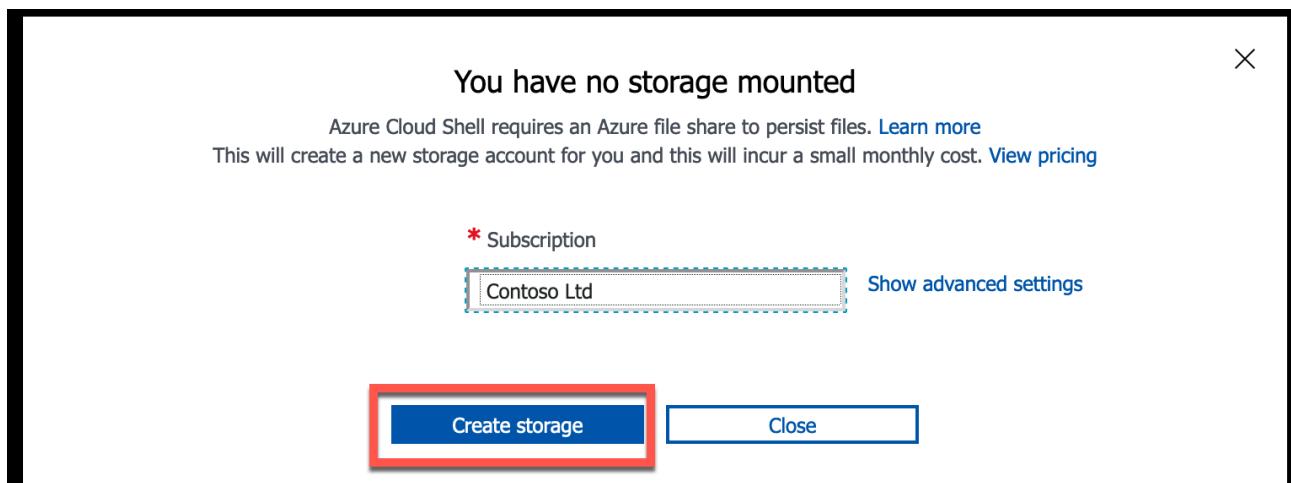
From the Azure portal, click on the cloud shell icon on the top right of the portal.



A shell will open at the bottom of the screen. Click on PowerShell as shown below.

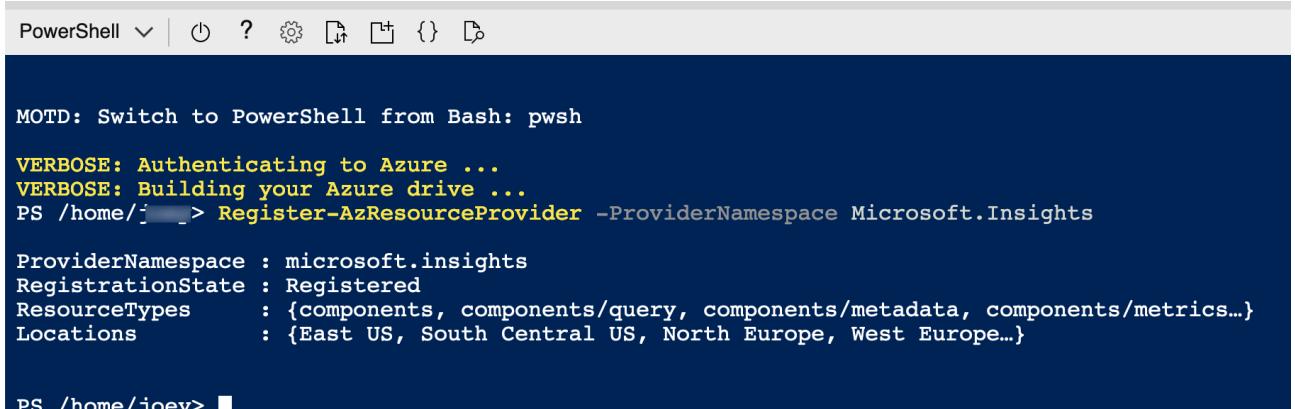


You may be prompted to create a storage account. Click Create Storage.



After the PowerShell window opens, paste in the following command:

```
register-AzResourceProvider -ProviderNamespace Microsoft.Insights
```



```
MOTD: Switch to PowerShell from Bash: pwsh

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/joev> Register-AzResourceProvider -ProviderNamespace Microsoft.Insights

ProviderNamespace : microsoft.insights
RegistrationState : Registered
ResourceTypes    : {components, components/query, components/metadata, components/metrics...}
Locations        : {East US, South Central US, North Europe, West Europe...}

PS /home/joev>
```

**Lab files:** The files for this lab are in the D:\LabFiles\Automate Resources folder.

## 38 Lab overview

The students will take the information gained in the lessons to configure and subsequently implement automate processes within AdventureWorks.

## 39 Lab objectives

After completing this lab, you will be able to:

- Deploy an Azure resource from a GitHub Quickstart template
- Configure performance metric related notifications
- Deploy an Azure Automation Runbook to perform index maintenance in an Azure SQL Database

## 40 Scenario

You have been hired as a Senior Data Engineer to help automate day to day operations of database administration. This automation is to help ensure that the databases for AdventureWorks continue to operate at peak performance as well as provide methods for alerting based on certain criteria. AdventureWorks utilizes SQL Server in both Infrastructure as a Service and Platform as a Service offerings.

## 41 Exercise 1: Deploy an Azure Quickstart Template

Estimated Time: 30 minutes

Individual Exercise

The main task for this exercise are as follows:

- Deploy an Azure resource from a GitHub Quickstart template

### 41.1 Task: Deploy an Azure SQL Database from a template

1. Navigate to the following GitHub using a web browser.

<https://github.com/Azure/azure-quickstart-templates/tree/master/201-sql-database-transparent-encryption-create>

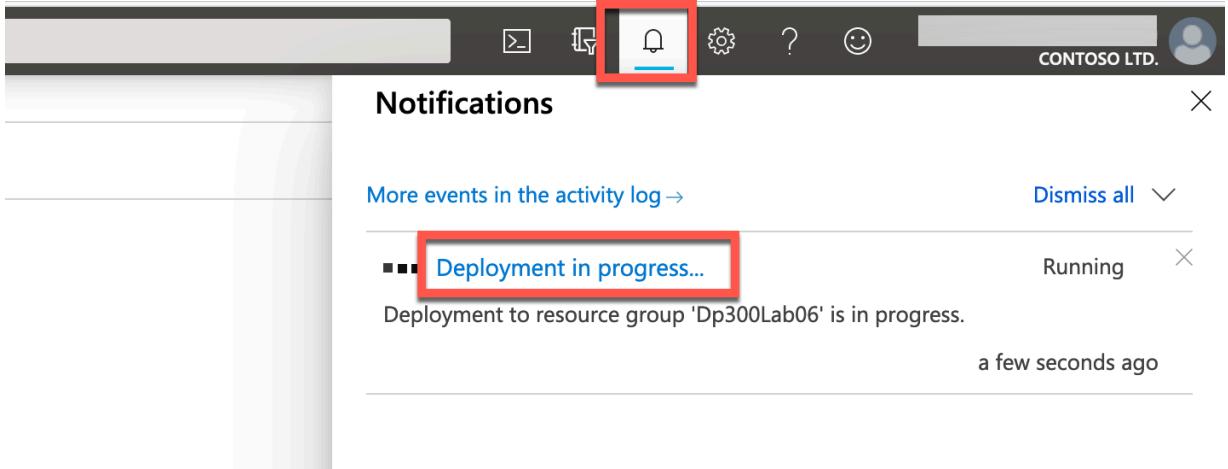
Click on `azuredeploy.json`, and review the file.

2. Navigate back to the above link (click the back arrow in the top left of the browser), and click on the Deploy to Azure button. You may be prompted to login to the Azure portal. Login with your supplied credentials.
3. You will see a screen like the one below. In order to deploy this template, you need to complete the blank fields.

In the Resource Group field, click “Create New” and type **DP300-Lab06**, and then click OK. Change the region to a region near you. For the remaining fields, use the following values:

- SQL Administrator Login: **labadmin**
- SQL Administrator Login Password: **Azur3Pa\$\$**
- Transparent Data Encryption: **Enabled**
- Location: **[resourceGroup().location]**

- Click Review + create, then click Create. Your deployment will begin. You can track the status of your deployment by clicking the bell (highlighted in the screenshot below) and then clicking on the Deployment in progress link in the Notifications pane.



Your deployment will take approximately 5-10 minutes to deploy. If you have clicked on the link above, you will be able to track your deployment.

This screenshot shows the 'Microsoft.Template | Overview' page. The left sidebar has 'Overview' selected. The main content area displays the message 'Your deployment is underway'. It shows deployment details: Deployment name: Microsoft.Template, Subscription: Contoso Ltd, Resource group: DP-300-RG. To the right, it shows start time: 5/10/2020, 12:14:23 PM and Correlation ID: 1a9dbc0b-8ac6-4ad3-9d1a-ec0dbaa8dda8. Below this, there is a table titled 'Deployment details' with one row: Resource: sqlservers4dvuy42dhvug, Type: Microsoft.Sql/servers, Status: Accepted, and a 'Operation details' link.

- Upon completion, the screen will update with a link to your newly created resource.

This screenshot shows the same 'Microsoft.Template | Overview' page after the deployment has completed. The main content area now displays the message 'Your deployment is complete' with a green checkmark icon. The deployment details remain the same. Below the message, there is a 'Next steps' section with a 'Go to resource' button.

