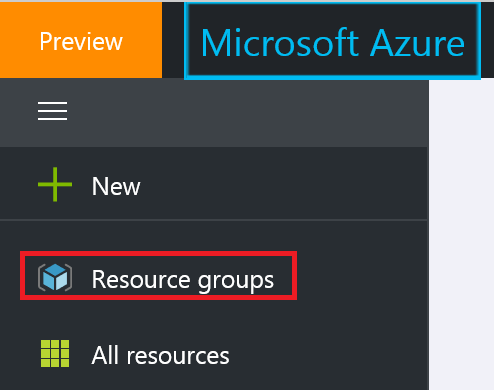
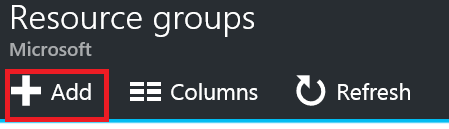
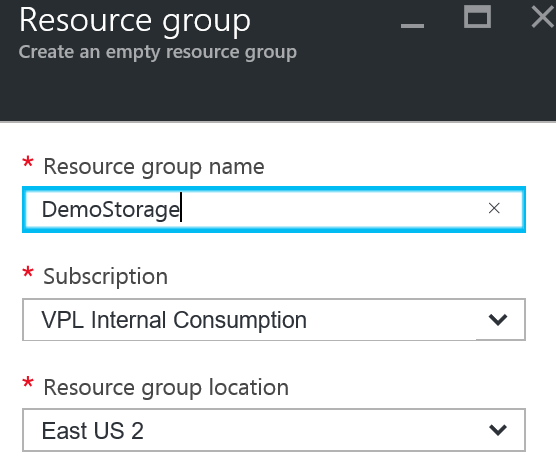
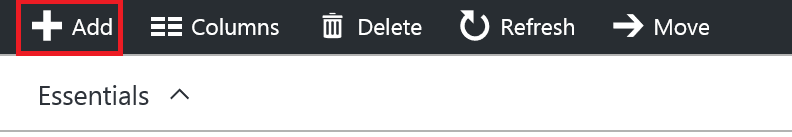
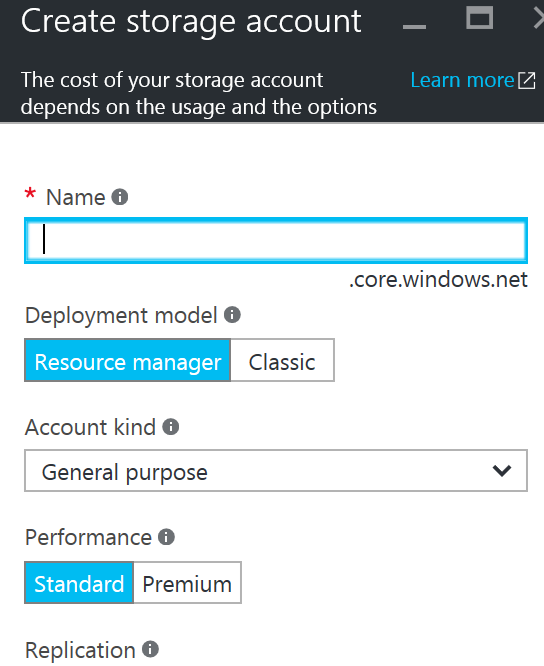
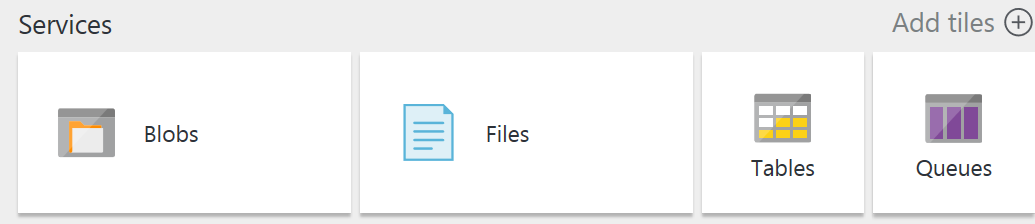
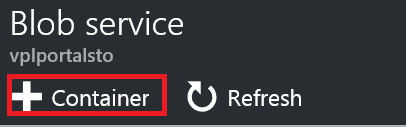
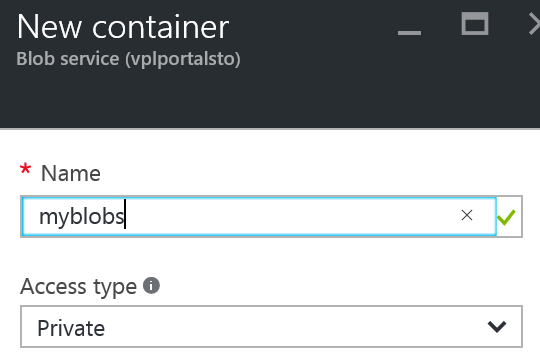
# Lab

