

# Logon / Startup Script Scan Setup Azure Environments

The following describes how to create a network share in Azure storage to be able to deploy AIT to perform a network inventory scan via logon script when you have an Azure AD environment. This method is meant to replace the initial steps of storage creation for [Logon/Startup](#) script processes when endpoints are not reachable through the corporate network. It covers basic setup for the creation of the storage solution and making it reachable by all end clients that have internet access, further planning may be required depending on different components in your environment if needed.

## Prerequisites

To use an Azure file share with Windows, you must either mount it, which means assigning it mount point path to access it via its UNC path.

On a **Azure storage account**, create a share called **ADSK** to hold 2 folders 1 called **ait** and a folder to hold the collected data, called **data**, here is an of the share and the 2 folders created inside:

### File Share

File share


Refresh

File share settings

Active Directory: Not configured    Soft delete: 7 days    Maximum capacity: 5 TiB    Security: Custom

Search file shares by prefix (case-sensitive)

Show deleted shares

Name	Modified	Tier	Quota	
 adsk	3/15/2022, 1:39:42 PM	Transaction optimized	5 TiB	...

Connect

Upload



Add directory

Refresh

Delete share

Edit quota

Search files by prefix

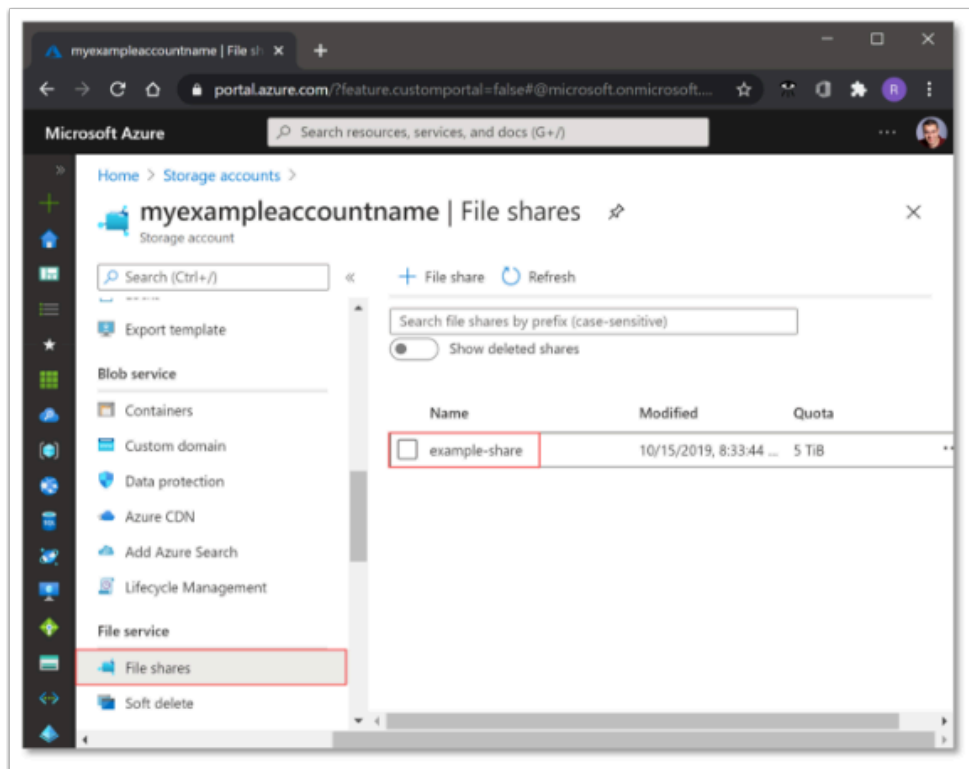
Name
 ait
 data

# Mount the Azure file share

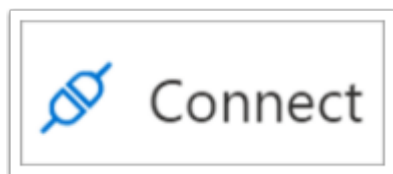
The Azure portal provides you with a script that you can use to mount your file share directly to a host. We recommend using this provided script.

To get this script:

