WorkshopPLUS

Microsoft Azure Infrastructure as a Service

Working with Azure Storage

Student Lab Manual

V1.0, November 28, 2016

Information in this document is subject to change without notice. The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarked, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2014 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, MS, Windows, Windows NT, MSDN, Active Directory, BizTalk, SQL Server, SharePoint, Outlook, PowerPoint, FrontPage, Visual Basic, Visual C++, Visual J++, Visual InterDev, Visual SourceSafe, Visual C#, Visual J#,  and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

Other product and company names herein may be the trademarks of their respective owners.

Contents

[Working with Azure Storage 4](#_Toc510688390)

[Prerequisites 4](#_Toc510688391)

[Exercise 1 – Storage Account and Environment Setup 5](#_Toc510688392)

[Task 1 – Setup your Azure Resource Manager (ARM) template 5](#_Toc510688393)

[Task 2 – Deploy your Azure Resource Manager (ARM) template 7](#_Toc510688394)

[Task 3 – Confirm Deployment of your Azure Resource Manager (ARM) template 8](#_Toc510688395)

[Task 4 – Install the Microsoft Azure Storage Explorer 9](#_Toc510688396)

[Task 5 – Review metrics information being gathered 10](#_Toc510688397)

[Exercise 2 – Allow access to Azure Storage table data 12](#_Toc510688398)

[Task 1 – Create the SAS Key 12](#_Toc510688399)

[Task 2 – Using the SAS Key to query the table data 15](#_Toc510688400)

[Exercise 3 – Using Azure Files 17](#_Toc510688401)

[Task 1 – Create your file share and directory 17](#_Toc510688402)

[Task 2 – RDP in to your VM and connect to the file share 20](#_Toc510688403)

# Working with Azure Storage

In this lab, you will learn:

* How to create an Azure Storage account and VM using an ARM template. The VM will output diagnostics information to the storage account
* How to add Windows Azure Diagnostics to an ARM template
* How to use Microsoft Azure Storage Explorer
* How to setup a SAS key on a blob container to share access to another user
* How to implement Azure Files functionality

## Prerequisites

The following is required to complete this hands-on lab:

* Microsoft Visual Studio 2015 Professional or Enterprise Edition with Update 3
* Microsoft Azure SDK for .NET (VS 2015) – 2.9.6
* A Microsoft Azure subscription

# Exercise 1 – Storage Account and Environment Setup

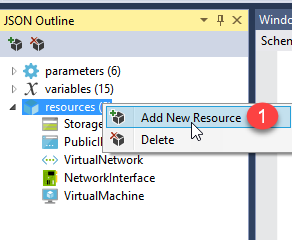
## Task 1 – Setup your Azure Resource Manager (ARM) template

In this lab exercise, you will be modifying an existing ARM template to add the Windows Azure Diagnostics feature. Windows Azure Diagnostics will be used to gather diagnostic information from the virtual machine and that information will be sent to an Azure Storage table.

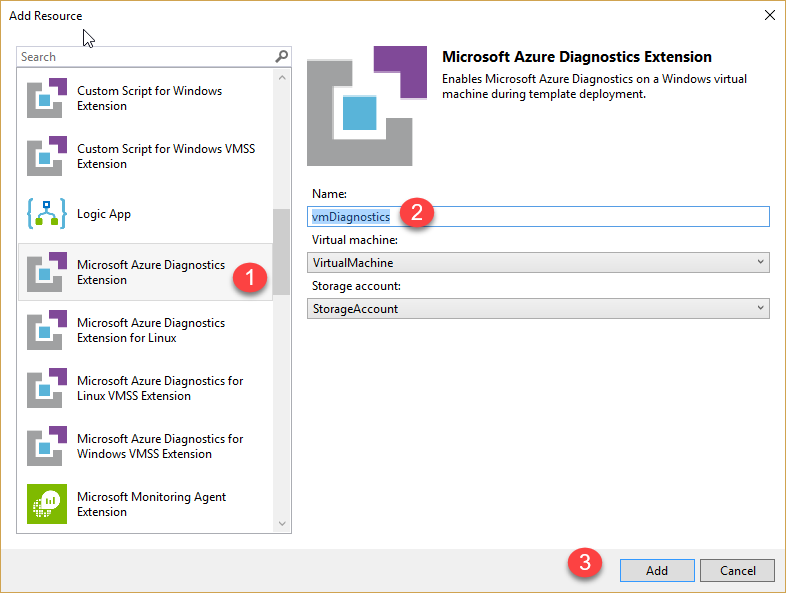
1. Open **Visual Studio** as an Administrator and select **File | Open | Project/Solution** and browse to **C:\AzureIaaSWS\M3 - Azure Storage\Labs\Exercise1\Begin\DeployStorage.**
2. Select the **DeployStorage.sln** file and click **Open**.
3. The pre-existing template has everything it needs for a deployment ‘without’ diagnostics; however, you will want to gather diagnostics from the virtual machine.

