**Configure Azure Files on Active Directory Domain Services**

**Prerequisites**

1. An account that synced with Azure that has both domain admin in Active Directory and Global Admin in Azure. Create a temporary one if you have to.
2. Download and install AzFilesHybrid module on a machine from within the environment with internet access

https://github.com/Azure-Samples/azure-files-samples/releases

1. **Create Storage account**  
   1. In portal go to storage, create storage account

2. Enable azure files and create shared folder

3. Make sure to grab the share properties and storage account key

1. **From a machine within the environment with internet access**
2. In Powershell ISE import az and azfileshybrid modules
3. Navigate to where AzFilesHybrid is unzipped and stored and run to copy the files into your path

Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUse

.\CopyToPSPath.ps1

1. Connect to azure
2. Run the following commands

Join-AzStorageAccountForAuth

Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser

#Define parameters

$SubscriptionId = "<your-subscription-id-here>"

$ResourceGroupName = "<resource-group-name-here>"

$StorageAccountName = "<storage-account-name-here>"

#Select the target subscription for the current session

Select-AzSubscription -SubscriptionId $SubscriptionId

#Select a target OU. In active directory select an ou you want to target and get the DN

Join-AzStorageAccountForAuth `

-ResourceGroupName $ResourceGroupName `

-StorageAccountName $StorageAccountName `

-DomainAccountType "<ComputerAccount|ServiceLogonAccount>" <# Default is set as ComputerAccount #> `

-OrganizationalUnitDistinguishedName "<ou-distinguishedname-here>" <# If you don't provide the OU name as an input parameter, the AD identity that represents the storage account is created under the root directory. #> `

-EncryptionType "AES256” #Default is RC4 or AES256

Update-AzStorageAccountAuthForAES256 -ResourceGroupName $ResourceGroupName -StorageAccountName $StorageAccountName

1. Your storage account and active directory settings should be set. Test configuration by running

Debug-AzStorageAccountAuth -StorageAccountName $StorageAccountName -ResourceGroupName $ResourceGroupName -Verbose

$storageaccount = Get-AzStorageAccount `

-ResourceGroupName "<your-resource-group-name-here>" `

-Name "<your-storage-account-name-here>"

$storageAccount.AzureFilesIdentityBasedAuth.DirectoryServiceOptions

$storageAccount.AzureFilesIdentityBasedAuth.ActiveDirectoryProperties

1. Assign Share Level Permissions
2. In portal go to the storage account, go to Access Control (IAM)
3. Assign the rds user group Storage File Data SMB Share Contributor
4. Assign the admin account Storage File Data SMB Share Elevated Contributor
5. Assign AD NTFS permissions
6. Mount the drive with the storage account name and the key. Create a folder and assign it windows permissions. The following script will mount the drive

$connectTestResult = Test-NetConnection -ComputerName <storage-account-name>.file.core.windows.net -Port 445

if ($connectTestResult.TcpTestSucceeded)

{

net use <desired-drive-letter>: \\<storage-account-name>.file.core.windows.net\<share-name> /user:Azure\<storage-account-name> <storage-account-key>

}

else

{

Write-Error -Message "Unable to reach the Azure storage account via port 445. Check to make sure your organization or ISP is not blocking port 445, or use Azure P2S VPN, Azure S2S VPN, or Express Route to tunnel SMB traffic over a different port."

}

Or In file explorer you can right click your computer and map network drive and click user alternate credentials

\\%storageaccount%.file.core.windows.net\share

User = azure\%storageaccount

Password = storage account key

1. Create a folder, right click go to properties. Assign the proper ACLs.
2. Next log into a domain joined computer and test the mount. You can test

\\servername(storage computer account created earlier)\share\folder

Or

\\<storage-account-name>.file.core.windows.net\<share-name>\folder