Click on the Go to resource link. You will be taken to the Azure Resource Group your deployment just created. You should see both a SQL server and a SQL database in the Resource group. Make note of the name of your SQL server, as you will it in the next exercise.

## 42 Exercise 2: Configure Performance Metrics Based Alerts

Estimated Time: 30 minutes

Individual Exercise

The main task for this exercise is as follows:

- Configure performance metric related notifications

### 42.1 Task: Creating an alert when CPU exceeds an average of 80 percent.

1. Navigate to portal.azure.com if you are not already there. You may need to login again. In the search bar at the top of the Azure Portal, type SQL, and click on SQL databases. Click on the database name: sample-db-with-tde.

2. On the main blade for the sample-db-with-tde database, navigate down to the monitoring section
3. Click on Alerts as shown below.

sample-db-with-tde (sqlserverzb3h576ea23y2/sample-db-with-tde)

SQL database

Search (Cmd+ /) <

[Copy](#) [Restore](#) [Export](#) [Set server firewall](#) [Delete](#) [Connect with](#)

Resource group (change) : Dp300Lab06

Status : Online

Location : East US 2

Subscription (change) : Contoso Ltd

Subscription ID : 424d0f78-5980-4d31-98ec-624616db8e74

Tags (change) : displayName : Database

Show data for last: [1 hour](#) [24 hours](#) [7 days](#)

**Compute utilization**

DTU percentage (Max)  
sqlserverzb3h576ea23y2/sample-db-with-tde  
**3 %**

**Database data storage** ⓘ

Used space **4 MB**  
Allocated space **16 MB**

Notifications (0) [Data](#)  
[All](#) [Security \(4\)](#)  
[Transparent data](#)

**Alerts** (highlighted with a red box)

[Metrics](#)  
[Diagnostic settings](#)  
[Logs](#)

4. You'll see the screen below. Click on + New alert rule.

sample-db-with-tde (sqlserverzb3h576ea23y2/sample-db-with-tde) | Alerts

[+ New alert rule](#) [Manage alert rules](#) [Manage actions](#) [View classic alerts](#) [Refresh](#) [Provide feedback](#)

Don't see a subscription? [Open Directory + Subscription settings](#)

Subscription \* [Contoso Ltd](#) Resource group [Dp300Lab06](#) Resource [sqlserverzb3h576ea23y2/sample-db...](#)

Selected subscriptions > Dp300Lab06 > sample-db-with-tde

**Pay attention to what matters.**

You have not configured any alert rules.

Configure alert rules and attend to fired alerts to efficiently monitor your Azure resources. [Learn More](#)



[+ New Alert Rule](#)

The classic alerts can be accessed from [here](#).

5. You will notice that the resource is already populated for you with the SQL Server you created. Under the Condition section, click Add condition.

Dashboard > Microsoft.Template-20210305131841 > DP300-Lab06 > sqlverkabtqduqwsdua > sample-db-with-tde (sqlverkabtqduqwsdua/sample-db-with-tde) >

## Create alert rule [...](#)

Rules management

Create an alert rule to identify and address issues when important conditions are found in your monitoring data. [View tutorial + read more](#)  
When defining the alert rule, check that your inputs do not contain any sensitive content.

### Scope

Select the target resource you wish to monitor.

Resource  
 [sqlverkabtqduqwsdua/sample-db-with-tde](#)

Hierarchy

 [MSDN Platforms](#) >  [dp300-lab06](#)

[Edit resource](#)

### Condition

Configure when the alert rule should trigger by selecting a signal and defining its logic.

Condition name

No condition selected yet

[Add condition](#)

### Actions

Send notifications or invoke actions when the alert rule triggers, by selecting or creating a new action group. [Learn more](#)

Action group name

Contains actions

[Create alert rule](#)

6. In the Configure signal logic fly out menu, select CPU percentage.

## Configure signal logic

Choose a signal below and configure the logic on the next screen to define the alert condition.

Signal type ⓘ

Metrics

Monitor service ⓘ

All

Displaying 1 - 20 signals out of total 23 signals

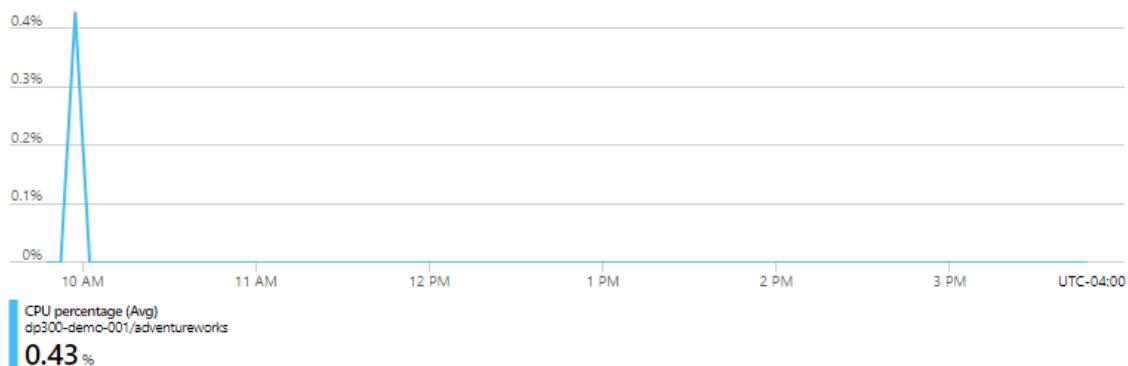
Search by signal name

Signal name	↑↓	Signal type	↑↓	Monitor service ↑↓
CPU percentage		Metric		Platform
Data IO percentage		Metric		Platform
Log IO percentage		Metric		Platform
Data space used		Metric		Platform

- Supply a threshold value of 80. Click Done.

## Configure signal logic

Over the last 6 hours ▾



### Alert logic

Threshold ⓘ

Static

Dynamic

Operator ⓘ

Greater than

Aggregation type \* ⓘ

Average

Threshold value \* ⓘ

80

%

### Condition preview

Whenever the cpu percentage is greaterthan 80 %

### Evaluated based on

Aggregation granularity (Period) \* ⓘ

5 minutes

Frequency of evaluation ⓘ

Every 1 Minute

**Done**

- Under Action group section, click Add action groups

Home > Microsoft.Template | Overview > Dp300Auto > sqlservermddgovwp6mry > sample-db-with-tde (sqlservermddgovwp6mry/sample-db-with-tde) | Alerts >

## Create alert rule

Rules management

Create an alert rule to identify and address issues when important conditions are found in your monitoring data. [Learn more](#)  
When defining the alert rule, check that your inputs do not contain any sensitive content.

### Scope

Select the target resource you wish to monitor.

Resource	Hierarchy
 sqlservermddgovwp6mry/sample-db-with-tde	 Contoso Ltd >  Dp300Auto
<a href="#">Edit resource</a>	

### Condition

Configure when the alert rule should trigger by selecting a signal and defining its logic.

#### Condition name

No condition selected yet

[Select condition](#)

### Action group

Send notifications or invoke actions when the alert rule triggers, by selecting or creating a new action group. [Learn more](#)

Action group name	Contains actions
No action group selected yet	
<a href="#">Select action group</a>	

### Alert rule details

Provide details on your alert rule so that you can identify and manage it later.

Alert rule name \* 

Specify the alert rule name

Description

Specify the alert rule description

Enable alert rule upon creation 

9. In the fly out for Action group, Click + Create action group.

## Select an action group to attach to this alert rule

The action group selected will attach to this alert rule

 [Create action group](#)

In the Create action group screen, type emailgroup in the Action group name field.

## Create action group

Basics Notifications Actions Tags Review + create

An action group invokes a defined set of notifications and actions when an alert is triggered. [Learn more](#)

### Project details

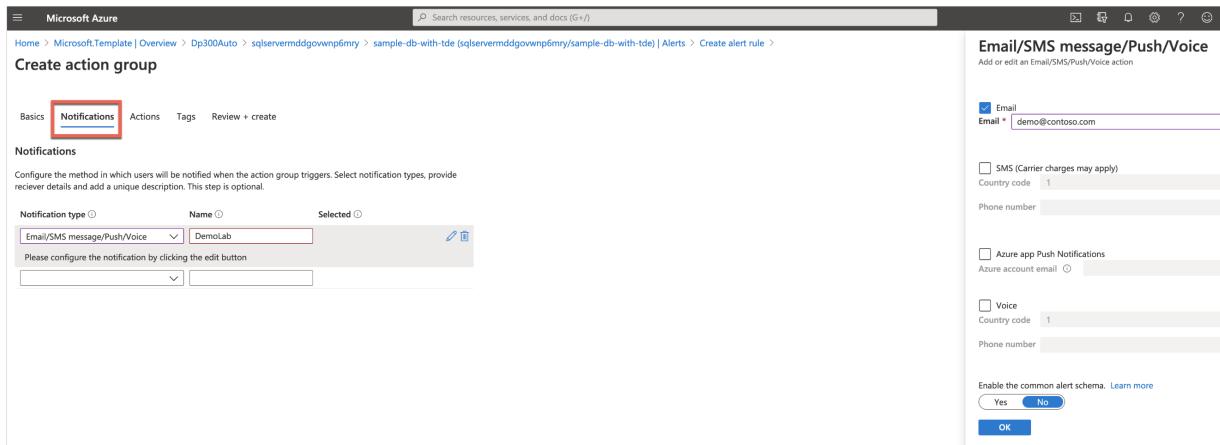
Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

<b>Subscription *</b> ⓘ 	<input type="text" value="Contoso Ltd"/> 
<b>Resource group *</b> ⓘ 	<input type="text" value="Dp300Auto"/>  <a href="#">Create new</a>

### Instance details

<b>Action group name *</b> ⓘ 	<input type="text" value="emailgroup"/> 
<b>Display name *</b> ⓘ 	<input type="text" value="emailgroup"/>  <small>This display name is limited to 12 characters</small>

10. Then click on the notifications tab.



The screenshot shows the Azure portal interface for creating an action group. The 'Notifications' tab is highlighted. On the right, a modal dialog titled 'Email/SMS message/Push/Voice' is open, showing fields for 'Email' (with value 'demo@contoso.com') and 'SMS' (with checked 'Carrier charges may apply' checkbox). Below these are fields for 'Azure app Push Notifications' and 'Voice'. At the bottom of the modal is an 'OK' button.