From within Solution Explorer, click on the **WindowsVirtualMachine.json** file which should open the JSON outline editor on the left-hand side of your Visual Studio window. If the JSON Outline editor does not appear, select the **View | Other Windows | JSON Outline** menu.

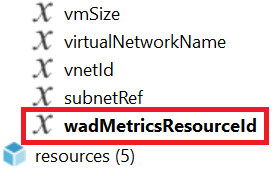
1. Right-click on the *resources* treeview item and select **Add New Resource**.



1. Scroll down in the *Add Resource* window and:
   1. Select the **Microsoft Azure Diagnostics Extension**.
   2. Enter a name (the name is up to you) in the **Name** field and leave the other settings as they are. Since you only have one VM being created and one storage account, the diagnostics will be placed in the single storage account.
   3. Click the **Add** button.



1. If you now look in the JSON Outline window in the **variables** section, you will see 1 new variable that is added for you. This variable is used to handle communication with the diagnostics framework. You do not need to fill in a value for this variable.



## Task 2 – Deploy your Azure Resource Manager (ARM) template

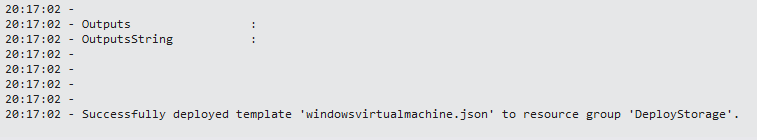
1. In the **Solution Explorer**, right-click on the **DeployStorage** project and select **Deploy | New…**
2. Sign-in to your Azure subscription, if you are not already signed in. For the *Resource group* drop-down, select **<Create New…>**.
   1. Create a resource group name that is unique within your subscription.
   2. Select the region you want your resource group to be in.
   3. Click the **Create** button.
3. Click on the **Edit Parameters**... button. Fill in the required parameters:

|  |  |
| --- | --- |
| 1. **adminUser** – enter a RDP login user name for the VM 2. **adminPassword** – enter a RDP login password for the VM 3. **dnsNameForPublicIP** – enter a dns name for the Public IP   Although it is optional to provide a DNS name for your VM, in this lab you will need to provide a dns name unique in all of Azure.   1. **storageAcctName** – enter a name for the storage account   It must be unique in all of Azure and provide the name of a NEW storage account.   1. **publicIPAddressName** – enter a name for the public IP address.   This name will be used for the reserved public IP address used by the VM.   1. **windowsOSVersion** – leave the setting as is |  |

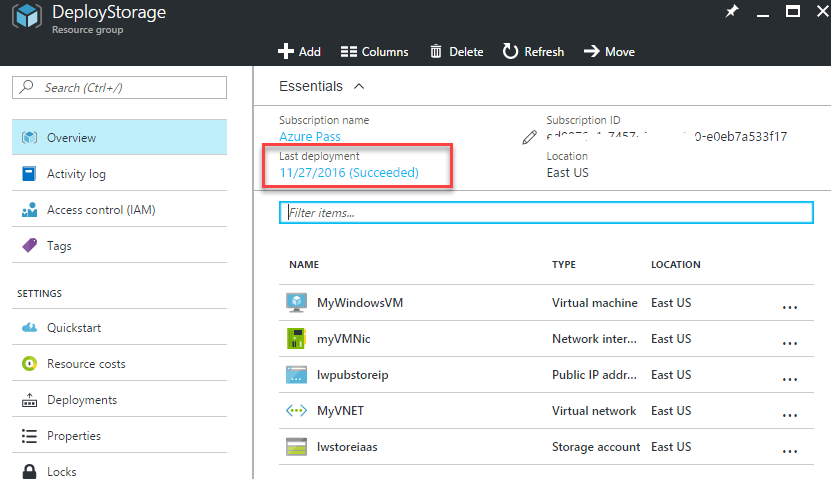
1. Click the **Deploy** button on the *Deploy to Resource Group* dialog box.

