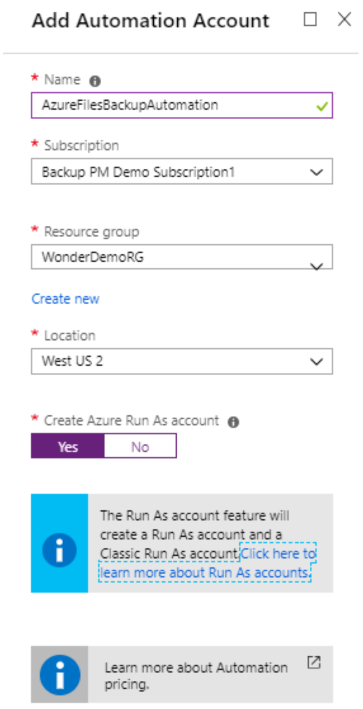
Automate On-demand backup for Azure File Shares using PowerShell for Azure Backup

Steps to automate using Azure Automation

Step1: Create an Automation resource with "Run As" account



Add Automation Account

* Name
AzureFilesBackupAutomation

* Subscription
Backup PM Demo Subscription1

* Resource group
WonderDemoRG

Create new

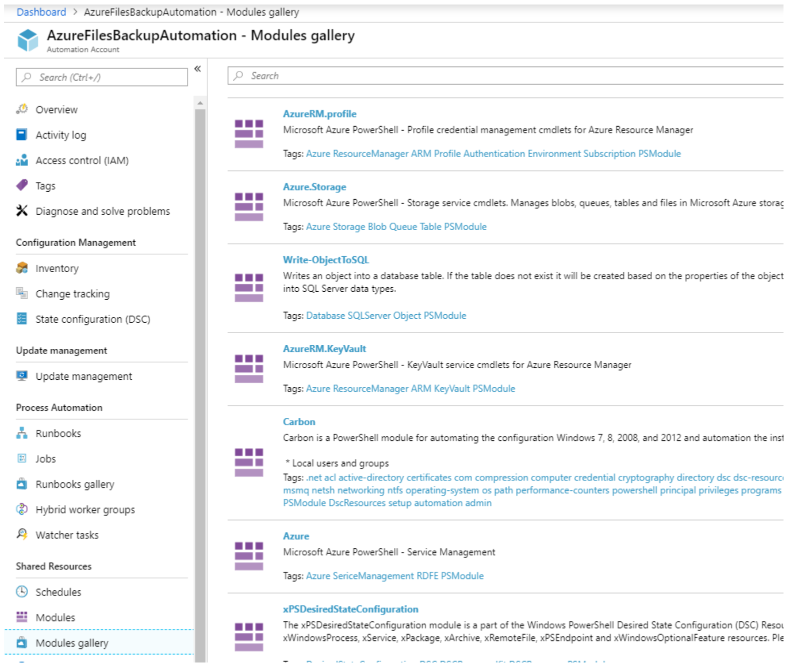
* Location
West US 2

* Create Azure Run As account
[✓ Yes] No

The Run As account feature will create a Run As account and a Classic Run As account. Click here to learn more about Run As accounts.

Learn more about Automation pricing.

Step2: Import modules from Gallery in the Automation resource



AzureFilesBackupAutomation - Modules gallery
Automation Account

AzureRM.profile
Microsoft Azure PowerShell - Profile credential management cmdlets for Azure Resource Manager
Tags: Azure ResourceManager ARM Profile Authentication Environment Subscription PSModule

Azure.Storage
Microsoft Azure PowerShell - Storage service cmdlets. Manages blobs, queues, tables and files in Microsoft Azure storag
Tags: Azure Storage Blob Queue Table PSModule

Write-ObjectToSQL
Writes an object into a database table. If the table does not exist it will be created based on the properties of the object into SQL Server data types.
Tags: Database SQLServer Object PSModule

AzureRM.KeyVault
Microsoft Azure PowerShell - KeyVault service cmdlets for Azure Resource Manager
Tags: Azure ResourceManager ARM KeyVault PSModule

Carbon
Carbon is a PowerShell module for automating the configuration Windows 7, 8, 2008, and 2012 and automation the inst
* Local users and groups
Tags: .net acl active-directory certificates com compression computer credential cryptography directory dsc dsc-resourc msmq netsh networking ntfs operating-system os path performance-counters powershell principal privileges programs PSModule DscResources setup automation admin

Azure
Microsoft Azure PowerShell - Service Management
Tags: Azure SericeManagement RDFE PSModule