11. Select Email/SMS message/Push/Voice and enter the name DemoLab. In the flyout screen on the right (shown in the image above), click the check box next to email, enter an email address and click OK. Then click on the Review + create button, and then click Create on the Create action group screen.

From the Create alert rule screen, add an Alert rule with the name DemoAlert, and then click Create alert rule as show below.

**Create alert rule**

Rules management

Create an alert rule to identify and address issues when important conditions are found in your monitoring data. [Learn more](#)  
When defining the alert rule, check that your inputs do not contain any sensitive content.

**Scope**

Select the target resource you wish to monitor.

Resource	Hierarchy
sqlservermddgovnp6mry/sample-db-with-tde	Contoso Ltd > Dp300Auto

[Edit resource](#)

**Condition**

Configure when the alert rule should trigger by selecting a signal and defining its logic.

Condition name	Estimated monthly cost (USD) ⓘ
Whenever the average cpu percentage is greater than 80 %	\$ 0.10

[Select condition](#) Total \$ 0.10

**Action group**

Send notifications or invoke actions when the alert rule triggers, by selecting or creating a new action group. [Learn more](#)

Action group name	Contains actions
emailgroup	1 Email ⓘ

[Select action group](#)

**Alert rule details**

Provide details on your alert rule so that you can identify and manage it later.

Alert rule name * ⓘ	DemoAlert
Description	Specify the alert rule description
Severity * ⓘ	Sev 3
Enable alert rule upon creation	<input checked="" type="checkbox"/>

[Create alert rule](#)

## 43 Exercise 3: Deploy an Automation Runbook

Estimated Time: 30 minutes

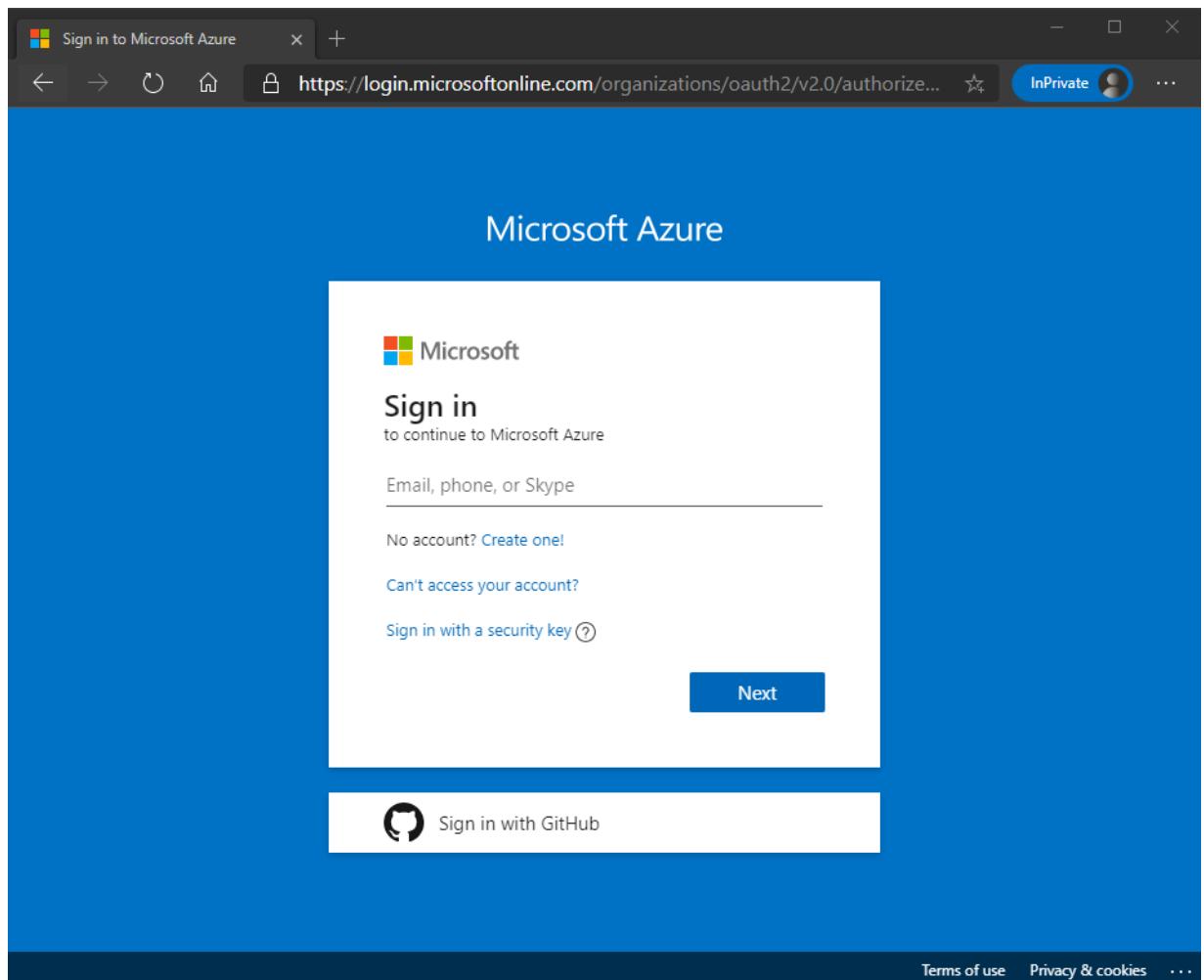
Individual Exercise

The main tasks for this exercise are as follows:

- Deploy a maintenance task script.
- Deploy an Azure Automation Runbook to perform index maintenance in an Azure SQL Database

### 43.1 Task: Deploy an Automation Runbook to rebuild indexes in an Azure SQL Database.

1. From the lab virtual machine, if you are not in the Azure portal, start a browser session and navigate to <https://portal.azure.com>. Provide appropriate credentials.



2. In the Azure Portal in the search bar type Automation and then click on Automation Accounts

A screenshot of the Microsoft Azure portal. The search bar at the top contains the text 'auto'. Below the search bar, the 'Services' section is visible, showing a card for 'Automation Accounts' with a gear icon.

Click on the + Add button in the portal.

A screenshot of the 'Automation Accounts' list in the Azure portal. The 'Add' button is highlighted with a red box. The list shows one record: 'Contoso Ltd.' with a status of 'Active'. There are buttons for 'Manage view', 'Refresh', 'Export to CSV', 'Assign tags', and 'Feedback'. Filter options at the bottom include 'Filter by name...', 'Subscription == all', 'Resource group == all', 'Location == all', and 'Add filter'.

Showing 1 to 3 of 3 records.

3. Create your automation account

## Add Automation Account

Name \* ⓘ  
DP300Lab06

Subscription \*  
Azure Pass - Sponsorship

Resource group \*  
DP300-Lab06

Create new

Location \*  
West US

Create Azure Run As account \* ⓘ  
Yes No

This will create Azure Run As account in the Automation account which are useful for authenticating with Azure to manage Azure resources from Automation runbooks. Note that the creation of Azure Run As account may affect the security of the subscription.[Learn more](#)

**Create**

Name: **DP300Lab06**

Resource Group: Use the resource group you created earlier in this lab. **DP300-Lab06**

Location: Use the region in which you created your Azure SQL server in Lab 2

Click **Create**. Your automation account should deploy in 1-3 minutes.

4. Navigate to the **github** page for AdaptiveIndexDefragmentation. [https://github.com/microsoft/tigertoolbox/blob/master/tigertoolbox/AdaptiveIndexDefrag/usp\\_AdaptiveIndexDefrag.sql](https://github.com/microsoft/tigertoolbox/blob/master/tigertoolbox/AdaptiveIndexDefrag/usp_AdaptiveIndexDefrag.sql)

Read the guide

microsoft / tigertoolbox

Branch: master

Issues 44 Pull requests 11 Actions Projects 0 Wiki Security 0 Insights

tigertoolbox / AdaptiveIndexDefrag / usp\_AdaptiveIndexDefrag.sql

5 contributors cd86c2b on Feb 27

3335 lines (2862 sloc) 187 KB

Raw Blame History

```
1 -- If you are using AdaptiveIndexDefrag together with the maintenance plans in http://blogs.msdn.com/b/blogdozequel/archive/2014/01/23/using-adaptive-index-defragmentation-with-maintenance-plans.aspx
2 -- please note that the job that runs AdaptiveIndexDefrag is expecting msdb. As such, change the database context accordingly.
3
4 -- For deployment in Azure SQL Database, remove or comment the USE statement below.
5 USE msdb
6 GO
7
8 SET NOCOUNT ON;
9
10 DECLARE @deploymode bit
11 SET @deploymode = 0 /* 0 = Upgrade from immediately previous version, preserving all historic data;
12                                1 = Rewrite all objects, disregarding historic data */
13
14 /* Scroll down to line 429 to see notes, disclaimers, and licensing information */
15
16 RAISERROR('Dropping existing objects', 0, 42) WITH NOWAIT;
17
18 IF EXISTS(SELECT [object_id] FROM sys.views WHERE [name] = 'vw_CurrentExecStats')
19 DROP VIEW vw_CurrentExecStats
20
21 IF EXISTS(SELECT [object_id] FROM sys.views WHERE [name] = 'vw_ErrLst30Days')
22 DROP VIEW vw_ErrLst30Days
23
24 IF EXISTS(SELECT [object_id] FROM sys.views WHERE [name] = 'vw_LastRun_Log')
25 DROP VIEW vw_LastRun_Log
```

Click on Raw. This will provide the code in a format where you can copy it. Select all of the text (Ctrl-A) and copy it to your clipboard (Ctrl-C).