This is the *Storage* lab.

# Lab objectives

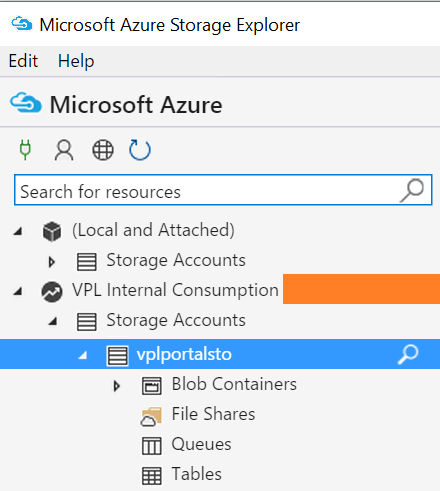
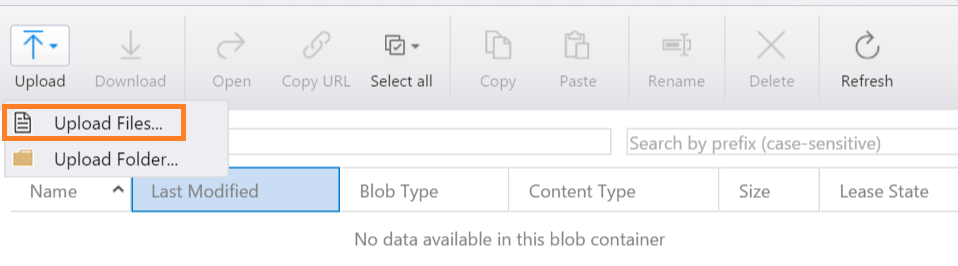
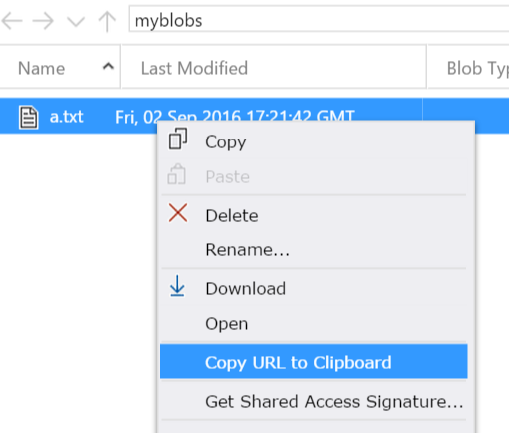
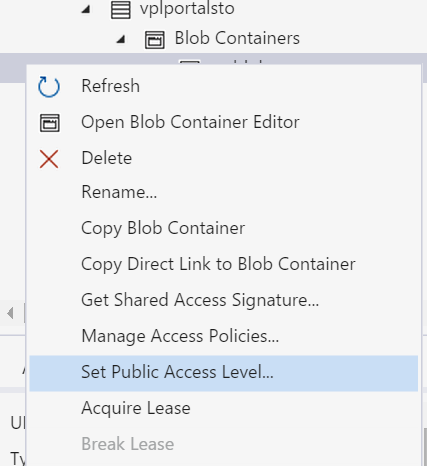
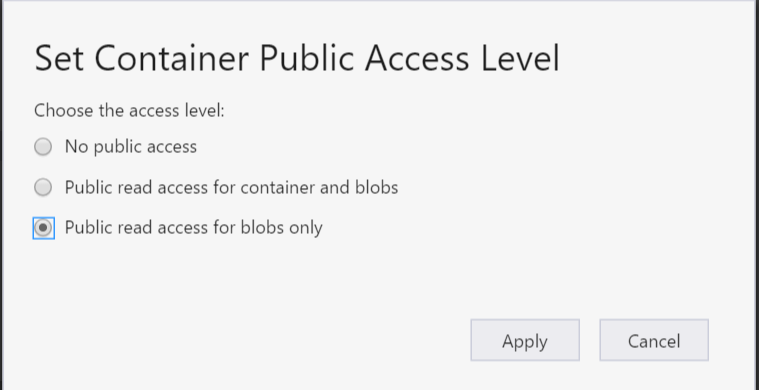
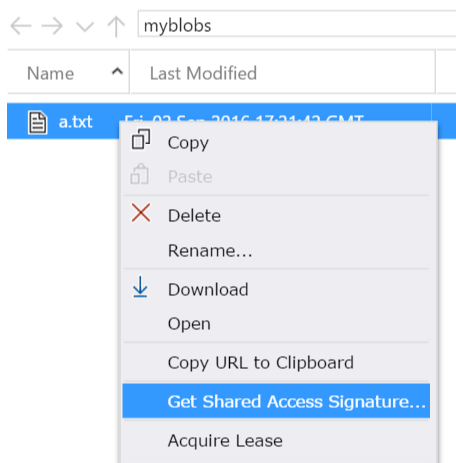
Create a storage account in three ways: portal, PowerShell, ARM template.