## Task 3 – Confirm Deployment of your Azure Resource Manager (ARM) template

1. If the Visual Studio deployment succeeded, you will see something similar to the following in the Visual Studio output window; otherwise, correct any errors that appear within the Visual Studio output window and re-deploy.



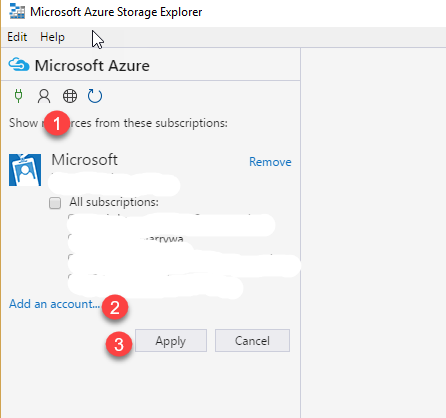
1. Log in to the Azure portal at <https://portal.azure.com>. Look for your resource group name to confirm that the resources have been deployed successfully.



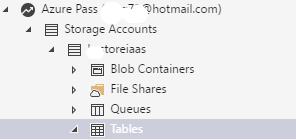
## Task 4 – Install the Microsoft Azure Storage Explorer

There are many ways to analyze data that is sent to Azure storage. You can use Visual Studio, the Azure Portal, the Azure command line interface or the REST API. For this lab, you will use a free tool named Microsoft Azure Storage Explorer.

1. Use your web browser to browse to <http://storageexplorer.com>.
2. Select the **Free download for Windows** button and download the install utility to your hard drive.
3. Install the Storage Explorer utility and select the option to have the Storage Explorer start up.
4. When the *Storage Explorer* window opens, enter your credentials to log in to your Azure account. If not prompted to log in:
   1. Click on the icon that looks like a *person* and then select the **Add an account.**.. link.
   2. Enter your Azure login information
   3. Select **Apply**. If prompted by the Storage Explorer, select ‘*All subscriptions: <name of your subscription>*’.



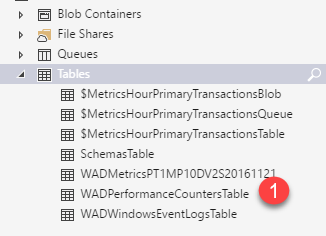
1. Once the *Storage Explorer* has logged in to Azure, you should be able to see the new storage account you created with the ARM template.



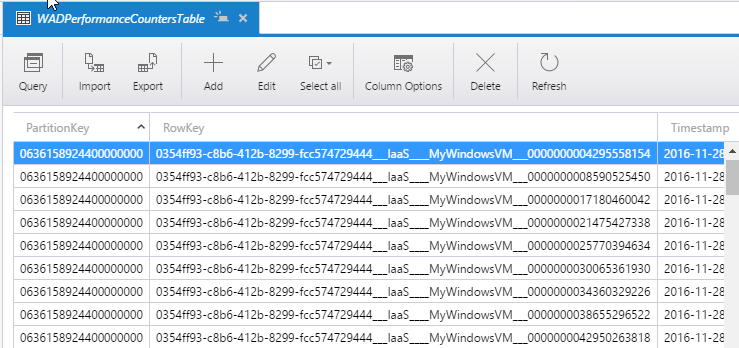
## Task 5 – Review metrics information being gathered

In a previous task, you added Windows Azure Diagnostics to your ARM template. The diagnostics settings have an abundance of performance counter configuration settings and this performance counter data will be gathered and pushed out to an Azure storage table named **WADPerformanceCountersTable**. You will view this table data.

1. From within the *Storage Explorer*, expand the **Tables** folder and then click on **WADPerformanceCountersTable**.



1. Within the table output window, you should be able to see the current metrics that have been gathered. From within this window, you can do sorting, searching, exporting of the data and many more activities.



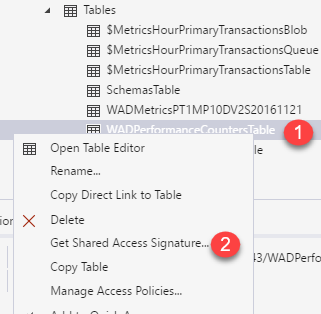
# Exercise 2 – Allow access to Azure Storage table data

In this lab exercise, you will create a Shared Access Signature (SAS) key that will be used to allow another person access to the table storage data that has been generated in the previous exercise. This type of SAS key can be beneficial when needing to allow temporary access to your data without giving full access to the rest of your storage account.

