# Function Find-AzVms ()

## Description

Finds informations about Azure Virtual Machines

## Examples

$idList = @("532rg642-af1b-58j7-8d3c-98658455540t")

$ExportFilePath = "c:\users\john\documents\azvmlist.txt"

# List all VMs informations in a CSV file

Find-AzVms –SubscriptionList $IdList –ExportFileFormat “csv” –ExportFilePath $ExportFilePath -Delimiter ";"

# List all VMs informations in a JSON file

Find-AzVms –SubscriptionList $IdList –ExportFileFormat “json” –ExportFilePath $ExportFilePath

# List VMs names and Ip addresses in a file

Find-AzVms –SubscriptionList $IdList | Select-Object -Property VmName, PrivateIpAddress, PublicIPAddress | Export-CSV –path $ExportFilePath

# Store all Windows VMs in an array

$AzVmList = Find-AzVms –SubscriptionList $IdList | Where-Object {$\_.OsType -EQ "Windows"}

## Parameters

$SubscriptionId [array] : list of Azure subscriptions ids  
$ExportFilePath [string] : output file (optional)  
$Delimiter [string] : fields delimiter (optional, CSV only)

## Inputs

None

## Outputs

$AzVmList [array] : array with virtual machines informations  
CSV File (–ExportFileFormat “csv”)  
JSON File (–ExportFileFormat “json”)

## Notes

None

## Dependencies

## Az.Compute, Az.Accounts, Az.Network (Powershell modules)

## Todo

* Write similar function to « get » virtual machines or add it to this one
* Format Subscription name in Json export
* Add help
* Output to generic file name if exportfilepath not specified with exportfileformat

# Function New-BlobContainer ()

## Description

Create Blob Container in specified existing or new Resource Group and Storage Account

## Examples

$Container = New-BlobContainer -Region "francecentral" `

-ResGroupName "myResGroup" `

-StorageAccountName "mystorageaccount" `

-ContainerName "mycontainer"

## Parameters

[string] Region : Azure location

[string] ResGroupName : Resource Group name (will be created if doesn’t exist)

[string] StorageAccountName : Storage Account name (will be created if doesn’t exist in specified Resource group)

[string] ContainerName : New container name (will stop with error if already exists)

## Inputs

None

## Outputs

Container object if created, $False if error was encountered

## Notes

Must be connected to Azure (e.g Connect-AzAccount)

List available region : *Get-AzRegion | select Region | where-object {$\_.Region -match "france"}*

## Dependencies

## Az.Accounts (Powershell module)

## Todo

# Function Set-BlobContent ()

## Description

Upload Blob content on specified Azure Storage container

## Examples

# set vars  
$StorageAccountName = "mystorageaccountname"

$StorageAccountKey = "F1OlgCjAfHJK7DfFES3qefzfzfzQgggezMl7QWlZYjwrzJKe4yqI5NWneZg=="

$ContainerName = "mycontainer"  
$SourceFile = "C:\documents\file1.txt"

$BlobName = "testblobfile01.txt"

# call function

$upload = Set-BlobContent -StorageAccountName $storageaccountname `

-StorageAccountKey $storageaccountkey -ContainerName `

$containername -SourceFile $sourcefile -BlobName $blobname

## Parameters

## [string] StorageAccountName : Storage account name

## [string] StorageAccountKey : Storage account key

## [string] ContainerName : Container name

## [string] SourceFile : full path to local file to upload

## [string] BlobName : name of blob on destination container

## Inputs

None

## Outputs

AzureStorageBlob if successfully uploaded, or $False if error was encountered

## Notes

Existing blob content will be overwritten !

## Dependencies

## Az.Storage (Powershell module)

## Todo

# Function Get-BlobContent ()

## Description

Download Blob content on specified Azure Storage container

## Examples

# set vars  
$StorageAccountName = "mystorageaccountname"

$StorageAccountKey = "F1OlgCjAfHJK7DfFES3qefzfzfzQgggezMl7QWlZYjwrzJKe4yqI5NWneZg=="

$ContainerName = "mycontainer"  
$BlobName = "testblobfile02.txt"

$DestinationFile = "C:\documents\file2.txt"

# call function

$upload = Get-BlobContent -StorageAccountName $storageaccountname `

-StorageAccountKey $storageaccountkey -ContainerName `

$containername -BlobName $blobname -DestinationFile $destinationfile

## Parameters

## [string] StorageAccountName : Storage account name

## [string] StorageAccountKey : Storage account key

## [string] ContainerName : Container name

## [string] BlobName : name of blob on source container

## [string] DestinationFile : full path to local destination file

## Inputs

None

## Outputs

AzureStorageBlob if successfully downloaded, or $False if error was encountered

## Notes

Existing local file will be overwritten !

## Dependencies

## Az.Storage (Powershell module)

## Todo