# Portal

1. Go to the portal, <https://portal.azure.com/>
2. First we’ll create a resource group to contain the storage accounts we’re going to create
   1. Navigate to your resource groups  
      
   2. Select *Add*  
      
   3. Fill the fields: *DemoStorage* for the name & *East US 2* for the location  
      
   4. Press *Create*
3. If you do not see the resource group you just created in the resource group list, click *Refresh* until you do
4. Select *DemoStorage* resource group
5. Select *Add*  
   
6. You’ll be brought to a search box on the Azure Marketplace
7. Type *Storage* and enter
8. In the search result, select *Storage Account*, published by Microsoft (it should be the first or first two)
9. Select *Create*
10. You should land on the following form  
    
11. For name, you have to enter something unique across all the storage accounts in Azure (not only yours), as this is used as a domain name ; in here, I’ll use vplportalsto ; you might want to use your initials (maybe with numbers) instead of *vpl* at the beginning
12. If the name you’ve chosen isn’t used by anyone else, you should get a green mark when you tab-out of the text box  
    
13. For deployment model, leave it to *Resource manager* ; this is ARM, the *Classic* deployment model is legacy
14. For Account kind, leave it to *standard* ; the alternative is *Blob storage*, which allows you to select between *hot* & *cool* access tier
15. For Performance, leave it to *standard* ; the alternative is *Premium*, SSD-backed, but currently only supports page blobs (i.e. no table, queue or files)
16. For *Replication*, select *Locally-redundant storage*, LRS
17. For the resource group, leave it to *Use existing* which should default to the resource group we just created, i.e. *DemoStorage*
18. Click *Create*
19. Go back to *DemoStorage* resource group and hit refresh until the storage account appears
20. Select the storage account
21. You should see the four different services under storage account  
    
22. Select *Blobs*
23. Select *add a container*  
    
24. Fill the form  
    
    1. For container name, type *myblobs*
    2. For access type, leave it to *private*, the most secure access type

# Access Type / SAS – Portal

Let’s explore the different ways to access a blob

1. If you do not have it installed, install *Storage Explorer* (<http://storageexplorer.com/>)
2. Open Storage Explorer, add an account corresponding to the account you are using for accessing the Azure Portal
3. You should see the storage account you just created under *Storage Accounts*  
   