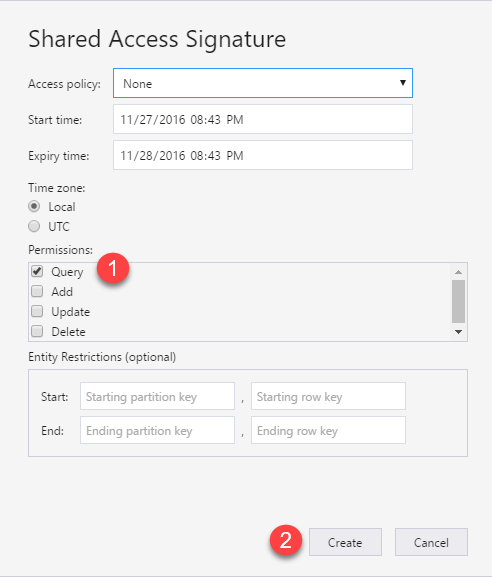
## Task 1 – Create the SAS Key

Your QA team would like access to the performance counter data you have gathered; therefore, you want to give them access to ONLY the WADPerformanceCountersTable.

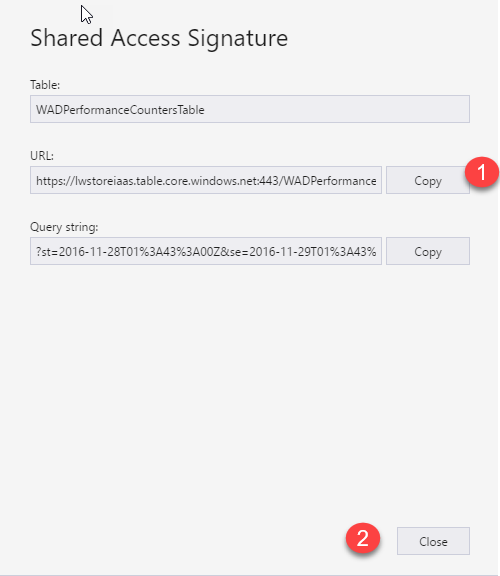
1. From within the *Storage Explorer*:
   1. Right-click on the **WADPerformanceCountersTable** table.
   2. Select the **Get Shared Access Signature…** menu item.



1. In the *Shared Access Signature* dialog box, you can set a valid time limit that the key will be valid for, but in this case, leave the default value you see in your own windows as it is.
   1. Make sure though that you only allow **Query** access.
   2. Select the **Create** button.



1. The SAS key will be generated.
   1. Click the **Copy** button for the URL to copy the key to your clipboard. Paste it in to Notepad (or some location where you retrieve the key from).
   2. Click **Close**.



1. Your SAS url should look something similar to this:

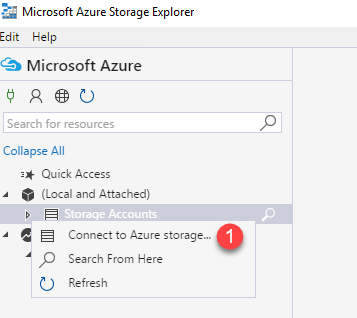
https://lwstoreiaas.table.core.windows.net:443/WADPerformanceCountersTable?st=2016-11-28T01%3A43%3A00Z&se=2016-11-29T01%3A43%3A00Z&sp=r&sv=2015-12-11&tn=wadperformancecounterstable&sig=ZwFpp0h.....

For ease of reading, the QA team should also be using the Microsoft Azure Storage Explorer. In the next task, you will learn how to use the SAS key to retrieve the performance counter data.

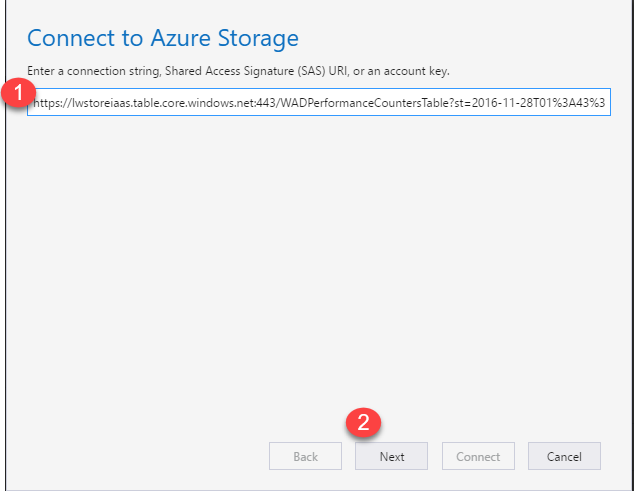
## Task 2 – Using the SAS Key to query the table data

In the previous task, you created a SAS key and copied it to the clipboard.