5. In the Azure Portal, navigate back to your database and click on the Query editor as shown below.

sample-db-with-tde (sqlserverzb3h576ea23y2/sample-db-with-tde)

SQL database

Search (Cmd+ /) <>

Copy Restore Export Set server firewall Del

**Overview**

- Activity log
- Tags
- Diagnose and solve problems
- Quick start
- Query editor (preview)**

Resource group (change) : Dp300Lab06

Status : Online

Location : East US 2

Subscription (change) : Contoso Ltd

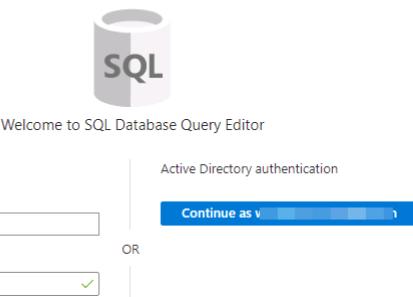
Subscription ID : 424d0f78-5980-4d31-98ec-624616db8e74

Tags (change) : displayName : Database

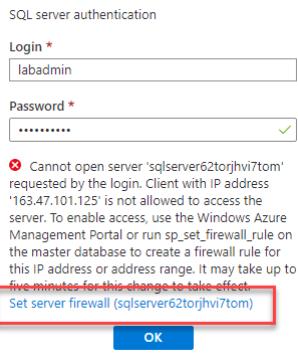
- You will be prompted for credentials to login to your database. Use the credentials you created in Exercise 1.

**Login:** labadmin

**Password:** Azur3Pa\$\$



You may receive an error that a firewall rule needs to be created as shown below.



Click on the blue Set server firewall link at the end of the error message as shown above. The server firewall screen will come up. Click + Add client IP as highlighted below.

Home > SQL databases > sample-db-with-tde (sqlservers4dvuy42dhvug/sample-db-with-tde) | Query editor (preview) > Firewall settings

**Firewall settings**  
sqlservers4dvuy42dhvug (SQL server)

Deny public network access  Yes  No

**Connection Policy**  Default  Proxy

Allow Azure services and resources to access this server  Yes

**Client IP address** 71.230.244.45

Rule name	Start IP	End IP	...
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="..."/>

No firewall rules configured.

**Information:** Connections from the IPs specified below provides access to all the databases in sqlservers4dvuy42dhvug.

Click the Save button to save the firewall configuration, and click OK on the Success popup.

Home > SQL databases > **sample-db-with-tde (sqlservers4dvuy42dhvug/sample-db-with-tde) | Query editor (preview)** > Firewall settings

**Firewall settings**  
sqlservers4dvuy42dhvug (SQL server)

**Success!**  
Successfully updated server firewall rules

After you see the success message above, click on the link to Query editor (outlined in red in the above image) above the firewall settings pane.

When you return to the query editor, click OK, to login to your database. Paste the text you copied earlier in this task into the Query 1 pane.

**Query 1** 

[Results](#)      [Messages](#)

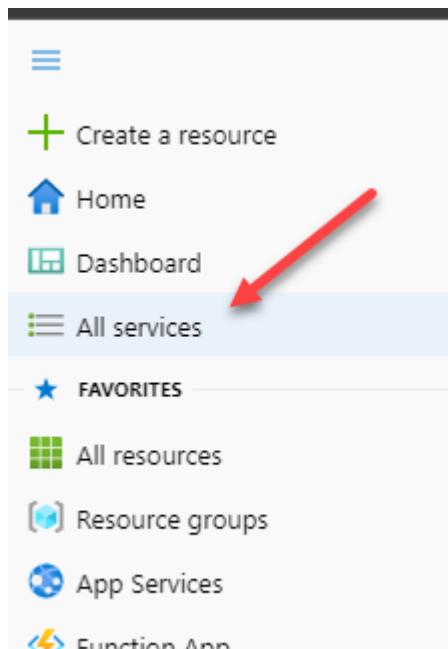
Delete the USE msdb and GO statements on lines 5 and 6 of the script (that are highlighted in the above screenshot) and then click Run.

 Results Messages

 Query succeeded | 18s

The query may take a few seconds to complete.

7. In the Azure portal, expand the left menu and click on All Services within the main Azure panel.



8. Type automation within the search field. Click on Automation Accounts.

A screenshot of the Microsoft Azure search results. The search bar contains 'automation'. Below it, a card for 'Automation Accounts' is shown, featuring a gear icon and the text 'Keywords: automate'. A red arrow points to this card.

Identify the automation account that has been created in your subscription and click on it.

A screenshot of the 'Automation Accounts' blade. The page title is 'Automation Accounts'. It shows one record: 'Contoso Ltd.' under 'Subscription == all', 'dp300lab06' under 'Resource group == all', and 'all' under 'Location == all'. The first record, 'DP300Lab06', is highlighted with a red box. At the bottom, there is a table with columns 'Name ↑' and 'Type ↑', showing the single record.

9. Select Modules from the Shared Resources section of the Automation blade.

**DP300Lab06** ↗

Automation Account

Delete Move Feedback Refresh

**Overview**  

- ☰ Activity log
- 👤 Access control (IAM)
- 🏷️ Tags
- 📝 Diagnose and solve problems

---

**Configuration Management**

- 📦 Inventory
- 📄 Change tracking
- 🔧 State configuration (DSC)

---

**Update management**

- 🕒 Update management

---

**Process Automation**

- 📄 Runbooks
- 📅 Jobs
- 📸 Runbooks gallery
- 👤 Hybrid worker groups
- ⚡ Watcher tasks

---

**Shared Resources**

- 🕒 Schedules
- ParallelGroups **Modules**
- 📸 Modules gallery

Resource gro... ([change](#)) : Dp300Lab06

Location : East US 2

Subscription ([change](#)) : Contoso Ltd

Tags ([change](#)) : [Click here to add tags](#)

---

**Job Statistics**  
Last 24 Hours

Status	Count
Failed	0
Suspended	0
Completed	0
Running	0
Queued	0
Stopped	0

10. Click on Browse Gallery

Home >

**DP300Lab06 | Modules** ↗

Automation Account

Add a module Update Azure modules Learn about module updates **Browse gallery** Refresh

**Overview**  

- ☰ Activity log
- 👤 Access control (IAM)

Name	Last modified	Status
AuditPolicyDsc	5/13/2020, 6:33 AM	Available

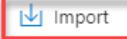
11. Search for sqlserver within the Gallery

Microsoft Azure  Home > Automation Accounts > DP300Lab06 | Modules >

## Browse Gallery

<b>SqlServer</b>	Created by: matteot_msft 4734130 downloads Last updated: 3/23/2020
This module allows SQL Server developers, administrators and business intelligence professionals to automate database development and server administration, as well as both multidimensional and tabular cube processing. Tags: <a href="#">SQL</a> <a href="#">SqlServer</a> <a href="#">SQLPS</a> <a href="#">Databases</a> <a href="#">SqlAgent</a> <a href="#">Jobs</a> <a href="#">SSAS</a> <a href="#">AnalysisServices</a> <a href="#">Tabular Cubes</a> <a href="#">SSIS</a> <a href="#">ExtendedEvents</a> <a href="#">xEvents</a> <a href="#">VulnerabilityAssessment</a> <a href="#">DataClassification</a> <a href="#">PSModule</a>	Created by: matteot_msft 4734130 downloads Last updated: 3/23/2020
<b>Write-ObjectToSQL</b>	Created by: JohnRoos 8690469 downloads Last updated: 2/11/2017
Writes an object into a database table. If the table does not exist it will be created based on the properties of the object. For every property of the object a column will be created. The data type for each column will be converted from .Net data types into Tags: <a href="#">Database</a> <a href="#">SQLServer</a> <a href="#">Object</a> <a href="#">PSModule</a>	Created by: matteot_msft 4734130 downloads Last updated: 3/23/2020

**SqlServer**  
PowerShell Module

 Import 

This module allows SQL Server developers, administrators and server administration, as well as both multidimensional and tabular cube processing.

For Release Notes, expand the Package Details section

For automation with SSRS & Power BI Report Server, <https://www.powershellgallery.com/packages/Report>

**Created by:** matteot\_msft  
**Tags:** [SQL](#) [SqlServer](#) [SQLPS](#) [Databases](#) [SqlAgent](#) [Jobs](#) [SSAS](#) [AnalysisServices](#) [Tabular Cubes](#) [SSIS](#) [ExtendedEvents](#) [xEvents](#) [VulnerabilityAssessment](#) [DataClassification](#) [PSModule](#)

**Learn more**  
[View in PowerShell Gallery](#)  
[Licensing information](#)

**Content**

Type	Name

12. Click on the SqlServer text which will direct to the next screen. Click on Import
13. Click on Ok in the bottom corner of the screen in order to import the module. This will import the Power-

**SqlServer** 

This module allows SQL Server developers, administrators and business intelligence professionals to automate database development and server administration, as well as both multidimensional and tabular cube processing.