4. Open the *Blob Containers* node
5. You should see *myblobs* underneath
6. Select the container
7. Create a text file (e.g. using notepad) locally (e.g. on your desktop) containing the text “AAA” with the name a.txt
8. Upload the file to the container  
     
   (You can also drag & drop the files to the container)
9. Right-click on a.txt and select *Copy URL to Clipboard*  
   
10. Paste the url in a browser
11. You should obtain an HTTP 404 error with the following payload
12. This is because the blob container is private and blobs can’t be accessed
13. Right click on the container and select *Set Public Access Level*  
    
14. Let’s select the last one, i.e. *Public read access for blobs only*  
    (This could also have been done in the portal)
15. Do a refresh on the browser page pointing to the a.txt blob
16. This time, it should give an HTTP 200 with “AAA” as payload
17. Let’s put the blob container private again and test that the blob returns an HTTP 404 error again
18. Right-click the blob and select *Get Shared Access Signature*  
    
19. Leave the default as they are ; for permissions, make sure “Read” is selected
20. Click *Create*
21. Next to the URL text box, click *Copy*  
    The query string of that URL is a SAS token
22. Paste that url in a browser
23. This should return an HTTP 200 with payload “AAA” ; the SAS token is valid only for the duration you specified

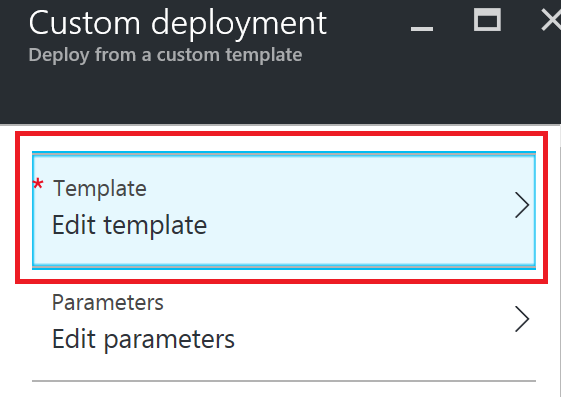
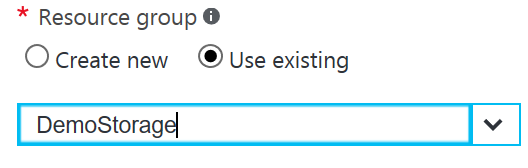
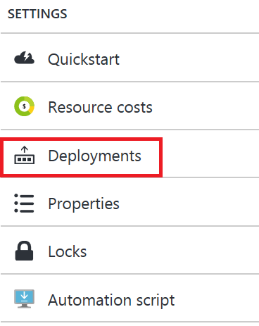
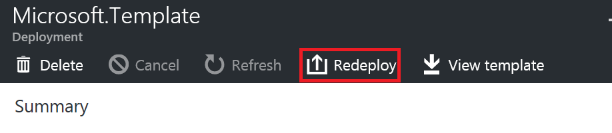
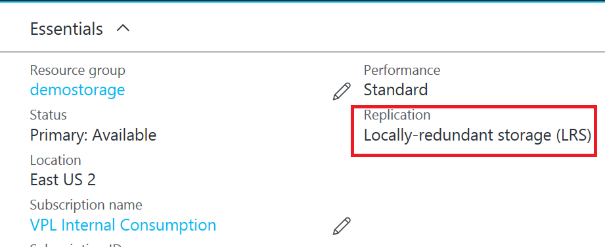
# PowerShell

We’ll create another storage account, in the same resource group, using PowerShell this time.