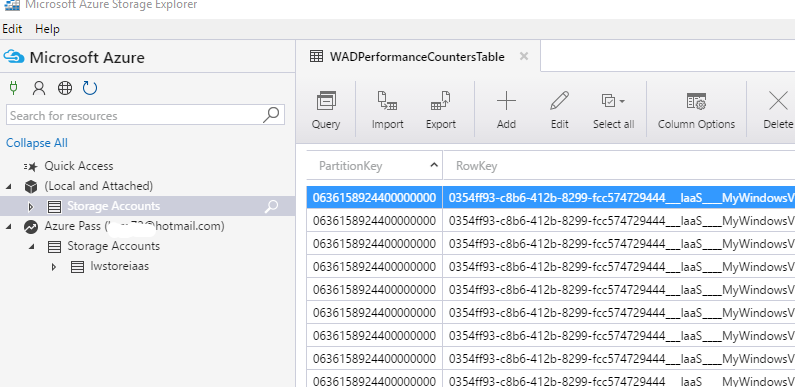
1. If necessary, open **Microsoft Azure Storage Explorer**. You do not have to sign in to any Azure account to use the SAS key.
2. Expand **Local and Attached** and right-click on **Storage Accounts**.
   1. Select **Connect to Azure storage**.



1. Select **Use a connection string or a shared access signature URI** and click **Next**.
2. Select **Use a SAS URI**.
   1. Paste the SAS Key in to the URI field
   2. Select the **Next** button.



1. Select the **Connect** button. What the user should see is the performance counter output data.

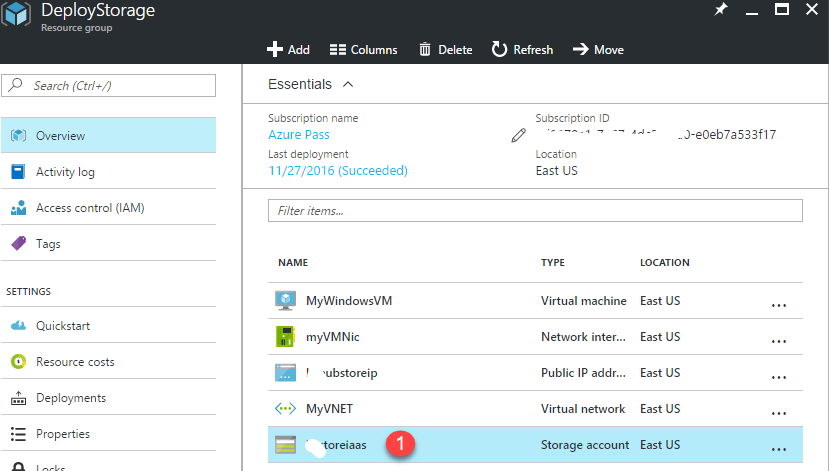


# Exercise 3 – Using Azure Files

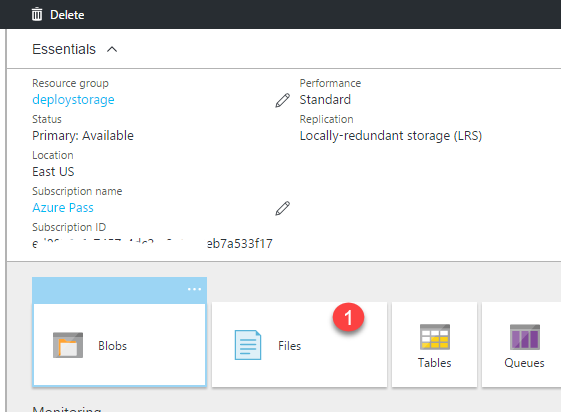
In this lab exercise, you will be creating a file share and directory using Azure Files and then you will RDP in to your VM to connect to the file share.

## Task 1 – Create your file share and directory

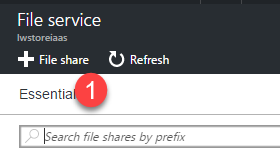
1. Log in to the Azure portal at https://portal.azure.com.
2. Find and browse to your resource group.
   1. Within the resource group, click on the storage account.