For Release Notes, expand the Package Details section on this page.

For automation with SSRS & Power BI Report Server, use the ReportingServicesTools module available at <https://www.powershellgallery.com/packages/ReportingServicesTools>

**Created by:** matteot\_mst **Version:** 21.1.18221  
**Tags:** SQL SqlServer SQLPS Databases SqlAgent Jobs SSAS AnalysisServices Tabular Cubes SSIS ExtendedEvents xEvents VulnerabilityAssessment DataClassification PSModule **Downloads:** 4,296,743 **Last updated:** 3/23/2020

**Learn more**  
[View in PowerShell Gallery](#)  
[Licensing information](#)

**Content**

Type	Name
Cmdlet	Add-RoleMember
Cmdlet	Add-SqlAvailabilityDatabase
Cmdlet	Add-SqlAvailabilityGroupListenerStaticIp
Cmdlet	Add-SqlAzureAuthenticationContext
Cmdlet	Add-SqlColumnEncryptionKeyValue
Cmdlet	Add-SqlFirewallRule
Cmdlet	Add-SqlLogin
Cmdlet	Backup-ASDatabase
Cmdlet	Backup-SqlDatabase
Cmdlet	Complete-SqlColumnMasterKeyRotation
Cmdlet	ConvertFrom-EncodedSqlName
Cmdlet	ConvertTo-EncodedSqlName

Shell module into your Automation account.

14. You will need to create a credential to securely login to your database. From the Automation Account blade navigate to the Shared Resources Section and click Credentials.

## DP300Lab06 | Credentials

Automation Account

   Add a credential  Refresh

 Hybrid worker groups

 Watcher tasks

**Shared Resources**

-  Schedules
-  Modules
-  Modules gallery
-  Python 2 packages
-  **Credentials**

**Name**

No credentials found.

Click + Add a credential, and then create a credential as shown below. Name the credential SQLUser

and use the labadmin username and Azur3Pa\$\$ for the password. Click Create.

The screenshot shows the 'New Credential' dialog box. On the left, there is a list of credentials with columns for Name, User name, and Last modified. A message says 'No credentials found.' On the right, the 'New Credential' form is displayed:

- Name \***: SQLUser
- Description**: (empty)
- User name \***: labadmin
- Password \***: (redacted)
- Confirm password \***: (redacted)

A blue 'Create' button is at the bottom right of the dialog.

15. Scroll to the Process Automation section of the Automation account blade and click on Runbooks and then Create a runbook.

The screenshot shows the 'Runbooks' blade for the automation account 'DP300Lab06'. The top navigation bar includes 'Search (Cmd+ /)', 'Create a runbook' (which is highlighted with a red box), 'Import a runbook', 'Browse gallery', 'Learn more', and 'Refresh'. The left sidebar has sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration Management (Inventory, Change tracking, State configuration (DSC)), Update management (Update management), and Process Automation (Runbooks). The 'Runbooks' link in the Process Automation section is also highlighted with a red box. The main content area displays a table of existing runbooks:

Name	Authoring status	Runbook type	Last modified
AzureAutomationTutorial	✓ Published	Graphical Runbook	6/8/2020, 7:28 PM
AzureAutomationTutorialPython	✓ Published	Python 2 Runbook	6/8/2020, 7:28 PM
AzureAutomationTutorialScript	✓ Published	PowerShell Runbook	6/8/2020, 7:28 PM

16. Supply a runbook name of IndexMaintenance and a runbook type of Powershell. You can supply a short description of your choosing. Click Create once finished.



## Create a runbook

X

Name \* ⓘ

IndexMaintenance



Runbook type \* ⓘ

PowerShell



Description

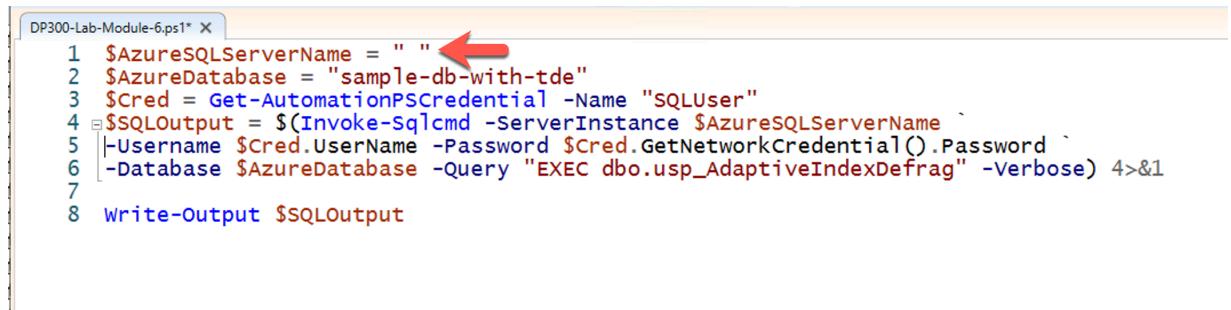
IndexMaintenance



Create

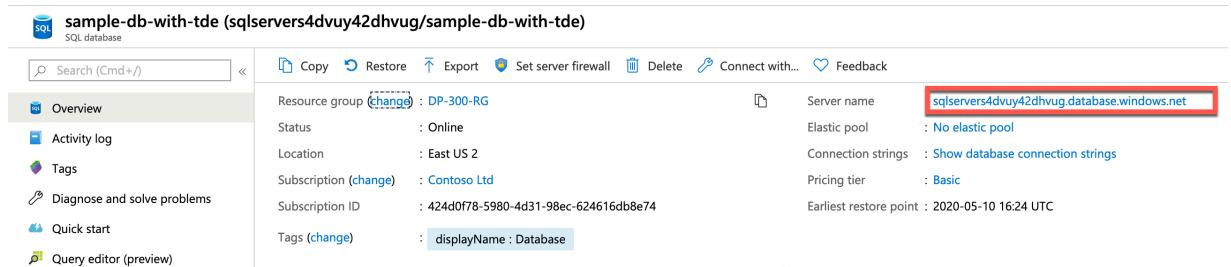
Once the runbook has been created, the process should drop you directly into the runbook.

17. Use Windows Explorer to navigate to the folder D:\LabFiles\Automate Resources. Right click on the DP300-Lab-Module-6.ps1 file, and click Edit. The PowerShell IDE will open.



```
DP300-Lab-Module-6.ps1* X
1 $AzureSQLServerName = " "
2 $AzureDatabase = "sample-db-with-tde"
3 $Cred = Get-AutomationPSCredential -Name "SQLUser"
4 $SQLOutput = $(Invoke-Sqlcmd -ServerInstance $AzureSQLServerName ` 
5 -Username $Cred.UserName -Password $Cred.GetNetworkCredential().Password ` 
6 -Database $AzureDatabase -Query "EXEC dbo.usp_AdaptiveIndexDefrag" -Verbose) 4>&1
7
8 Write-Output $SQLOutput
```

Edit the \$AzureSQLServerName variable to reflect the name of your Azure SQL server. You can get this value from the overview screen of your Azure SQL Database as shown below.



sample-db-with-tde (sqlservers4dvuy42dhvug/sample-db-with-tde)

Overview

Resource group (change) : DP-300-RG

Status : Online

Location : East US 2

Subscription (change) : Contoso Ltd

Subscription ID : 424d0f78-5980-4d31-98ec-624616db8e74

Tags (change) : displayName : Database

Server name : sqlservers4dvuy42dhvug.database.windows.net

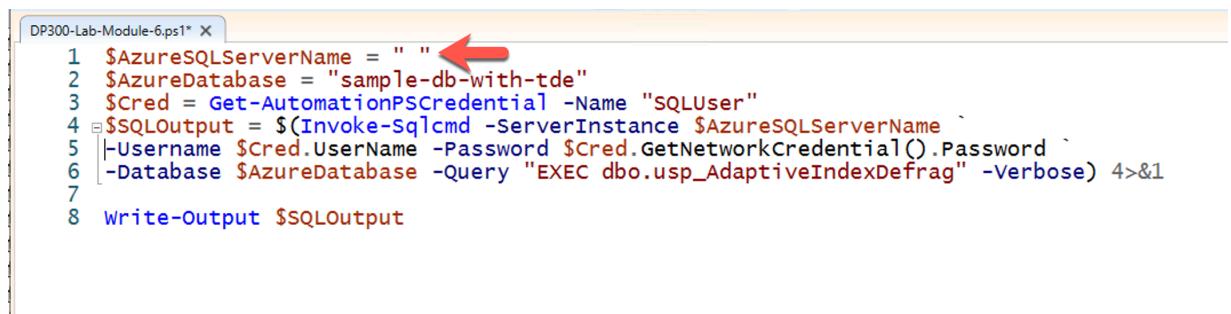
Elastic pool : No elastic pool

Connection strings : Show database connection strings

Pricing tier : Basic

Earliest restore point : 2020-05-10 16:24 UTC

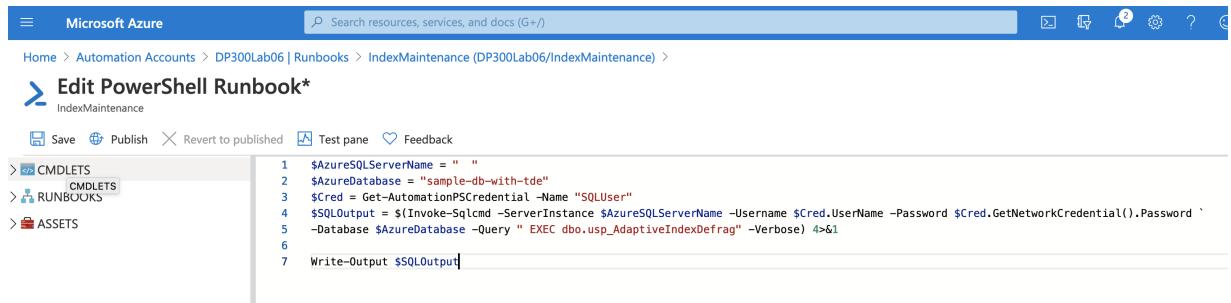
Copy the name of the server including the .database.windows.net domain suffix.



```
DP300-Lab-Module-6.ps1* X
1 $AzureSQLServerName = " "
2 $AzureDatabase = "sample-db-with-tde"
3 $Cred = Get-AutomationPSCredential -Name "SQLUser"
4 $SQLOutput = $(Invoke-Sqlcmd -ServerInstance $AzureSQLServerName ` 
5 -Username $Cred.UserName -Password $Cred.GetNetworkCredential().Password ` 
6 -Database $AzureDatabase -Query "EXEC dbo.usp_AdaptiveIndexDefrag" -Verbose) 4>&1
7
8 Write-Output $SQLOutput
```