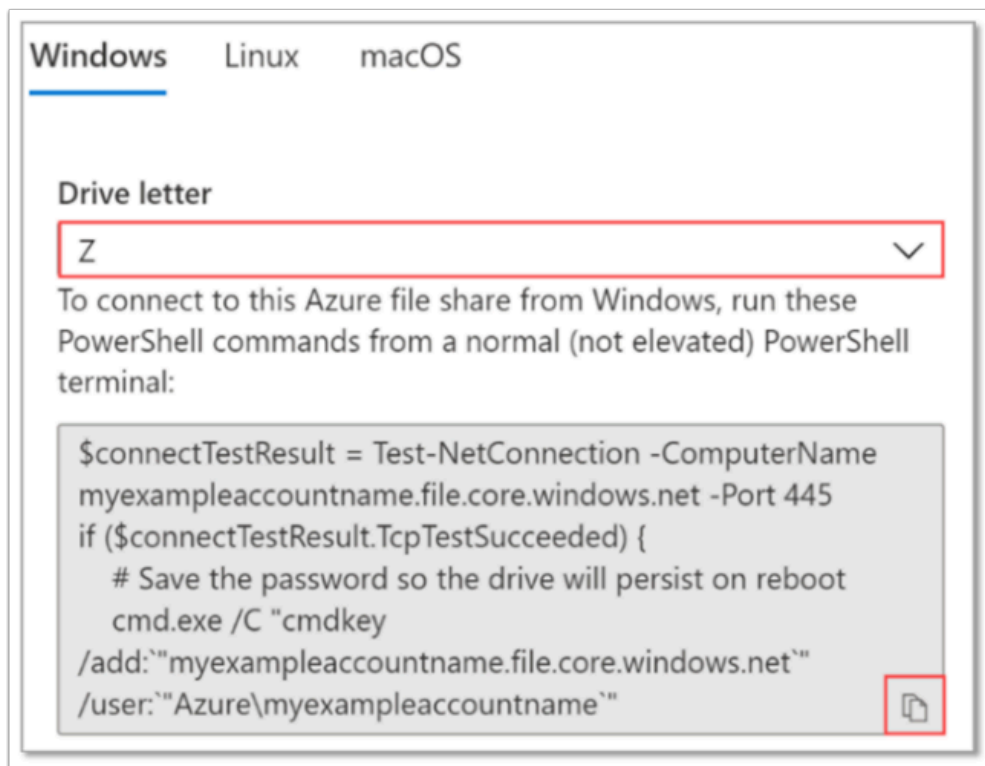
1. Sign in to the [Azure portal](#).
2. Navigate to the storage account that contains the file share you'd like to mount.
3. Select **File shares**.
4. Select the file share you'd like to mount.



5. Select Connect.



6. **Select any drive letter** to mount the share to, as we will not use it for the final setup.
7. Copy the provided script.



8. Autogenerated will look like this:

```
$connectTestResult = Test-NetConnection -ComputerName
aamerlab7158.file.core.windows.net -Port 445
if ($connectTestResult.TcpTestSucceeded) {
    # Save the password so the drive will persist on reboot
    cmd.exe /C "cmdkey /add:"myexampleaccount.file.core.windows.net"
/user:"localhost\myexampleaccount"
/pass:"VZO+RzuszxBqgkAKPcnwxzJ/c1b3EFfEULFTkQFk9riyXyhtfdCPMox1b4X+/LVyG/yfx
L443Vg8JX6AUgjMKA=="
    # Mount the drive
    New-PSDrive -Name Z -PSProvider FileSystem -Root
"\myexampleaccount.file.core.windows.net\adsk" -Persist
} 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."
}
```

Please comment “#” the line highlighted below to avoid mapping the drive letter as it is not needed.

```
$connectTestResult = Test-NetConnection -ComputerName
abarcelonlab3722.file.core.windows.net -Port 445
if ($connectTestResult.TcpTestSucceeded) {
    # Save the password so the drive will persist on reboot
    cmd.exe /C "cmdkey /add:"myexampleaccount.file.core.windows.net"
/user:"Azure\myexampleaccount"
/pass:"Myexampleautogeneratedverylongstorageaccountkey"
    # Mount the drive
    #New-PSDrive -Name Z -PSProvider FileSystem -Root
    "\\myexampleaccount.file.core.windows.net\scanwin" -Persist
} 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."
}
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."
}
```

9. Please add the following lines to the Powershell script before saving

- Comment “#” the line highlighted below to avoid mapping the drive letter as it is not need
- Add “net use \* \\myexampleaccount.file.core.windows.net\adsk” by copying the values in between the quotes in the New-PSDrive command from your storage account generated PS1, and pasting it after the net use \* command as shown below

NOTE: Mapping the share directly without using a mapped drive letter. Some applications may not reconnect to the drive letter properly, so using the full UNC path may be more reliable

```
# Mount the drive
# New-PSDrive -Name Z -PSProvider FileSystem -Root
"\\myexampleaccount.file.core.windows.net\adsk" -Persist
net use * \\myexampleaccount.file.core.windows.net\adsk
} else {
```

It should be something like this:

```

$connectTestResult = Test-NetConnection -ComputerName
myexampleaccount.file.core.windows.net -Port 445
if ($connectTestResult.TcpTestSucceeded) {
    # Save the password so the drive will persist on reboot
    cmd.exe /C "cmdkey /add:"myexampleaccount.file.core.windows.net"
/user:"localhost\myexampleaccount"
/pass:"VZO+RzuszXBqgkAKPcnwzzJ/clb3EFfEULFTkQFk9riyXyMDjNCPMox1b4X+/LVyG/yfx
L443Vg8JX6AUgjMKA=="
    # Mount the drive
    # New-PSDrive -Name Z -PSProvider FileSystem -Root
"\myexampleaccount.file.core.windows.net\adsk" -Persist
net use * \\myexampleaccount.file.core.windows.net\adsk
} 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."
}

```

10. Save it to a file named `ait.ps1` and add it to the [logon/startup](#) group policy created for scanning, as this will map the UNC path and credentials to each user that logs on or starts up their machine.

The File share should be reachable for all clients that have run the Powershell.