**Steps:**

1. Create a Function App (Create a Resource -> Compute -> Function App)
2. Go to Functions in the left navigation
3. Add a Timer Trigger and choose how often you want this function to run

*\*Steps 4-6 will allow the function to run without the user having to log in every time*

1. Go to Identity (under Settings in the navigation on the left side)
2. Turn on Status and click Save
3. Under Permissions, add an Azure Role Assignment (Contributor to the Storage Account the file share you want to autoincrease is located in)

1. Go back to the Function App
2. Click on Functions
3. Select the function you previously created
4. Click on Code+Test in the left navigation
5. Copy and paste the code below:

# Input bindings are passed in via param block.

param($Timer)

# Get the current universal time in the default string format.

$currentUTCtime = (Get-Date).ToUniversalTime()

# The 'IsPastDue' property is 'true' when the current function invocation is later than scheduled.

if ($Timer.IsPastDue) {

Write-Host "PowerShell timer is running late!"

}

#Variable Definitions

$subscription\_id = “subscription ID”

$resource\_group = “resource group name”

$storage\_account\_name = “storage account name”

$file\_share\_name = “file share name”

#Connect to Azure and Import Az Module

Connect-AzAccount -Identity

Import-Module -Name Az.Accounts

Import-Module -Name Az.Storage

Set-AzContext -SubscriptionId $subscription\_id

# Get file share

$StorageContext = New-AzStorageContext -StorageAccountName $storage\_account\_name -Anonymous

$PFS = Get-AzRmStorageShare -ResourceGroupName $resource\_group -StorageAccountName $storage\_account\_name -Name $file\_share\_name -GetShareUsage

# Get provisioned capacity and used capacity

$ProvisionedCapacity = $PFS.QuotaGiB

$UsedCapacity = $PFS.ShareUsageBytes

Write-Host "Provisioned Capacity:" $ProvisionedCapacity

Write-Host "Share Usage Bytes:" $UsedCapacity

# get storage account

$StorageAccount = Get-AzStorageAccount -ResourceGroupName $resource\_group -AccountName $storage\_account\_name

# if less than 20% of provisioned capacity is remaining, increase provisioned capacity by 20%

if (($ProvisionedCapacity - ($UsedCapacity / ([Math]::Pow(2,30)))) -lt ($ProvisionedCapacity\*0.2)) {

$Quota = $ProvisionedCapacity\*1.2

Update-AzRmStorageShare -StorageAccount $StorageAccount -Name $file\_share\_name -QuotaGiB $Quota

$ProvisionedCapacity = $Quota

}

Write-Host "New Provisioned Capacity:" $ProvisionedCapacity

# Write an information log with the current time.

Write-Host "PowerShell timer trigger function ran! TIME:" $currentUTCtime