On Line 1 of the file paste in your server name. Select all of the text and copy it to your clipboard.

18. Navigate back to your automation runbook, and paste in the PowerShell code you copied from the PowerShell IDE. Then click Save and then click Publish



Microsoft Azure

Home > Automation Accounts > DP300Lab06 | Runbooks > IndexMaintenance (DP300Lab06/IndexMaintenance) >

Edit PowerShell Runbook\*

IndexMaintenance

Save Publish Revert to published Test pane Feedback

```
1 $AzureSQLServerName = " "
2 $AzureDatabase = "sample-db-with-tde"
3 $Cred = Get-AutomationPSCredential -Name "SQLUser"
4 $SQLOutput = $(Invoke-Sqlcmd -ServerInstance $AzureSQLServerName -Username $Cred.UserName -Password $Cred.GetNetworkCredential().Password ` 
5 -Database $AzureDatabase -Query "EXEC dbo.usp_AdaptiveIndexDefrag" -Verbose) 4>&1
6
7 Write-Output $SQLOutput
```

19. Click Yes if prompted to over-write any previously published versions.

**Edit PowerShell Runbook**  
IndexMaintenance

Save Publish Revert to published Test pane Feedback

Publish Runbook  
This will publish this version of the runbook and override the previously published version. Do you want to proceed?

Yes No

> variables 8

20. Next you will schedule the runbook to execute on a regular basis. Click on Schedules in the left hand navigation menu. Then click on + Add a schedule at the top.

**DP300Lab06 | Schedules**

Automation Account

Search (Cmd+ /) << + Add a schedule Refresh

Name	Next run
No schedules found.	

Jobs

Runbooks gallery

Hybrid worker groups

Watcher tasks

Shared Resources

Schedules

Modules

Modules gallery

21. Click on Link a schedule to your runbook.
22. Select Create a new schedule.

> IndexMaintenance (DP300Lab06/IndexMaintenance) | Schedules > Schedule Runbook >

## Schedule

+ Create a new schedule

No schedules found.

23. Supply a descriptive schedule name and a description if desired.
24. Specify the start time of 4:00AM of the following day and in the Eastern Time zone. Configure the reoccurrence for every 1 Day. Do not set an expiration.

## New Schedule

Name \*

Dailyat0400



Description

Daily at 4AM EST



Starts \* ⓘ

05/09/2020



4:00 AM

Time zone

United States - Eastern Time



Recurrence

Once

Recurring

Recur every \*

1

Day



Set expiration

Yes

No

Expires

Never

**Create**

25. Click **Create**. Click **OK**.
26. The schedule is now linked to the runbook.

The screenshot shows the Microsoft Azure Runbook interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, a breadcrumb trail shows 'Home > DP300Lab06 | Runbooks >'. The main title is 'IndexMaintenance (DP300Lab06/IndexMaintenance) | Schedules'. On the left, there's a sidebar with 'Runbook' selected, and options for 'Overview', 'Activity log', and 'Traces'. The main area has a search bar, a 'Add a schedule' button, and a 'Refresh' button. A table lists a single scheduled run: 'Name' is 'DailyAt0400', and 'Next run' is '6/17/2020 4:00 AM (UTC Time)'.

43.2 lab: title: 'Lab 7 – Planning and Implementing a High Availability and Disaster Recovery Environment' module: 'Planning and Implementing a High Availability and Disaster Recovery Environment'

## 44 Lab 7 – Planning and Implementing a High Availability and Disaster Recovery Environment

**Estimated Time:** 90 minutes

**Pre-requisites:** The Azure SQL Database used by Exercise 1 was created in the lab for Module 3.

**Lab files:** The files for this lab are located in the D:\Labfiles\High Availability folder.

## 45 Lab overview

The students will execute two main tasks: make Azure SQL Database geo-redundant, and backup to and restore from a URL which uses Azure.

## 46 Lab objectives

After completing this lab, you will be able to:

- Enable geo-replication for Azure SQL Database
- Backup and restore a SQL Server database using a URL

## 47 Scenario

Now that you have automated day-to-day tasks in the previous lab, as the Senior Data Engineer, you are tasked with improving the availability of both IaaS and PaaS configurations for your database environment. You are tasked with the following objectives:

- Enable geo-replication for Azure SQL Database to increase availability for a database.
- Back up a database to a URL in Azure and restore it after a human error has occurred.

## 48 Exercise 1: Enable Geo-Replication for Azure SQL Database

**Estimated Time:** 45 minutes

### Overview

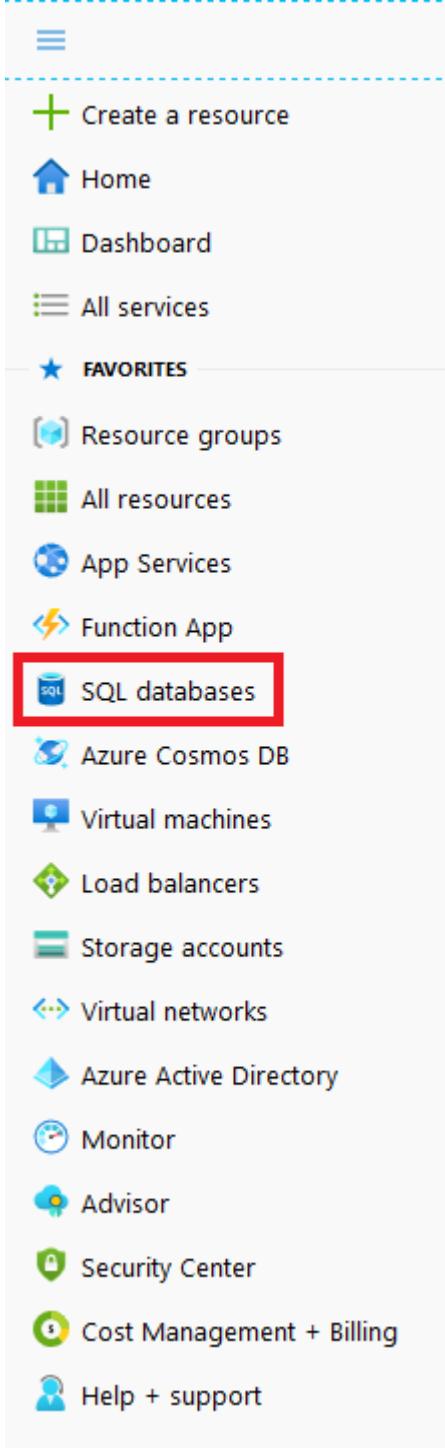
The Students will alter the configuration of the Azure SQL Database created during the lab for Module 3 to make it highly available.

### Scenario

As a DBA within WideWorldImporters you need to know how to enable geo-replication for Azure SQL Database, ensure it is working, and know how to manually fail it over to another region using the portal.

**Pre-requisites**

- Azure account created for the student – must provide a login (e-mail) and password
  - Azure SQL Database server and database pre-created
1. If you are not logged into the Azure portal via a browser window, do so using the Azure credentials provided to you.
  2. From the menu, select SQL databases as shown below.



3. Click on the Azure SQL Database that was created in Lab 3. An example is shown below.

The screenshot shows the 'SQL databases' blade in the Azure portal. At the top, there are buttons for '+ Add', 'Reservations', 'Edit columns', 'Refresh', 'Assign tags', and 'Delete'. Below this is a message: 'Try our new Azure SQL resource browser! This experience offers a unified view of all your SQL Server resources in Azure as well as improved sorting and filtering. Click here to go to the new experience.' A 'Subscriptions' section follows, with a note: 'Contoso Ltd – Don't see a subscription? Open Directory + Subscription settings'. The main area is a table with the following columns: Name, Status, Replication role, Server, Pricing tier, Location, and Subscription. The first row, 'AdventureWorks (dp300-lab... Online)', is highlighted with a red box.

4. In the blade for the database, under Data management, select Geo-Replication.

## Data management

- Geo-Replication**
- Sync to other databases

The region where the database is currently configured is shown in a blue hexagon with a white checkmark, as shown below. You will see that geo-replication is not configured.