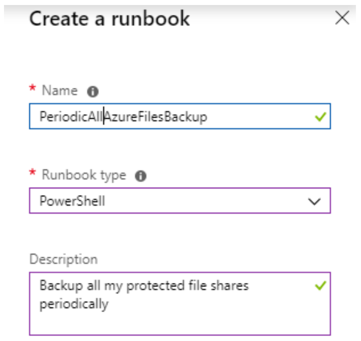
xPSDesiredStateConfiguration
The xPSDesiredStateConfiguration module is a part of the Windows PowerShell Desired State Configuration (DSC) Resou xWindowsProcess, xService, xPackage, xArchive, xRemoteFile, xPSEndpoint and xWindowsOptionalFeature resources. Ple

Import the following modules from the Modules gallery in the order given below:

1. AzureRM.Profile
2. AzureRM.RecoveryServices
3. AzureRM.RecoveryServices.Backup

Step3: Create PowerShell Runbooks in the Automation Resource

You can create multiple Runbooks based on which set of File shares you want to protect. In the example that will be shown below, a runbook is created to enable periodic backups for all protected file shares in a subscription.



Create a runbook

* Name: PeriodicAllAzureFilesBackup

* Runbook type: PowerShell

Description: Backup all my protected file shares periodically

Step4: Edit the Runbook

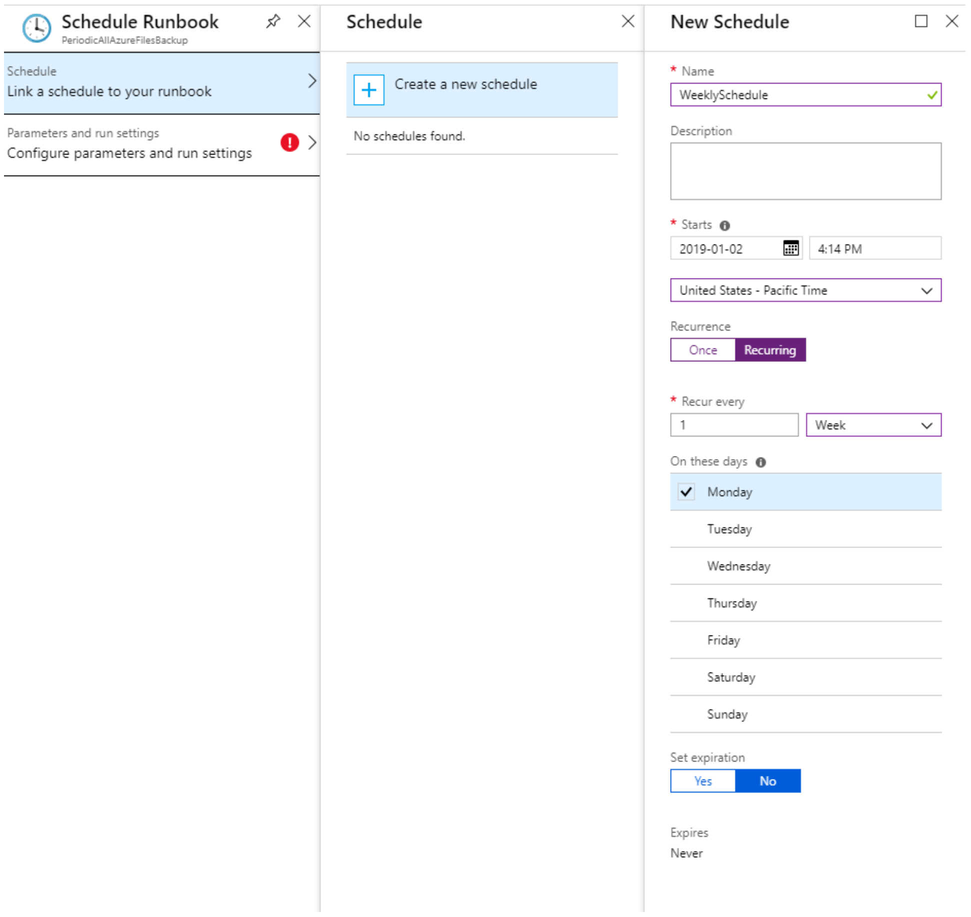
Edit the Runbook and write script to choose which file shares to take a backup. You can create scripts that suit your requirements.

- Save the script
- Test the script using "Test Pane"
- Publish the Runbook

A sample script to take a backup of all file shares in a subscription can be found in Appendix.