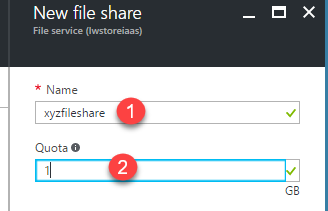
1. In the *Storage account* blade, select the **Files** tile.



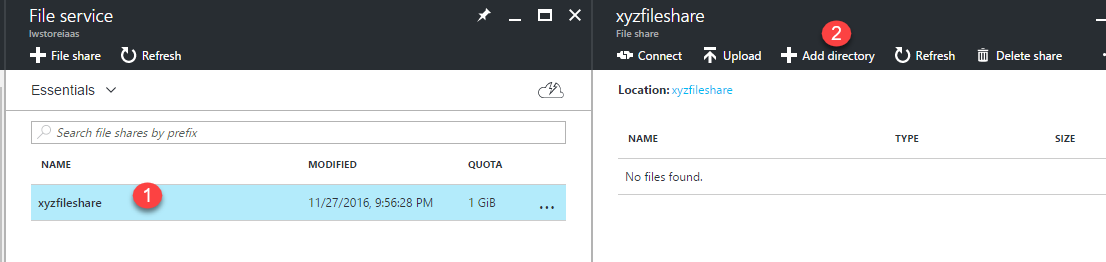
1. Click on the **File share** toolbar icon.



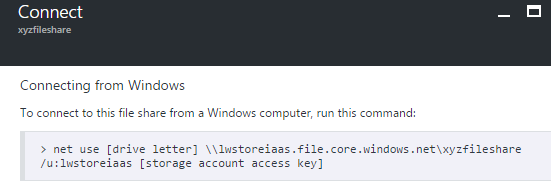
1. Enter the required information for the new file share and then click **OK**:
   1. Enter a name for your file share.
   2. Set a quota of 1 GB.



1. In the *File service* blade:
   1. Click on the file share name.
   2. Select the **Add directory** toolbar icon.



1. click**OK**
2. Prior to RDPing in to the VM, you can retrieve the connection command you will use inside of the VM using ‘net use’. In the *File share* blade, click the **Connect** menu item.
3. In the *Connect* blade, copy the net use connect command. If prompted, click **Allow access**. Paste the net use command in to your clipboard or Notepad.



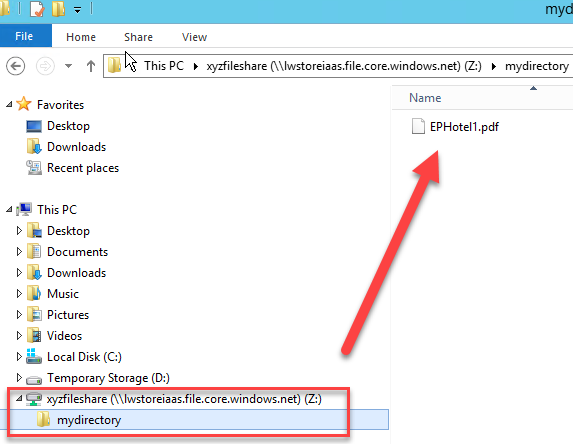
## Task 2 – RDP in to your VM and connect to the file share

1. From within your resource group, click on the virtual machine icon. When the *Virtual machine* blade opens, select the **Connect** toolbar icon at the top of the blade.
2. Click **Connect**, then enter your RDP log in credentials and click **OK**. If prompted, click **Yes** to proceed.
3. Paste the net use command that you copied for the VM in to the command prompt window and then press Enter. You should get a confirmation that the command completed successfully.

Although you can choose to map any drive letter you wish, as an example, the command below maps drive Z: to the share you created in the previous task (storage key has been truncated to make the command easier to read).

**net use Z: \\lwstoreiaas.file.core.windows.net\xyzfileshare /u:lwstoreiaas PHNcglGcc+Dyni+mQFS22LRYmoi…**

1. From within your VM, open up **Windows Explorer** and browse to the Z: drive. You should be able to browse to your directory and if you have already placed a file in to the directory, you should be able to see and manipulate it. You can drop files in to the directory from within the VM, programmatically or from within the Azure portal.



This is the end of this lab. Do not delete these resources, as they may be used in later labs. However, you should ensure that any existing VMs are in a stopped (deallocated) state, to conserve costs.