The screenshot shows the 'AdventureWorks (dp300-lab-msl/AdventureWorks) | Geo-Replication' blade. On the left, a sidebar lists options: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, Query editor (preview), Power Platform, Power BI (preview), Power Apps (preview), Power Automate (preview), Settings, Configure, and Geo-Replication. The 'Geo-Replication' tab is selected and highlighted with a red box. The main area shows the 'Primary' database is 'dp300-lab-msl/AdventureWorks' and the 'Failover policy' is 'None'. The 'Secondaries' section notes 'Geo-Replication is not configured'. The 'Target regions' section lists the following regions, each with a green hexagon icon: West US 2, Central US, North Central US, Canada Central, East US, Canada East, and East US 2.

5. Select a Target Region. All available regions will have a hexagon with a green outline.

## Target regions

- West US 2
- Central US
- North Central US
- Canada Central
- East US
- Canada East
- East US 2

6. In this example, West US 2 was selected as can be seen on the Create secondary blade. Select Target server.

## Create secondary



Create geo-replicated secondaries to protect against prolonged datacenter outages. Secondaries have price implications. [Learn more](#)

Region



West US 2

Database name

AdventureWorks

\*Secondary type



Readable

\*Target server



[Configure required settings](#)

Elastic pool



None

\*Pricing tier



[Configure required settings](#)

7. Select Create a new server. On the New server blade, enter a name of module7geo (appended with your initials or a unique suffix), a valid admin login, and a secure password. Make a note of these credentials. Click Select when done.

## Server

Create a new server

No servers found

## New server

Server name \*

Enter server name .database.windows.net

Server admin login \*

Enter server admin login

Password \*

Confirm password \*

Location

(US) West US 2

Allow Azure services to access server (i)

Select

- Click **OK** on the Create secondary blade. The secondary server and the database will now be created. To check the status, look under the bell icon at the top of the portal. If successful, it will progress from Deployment in progress to Deployment succeeded.

# Notifications

X

[More events in the activity log →](#)

[Dismiss all](#)

■■■ Deployment in progress...

Running

Deployment to resource group 'DP300-Lab02' is in progress.

a few seconds ago

You will also notice that on the Geo-Replicate page, the status column of the Secondary will go from Initializing, to a seeding percentage complete, as shown below, to Readable when the replication is synchronized

The screenshot shows the Azure portal interface for managing a geo-replicated database. On the left, there's a sidebar with links like Home, Overview, Activity log, Tags, Diagnose and solve problems, Quick start, Query editor (preview), Power Platform, and Power BI (preview). The main area is titled "AdventureWorks (dp300-lab-msl/AdventureWorks) | Geo-Replication". A message bar at the top says "Select a region from the Target Regions list to create a secondary database." Below this, there's an info icon with the text "You can now automatically manage replication, connectivity and failover of this database by adding it to failover group." The main content area has a table with three columns: "Server/Database", "Failover policy", and "Status". The "Primary" row shows "East US" and "dp300-lab-msl/AdventureWorks" with "None" under Failover policy and "Online" under Status. The "Secondaries" row shows "West US 2" and "dp300-lab-msl-wus2/AdventureWorks" with "Seeding 0%" under Status, which is highlighted with a red box.

Server/Database	Failover policy	Status
Primary East US dp300-lab-msl/AdventureWorks	None	Online
Secondaries West US 2 dp300-lab-msl-wus2/AdventureWorks		Seeding 0%

9. Now that the Azure SQL Database is configured with geo-replication, you will perform a failover. Select your secondary server (West US 2 as shown in the previous picture but you may have chosen a different region).
10. On the blade, click Forced Failover.

**West US 2**

Secondary database

**Forced Failover**  Stop Replication

Region  
West US 2

Database name  
**AdventureWorks**

Server  
module7uniqueusername

Pricing tier  
General Purpose: Serverless, Gen5, 1 vCore

Status  
Readable

- When prompted, click Yes.

### Failover

Forced failover will switch module7uniqueusername/AdventureWorks to the primary role. The old primary dp300-lab06-xyz/AdventureWorks will automatically become the new secondary if it is online. All TDS sessions will be disconnected. This operation may cause data loss. Are you sure you want to proceed?

**Yes**

**No**

The status of the primary replica will switch to Pending and the secondary, Failover. The process will take a few minutes. When complete, the roles will switch with the secondary becoming the new primary and the old primary becoming the secondary.

**AdventureWorks (dp300-lab-msl/AdventureWorks) | Geo-Replication**

Select a region from the Target Regions list to create a secondary database.

	Server/Database	Failover policy	Status
<b>Primary</b>	West US 2 dp300-lab-msl-wus2/AdventureWorks	None	Online
<b>Secondaries</b>	East US dp300-lab-msl/AdventureWorks		Readable

## 49 Exercise 2: Backup to URL and Restore from URL

Estimated Time: 45 minutes

The tasks for this exercise are as follows:

- Configure backup to URL
- Back up WideWorldImporters
- Restore WideWorldImporters

## 49.1 Task 1: Configure Backup to URL

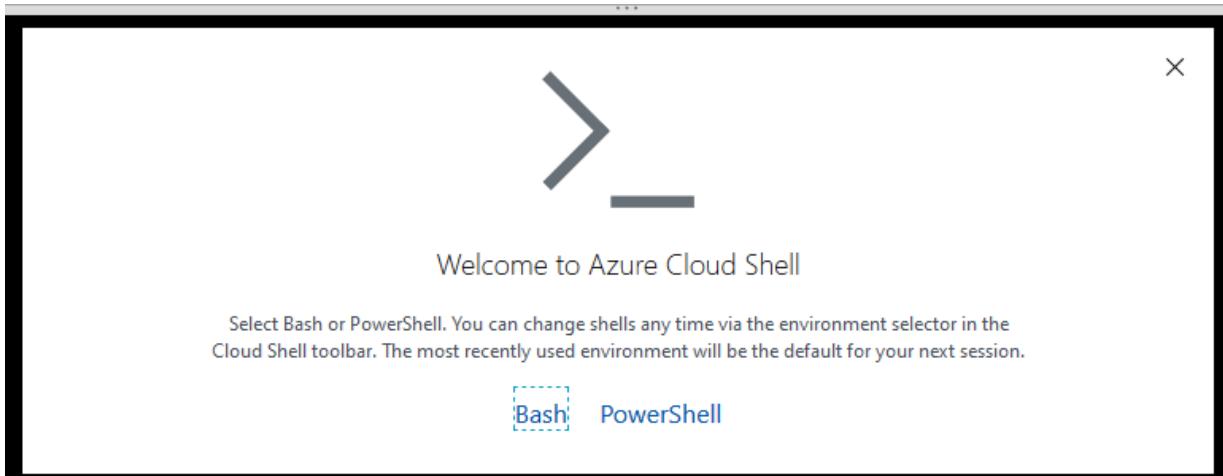
Before backing up a database in SQL Server to Azure, there are some configuration tasks to perform.

**Note:** There are several long strings, including storage account keys and shared access signatures, that are generated and then reused. You should consider opening up a Notepad file within the lab VM to use for holding these strings.

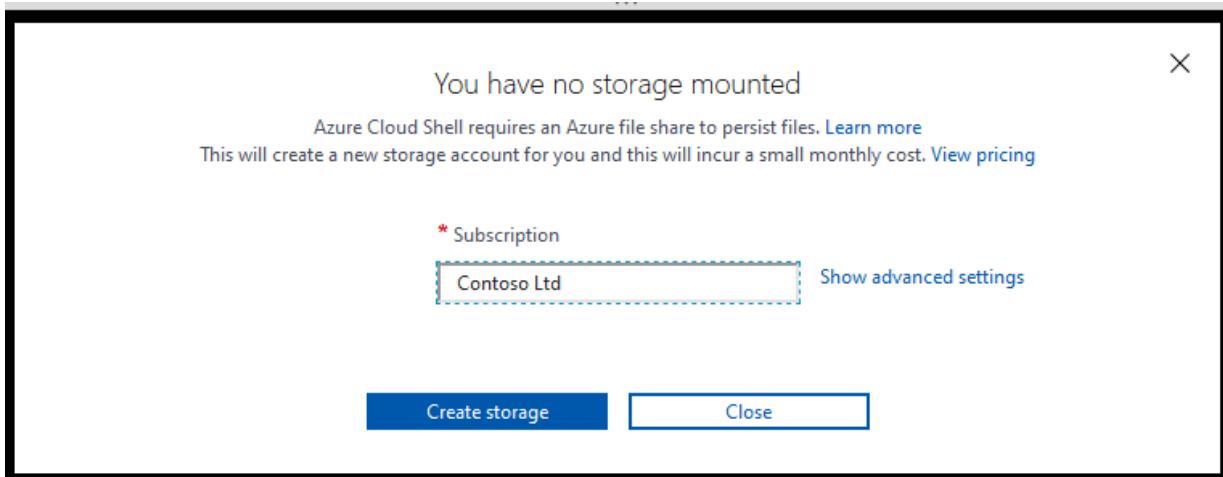
1. Verify that your context is the LON-SQL1 Virtual Machine.
2. Start Edge and log into the Azure portal (<https://portal.azure.com>) unless you are already there.
3. Open a Cloud Shell prompt by selecting the icon shown below in the upper right corner.



4. At the bottom half of the portal, you may see a message welcoming you to Azure Cloud Shell, if you have not yet used Cloud Shell. Select Bash.



5. If you have not previously used Cloud Shell, you must give it storage. Click Create Storage in the dialog below.



6. If you have already used Cloud Shell, just make sure the upper left corner of the Cloud shell screen shows Bash. You can use the drop down arrow to select either PowerShell or Bash.