1. In ISE, assuming you’ve logged in (Add-AzureRmAccount), type  
   New-AzureRmStorageAccount -ResourceGroupName DemoStorage  
   -Name vplpssto -Type Standard\_LRS -Location EastUS2  
   vplpssto here is the name of storage account and again must be globally unique
2. That created a new storage account
3. Type $context = (Get-AzureRmStorageAccount -ResourceGroupName DemoStorage -Name vplpssto).Context
4. That retrieve the context of the storage account, used to manipulate it
5. Type New-AzureStorageContainer -Context $context -Name myblobs -Permission Off
6. That created the *myblobs* container
7. In the portal, go to the *DemoStorage* resource group and refresh, you should see the new storage account
8. Explore it, it should contain a new private blob container (with no blob in it)

# ARM Template

We’ll create another storage account, in the same resource group, this time using ARM Template.

1. In the Azure Portal, go to the top left corner and click the plus (+) sign
2. In the search text box, type “template” & hit enter
3. *Template Deployment*, published by Microsoft should be the first item ; select it
4. Hit *Create*
5. Select *Edit Template*  
   
6. Copy the content of the file *Storage ARM - Template.txt* (in the same folder)
7. Note the following
   1. The template has one parameter, the storage account name
   2. The parameter has a default value *yourprefixarmsto*
   3. There are no variables defined
   4. It has one resource
   5. The location of the resource is derived from the location of the resource group, using a template function in brackets *[]*
   6. A tag is inserted just for designer reason
   7. There are no extra properties
   8. There are no children properties
   9. There are no resource that resource depends upon
8. Click the *Save* button at the bottom of the pane
9. Click *Edit Parameters*
10. Remove “yourprefix” at the beginning of the value and replace it by your initials or anything unique
11. Click OK
12. In the resource group, select *Use existing*  
    
13. Select the *DemoStorage* resource group as the target resource group
14. Review the legal terms & click purchase  
    This doesn’t engage you into purchasing anything specific beyond the storage account ; this is a generic screen mostly used in Azure Marketplace where ARM templates are the deployment mechanism
15. Click *Create* button
16. Return to the *DemoStorage* resource group and refresh until you see your storage account
17. Select your new storage account
18. Notice the following
    1. The replication type is GRS, as specified in the template
    2. There are no blob container in the storage account ; ARM template do not allow (yet) to create containers explicitly (they do allow implicit container creation when files are copied in it)
19. Go back to the resource group pane
20. Select *deployments*  
    
21. This will list the last couple of deployments occurring in the resource group
22. The last one should be at the top titled “Microsoft.Template” ; select it
23. In the deployment pane, select *Redeploy*  
    
24. Select *Edit Template*
25. You will see the template we used to deploy
26. Find the sku of the resource and replace *Standard\_RAGRS* by *Standard\_LRS*  
    
27. Hit the save button at the bottom of the pane
28. Click *Edit Parameters*
29. Enter the same storage account name you’ve entered the first time
30. Redo steps 12-15
31. Go back to the resource group *DemoStorage*
32. Wait for the deployment to be over and select the storage account
33. You should now see it as LRS  
    
34. Note the following:
    1. The storage account wasn’t deleted / recreated ; an update was done on the resource based on a delta between the existing and the new template
    2. The other storage accounts we created in previous section were untouched: by default ARM only adds / updates resources

# Notes

Read <https://azure.microsoft.com/en-us/documentation/articles/storage-powershell-guide-full/> for more details on PowerShell commands for storage.

Read <https://trevorsullivan.net/microsoft-azure/microsoft-azure-resource-manager/> & <https://azure.microsoft.com/en-us/documentation/articles/resource-manager-deployment-model/> for more details on ARM deployment model.

# Exercise

Try to modify the ARM template to take two string parameters and use the concatenation of both parameters for the name of the storage account.

Hint: you will need an ARM template function, see <https://azure.microsoft.com/en-us/documentation/articles/resource-group-template-functions>.

Create Premium storage account & verify that you only have access to blob storage (no Files, Queue & Tables).

# Clean up

We won’t be using the resource group we have created so we can delete it

1. In ISE, type Remove-AzureRmResourceGroup -Name DemoStorage
2. You will be prompted by a dialog box to confirm, click *Yes*
3. It takes a little while to delete the three storage accounts
4. If you refresh the resource group list in the portal, you should see the resource group has disappeared