Step5: Schedule the Runbook

While scheduling the Runbook, you can pass on the parameters required for the PowerShell Script. The sample script takes the retention as an input. So, if you need to schedule a weekly snapshot and retain for 8 weeks, create a weekly schedule as mentioned below and specify the retention as 56 days (8 weeks). You can do create monthly and yearly schedules (run every 12 months) in a similar manner.



Schedule Runbook
PeriodicAllAzureFilesBackup

Schedule
Link a schedule to your runbook

Parameters and run settings
Configure parameters and run settings

Schedule

Create a new schedule

No schedules found.

New Schedule

* Name
WeeklySchedule

Description

* Starts
2019-01-02 4:14 PM

United States - Pacific Time

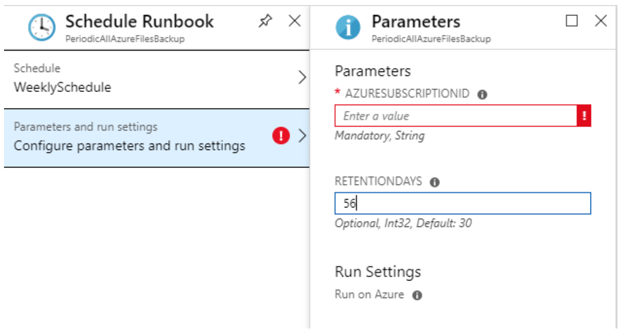
Recurrence
Once [✓ Recurring]

* Recur every
1 Week

On these days
[✓ Monday]
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Set expiration
Yes [✓ No]

Expires
Never


Schedule Runbook
PeriodicAllAzureFilesBackup
Schedule
WeeklySchedule
Parameters and run settings
Configure parameters and run settings

Parameters
PeriodicAllAzureFilesBackup
Parameters
* AZURESUBSCRIPTIONID
Enter a value
Mandatory, String
RETENTIONDAYS
56
Optional, Int32, Default: 30
Run Settings
Run on Azure

You can monitor the success/failure of these backups using the "Jobs" tab of Runbooks

Appendix

Sample Script

```
<#
   .DESCRIPTION
       An example runbook which takes On-demand backup for all file shares protected by Azure Backup in a subscription using
the Run As Account (Service Principal)

   .NOTES
       AUTHOR: Azure Backup
       LASTEDIT: 12/25/18
#>
Param
   (
       [Parameter(Mandatory=$true)][ValidateNotNullOrEmpty()]
       [String]
       $AzureSubscriptionId,
       [Parameter(Mandatory=$true)][ValidateNotNullOrEmpty()]
       [Int]
       $RetentionDays=30
   )

$connectionName = "AzureRunAsConnection"
try
{
   # Get the connection "AzureRunAsConnection "
   $servicePrincipalConnection=Get-AutomationConnection -Name $connectionName

   "Logging in to Azure..."
   Add-AzureRmAccount `
       -ServicePrincipal `
       -TenantId $servicePrincipalConnection.TenantId `
       -ApplicationId $servicePrincipalConnection.ApplicationId `
       -CertificateThumbprint $servicePrincipalConnection.CertificateThumbprint
}
catch {
   if (!$servicePrincipalConnection)
   {
       $ErrorMessage = "Connection $connectionName not found."
       throw $ErrorMessage
   } else{
       Write-Error -Message $_.Exception
```

```
        throw $_.Exception
    }
}

Select-AzureRmSubscription -SubscriptionId $AzureSubscriptionId

#Get all ARM vault resources from all resource groups
$vaults = Get-AzureRmRecoveryServicesVault
$currentDate = Get-Date
$RetailTill = $currentDate.AddDays($RetentionDays)
Write-Output ("Recoverypoints will be retained till " + $RetailTill)

foreach ($vault in $vaults)
{
    Write-Output ("Working on Vault: " + $vault.Name)
    Set-AzureRmRecoveryServicesVaultContext -Vault $vault

    $containers = Get-AzureRmRecoveryServicesBackupContainer -ContainerType AzureStorage
    Write-Output ("Got # of Backup Containers: " + $containers.Count)

    ForEach ($container in $containers)
    {
        Write-Output ("Working on container: " + $container.FriendlyName)
        $fileshares = Get-AzureRmRecoveryServicesBackupItem -WorkloadType AzureFiles -Container $container
        Write-Output ("Got # of Backup Items/shares: " + $fileshares.Count)

        ForEach($fileShare in $fileshares)
        {
          Write-Output ("Working on FileShare: " + $fileShare.Name)
          Backup-AzureRmRecoveryServicesBackupItem -Item $fileShare -ExpiryDateTimeUTC $RetailTill
        }
        Write-Output ("")
    }
    Write-Output ("")
}
```