Once complete, you will see a prompt similar to the one below.

```
Initializing your account for Cloud Shell...-
Requesting a Cloud Shell. Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

allan@Azure:~$ █
```

7. Create a storage account from the CLI using by copying the following command into cloud shell.

```
az storage account create -n dp300storage -g DP-300-Lab02 --kind StorageV2 -l eastus2
```

Edit the command so that your storage account name is unique and all lower case with no special characters. You should change dp300storage in the above to a unique name like dp300storagems123. The value DP-300-Lab02 is the name of an existing Resource Group. Make sure you use one that has been created in an earlier lab. Change the region if desired. Press Enter to run the command.

Next you will get the account keys for your account, which you will use in subsequent steps. Execute the following code in cloud shell, after editting to use the same name (after the -n) and resource group (after the -g) that you used in the previous command.

```
az storage account keys list -g DP-300-Lab02 -n dp300storage
```

Your account key will be in the results of the above command. Make sure you Copy the returned value for key1 (without the double quotes) as shown here. You might save it in Notepad.

```
:~$ az storage account keys list -g dp300lab06 -n dp300storagejd
[
  {
    "keyName": "key1",
    "permissions": "Full",
    "value": "jmnnHDN7A+liwtVBrB4OuUS896CCR8oMkPDBpDedRkKHAaYSauwBK2yVz1byYAZVKn5CW+6kDOMaUZIU3K9fXTQ=="
  },
  {
    "keyName": "key2",
    "permissions": "Full",
    "value": "W0y/UrDrBDlkN+48jhggKW5xGQmkQQtJOGVuT+g11tp3X5QP/mUqt2j1UOOAjiIdMuWwNfOIiGcT2+wVXLYZQ=="
  }
]
```

8. Backing up a database in SQL Server to a URL uses a storage account and a container within it. You will create a container specifically for backup storage in this step. To do this, execute the following command where dp300storage is the storage account name used when creating the storage account and storage\_key is the key generated above.

```
az storage container create --name "backups" --account-name "dp300storage" --account-key "storage_key"
```

The output should return true.

```
{
  "created": true
}
```

9. To further verify the container backups has been created, execute the following, where dp300storage is the storage account name used you created and storage\_key is the key you generated above.

```
az storage container list --account-name "dp300storage" --account-key "storage_key"
```

Part of the output should return something similar to below.

```
{
  "metadata": null,
  "name": "backups",
  "properties": {
    "etag": "\"0x8D7FACD95D34918\"",
    "hasImmutabilityPolicy": "false",
    "hasLegalHold": "false",
    "lastModified": "2020-05-18T01:48:38+00:00",
    "lease": {
      "duration": null,
      "state": null,
      "status": null
    },
    "leaseDuration": null,
    "leaseState": "available",
    "leaseStatus": "unlocked",
    "publicAccess": null
  }
},
```

10. A shared access signature (SAS) at the container level is required for security. This can be generated via Cloud Shell or PowerShell. Execute the following, where dp300storage is the storage account name you created above, storage\_key is the key generated above, and date\_in\_the\_future is a time later than now. date\_in\_the\_future must be in UTC. An example is 2020-10-31T00:00Z which translates to expiring at October 31, 2020 at midnight:

```
az storage container generate-sas -n "backups" --account-name "dp300storage" --account-key "storage
```

**Note:** The format for expiry is "YYYY-MM-DD", where YYYY is the four digit year, MM is the two digit Month, and DD is the two digit day.

The output should return something similar to the value shown below which will be used in the next Task. You can copy and save the value in Notepad along with the key you saved earlier.

```
se=2020-05-31T00%3A00Z&sp=rw&sv=2018-11-09&sr=c&sig=HoTQqk/AaBYWH02nPh4kBPK%2BZEIA809lQIRbqf2mjGY%3D
```

## 49.2 Task 2: Back Up WideWorldImporters

Now that the functionality is configured, you can generate a backup file as a blob in Azure.

1. Open SQL Server Management Studio and ensure you are connected to LON-SQL1.
2. Click New Query.
3. Create the credential that will be used to access storage in the cloud with the following Transact-SQL. (If a credential already exists, drop it first.) Fill in the appropriate values, where dp300storage is the storage account name created in Task 1, Step 8 and sas\_token is the value generated in Task 1, Step 10.

```
IF EXISTS
  (SELECT * FROM sys.credentials
  WHERE name = 'https://dp300storage.blob.core.windows.net/backups')
BEGIN
  DROP CREDENTIAL [https://dp300storage.blob.core.windows.net/backups]
END
GO
```

```
CREATE CREDENTIAL [https://dp300storage.blob.core.windows.net/backups]
```

```
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
```

```
SECRET = 'sas_token'
```

```
GO
```

4. Click Execute. This should be successful.

5. Back up the database WideWorldImporters to Azure with the following command in Transact-SQL:

```
BACKUP DATABASE WideWorldImporters
```

```
TO URL = 'https://dp300storage.blob.core.windows.net/backups/WideWorldImporters.bak';
```

```
GO
```

where dp300storage is the storage account name used in Task 1.

If successful, you should see output similar to this:

```
Processed 1240 pages for database 'WideWorldImporters', file 'WWI_Primary' on file 1.
```

```
Processed 53104 pages for database 'WideWorldImporters', file 'WWI_UserData' on file 1.
```

```
Processed 3865 pages for database 'WideWorldImporters', file 'WWI_InMemory_Data_1' on file 1.
```

```
Processed 1468 pages for database 'WideWorldImporters', file 'WWI_Log' on file 1.
```

```
BACKUP DATABASE successfully processed 59677 pages in 14.839 seconds (31.419 MB/sec).
```

```
Completion time: 2020-05-18T08:01:41.6935863+00:00
```

If something is configured wrong, you will see an error message similar to the following:

```
Msg 3201, Level 16, State 1, Line 33
```

```
Cannot open backup device 'https://dp300storage.blob.core.windows.net/container_name/WideWorldImporters.bak'.
```

```
Operating system error 50(The request is not supported.).
```

```
Msg 3013, Level 16, State 1, Line 33
```

```
BACKUP DATABASE is terminating abnormally.
```

Check that you did not mistype anything and that everything was created successfully.

6. To see that the file is actually in Azure, you can use Storage Explorer or Azure Cloud Shell. The syntax in Bash is as follows, where dp300storage is the storage account name used in Task 1, storage\_key is the key used there as well.

```
az storage blob list -c "backups" --account-name "dp300storage" --account-key "storage_key"
```

Sample output is shown below.

```
[  
  {  
    "content": null,  
    "deleted": false,  
    "metadata": null,  
    "name": "WideWorldImporters.bak",  
    "properties": {  
      "appendBlobCommittedBlockCount": null,  
      "blobTier": "Hot",  
      "blobTierChangeTime": null,  
      "blobTierInferred": true,  
      "blobType": "BlockBlob",  
      "contentLength": 493223936,  
      "contentRange": null,  
      "contentSettings": {  
        "cacheControl": null,  
        "contentDisposition": null,  
        "contentEncoding": null,  
        "contentLanguage": null,  
        "contentMd5": null,  
        "contentType": "application/octet-stream"  
      },  
      "copy": {  
        "completionTime": null,  
        "id": null,  
        "progress": null,  
        "source": null,  
        "status": null,  
        "statusDescription": null  
      },  
      "creationTime": "2020-05-18T08:01:26+00:00",  
      "deletedTime": null,  
      "etag": "0x8D7FB01B32ABD20",  
      "lastModified": "2020-05-18T08:01:41+00:00",  
      "lease": {  
        "duration": null,  
        "state": "available",  
        "status": "unlocked"  
      },  
      "pageBlobSequenceNumber": null,  
      "remainingRetentionDays": null,  
      "serverEncrypted": true  
    },  
    "snapshot": null  
  }  
]
```

### 49.3 Task 3: Restore WideWorldImporters

This task will show you how to restore a database.

1. In SQL Server Management Studio, in a New Query window, execute

```
USE WideWorldImporters;
GO
```

2. Now execute the statement below to return the very first row of the Customers table which has a CustomerID of 1. Note the name of the customer.

```
SELECT TOP 1 * FROM Sales.Customers;
GO
```

3. Run this command to change the name of that customer.

```
UPDATE Sales.Customers
SET CustomerName = 'This is a human error'
WHERE CustomerID = 1;
GO
```

4. Re-run Step 2 to verify that the name has been changed. Now imagine if someone had changed thousands or millions of rows without a WHERE clause – or the wrong WHERE clause.

5. To restore the database to get it back to where it was before the change you made in Step 3, execute the following, where dp300storage is the storage account name used in Task 1.

```
USE master;
GO
```

```
RESTORE DATABASE WideWorldImporters
FROM URL = 'https://dp300storage.blob.core.windows.net/backups/WideWorldImporters.bak';
GO
```

The output should be similar to this:

```
Processed 1240 pages for database 'WideWorldImporters', file 'WWI_Primary' on file 1.
Processed 53104 pages for database 'WideWorldImporters', file 'WWI_UserData' on file 1.
Processed 1468 pages for database 'WideWorldImporters', file 'WWI_Log' on file 1.
Processed 3865 pages for database 'WideWorldImporters', file 'WWI_InMemory_Data_1' on file 1.
RESTORE DATABASE successfully processed 59677 pages in 16.167 seconds (28.838 MB/sec).
Completion time: 2020-05-18T08:35:06.6344123+00:00
```

6. When the restore of WideWorldImporters is finished, re-run Steps 1 and 2. The data will be back to what it was.