# Analysis Service deployment

# Powershell

$TabularEditorDownloadUrl = "https://cdn.tabulareditor.com/files/TabularEditor.2.20.2.zip"

$DownloadDest = join-path (get-location) "TabularEditor.zip"

Invoke-WebRequest -Uri $TabularEditorDownloadUrl -OutFile $DownloadDest

Expand-Archive -Path $DownloadDest -DestinationPath (get-location).Path

Remove-Item $DownloadDest

-- Batch Windows

-- Use the Azure Key Vault component to get key vault secrets and instantiate them as environment variables

-- ($ASConnectionString and $asDatabase) in a DevOps pipeline

TabularEditor.exe "$(System.DefaultWorkingDirectory)/\_as-edwcore" -D "$(ASConnectionString)" "$(asDatabase)" -O -P -W -E

# Powershell

# Compose TMSL script for SSAS

$sqlPasswordAllCharacters = @'

$(sqlPassword)

'@

$TMSL = @"

{

"createOrReplace": {

"object": {

"database": "$(asDatabase)",

"dataSource": "DS-DV-DMT"

},

"dataSource": {

"type": "structured",

"name": "DS-DV-DMT",

"connectionDetails": {

"protocol": "tds",

"address": {

"server": "$(sqlServerName)",

"database": "$(sqlDatabase)"

}

},

"credential": {

"kind": "SQL",

"path": "$(sqlServerName);$(sqlDatabase)",

"AuthenticationKind": "UsernamePassword",

"Username": "$(sqlUser)",

"Password": "$sqlPasswordAllCharacters",

"EncryptConnection": true

}

}

}

}

"@

# Execute the TMSL script on SSAS

Invoke-ASCmd -ConnectionString "$(ASConnectionString)" -Query $TMSL

# Training Resourses Deployment Requirements

1. Powershell - Mandatory

https://learn.microsoft.com/en-us/powershell/scripting/install/install-debian?view=powershell-7.5

1. Azure Powershell Module - Mandatory

https://learn.microsoft.com/en-us/powershell/azure/install-azps-linux?view=azps-14.4.0

1. Microsoft Command Line Tools - Mandatory

https://learn.microsoft.com/en-us/sql/linux/sql-server-linux-setup-tools?view=sql-server-ver17&tabs=ubuntu-install%2Codbc-ubuntu-1804

1. Azure CLI - Optional

# Microsoft Tools on Ubuntu

## Powershell

# Update the list of packages

sudo apt-get update

# Install pre-requisite packages.

sudo apt-get install -y wget apt-transport-https software-properties-common

# Get the version of Ubuntu

source /etc/os-release

# Download the Microsoft repository keys

wget -q https://packages.microsoft.com/config/ubuntu/$VERSION\_ID/packages-microsoft-prod.deb

# Register the Microsoft repository keys

sudo dpkg -i packages-microsoft-prod.deb

# Delete the Microsoft repository keys file

rm packages-microsoft-prod.deb

# Update the list of packages after we added packages.microsoft.com

sudo apt-get update

# Install PowerShell

sudo apt-get install -y powershell

# Start PowerShell

pwsh

echo 'export alias powershell=pwsh' >> ~/.bashrc

source ~/.bashrc

## Azure Powershell Module

* Install: Install-Module -Name Az -Repository PSGallery -Force
* Update: Update-Module -Name Az -Force

## Microsoft CLI Tools

curl https://packages.microsoft.com/keys/microsoft.asc | sudo tee /etc/apt/trusted.gpg.d/microsoft.asc

curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list | tee /etc/apt/sources.list.d/mssql-release.list

sudo apt-get update

sudo apt-get install mssql-tools18 unixodbc-dev

sudo apt-get update

sudo apt-get install mssql-tools18

echo 'export PATH="$PATH:/opt/mssql-tools18/bin"' >> ~/.bashrc

source ~/.bashrc

## Azure CLI

sudo apt-get update

sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release

sudo mkdir -p /etc/apt/keyrings

curl -sLS https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor | sudo tee /etc/apt/keyrings/microsoft.gpg

sudo chmod go+r /etc/apt/keyrings/microsoft.gpg

AZ\_DIST=$(lsb\_release -cs)

echo "Types: deb

URIs: https://packages.microsoft.com/repos/azure-cli/

Suites: ${AZ\_DIST}

Components: main

Architectures: $(dpkg --print-architecture)

Signed-by: /etc/apt/keyrings/microsoft.gpg" | sudo tee /etc/apt/sources.list.d/azure-cli.sources

sudo apt-get update

sudo apt-get install azure-cli

# Operations

## Resource Usage

$tenant\_id = "bfcc27e6-aeae-460b-a71c-0e532f5d2555"

$subscription\_id = "010b2eb9-a8b8-48b5-a0d9-f0a65850eab0"

$location = "westeurope"

az login --use-device-code --tenant "$tenant\_id"

$location\_name=$(Get-AzLocation | Where-Object {$\_.Location -eq "$location"}).DisplayName

$json\_formatted\_string = az quota list --scope "/subscriptions/$subscription\_id/providers/Microsoft.Compute/locations/$location" | Out-String

$json = ConvertFrom-Json -InputObject $json\_formatted\_string

foreach ($structure in $json) {

Write-Host "Limit for"$structure.name.Trim()":"$structure.properties.limit.value"units"

}

foreach ($structure in $json) {

$out\_string = "Limit for " + [String]$structure.name + ": " + $structure.properties.limit.value + " units"

Write-Host $out\_string

}

concat(<inputValue>, <inputValue>[, <inputValue>...])

$strings\_array = az quota list --scope "/subscriptions/$subscription\_id/providers/Microsoft.Compute/locations/$location"

$strings\_array.GetType()

foreach ($structure in $json\_array) {

$structure

}

$indx = 1

foreach ($structure in $json\_array) {

Write-Host "position $indx "

$indx = $indx + 1

}

$json\_string = ConvertTo-Json -InputObject $(az quota list --scope "/subscriptions/$subscription\_id/providers/Microsoft.Compute/locations/$location")

$json\_string.GetType()

ConvertTo-Json

[-InputObject] <Object>

[-Depth <Int32>]

[-Compress]

[-EnumsAsStrings]

[-AsArray]

[-EscapeHandling <StringEscapeHandling>]

[<CommonParameters>]

ConvertFrom-Json -InputObject $json\_string -AsHashtable

az vm list-usage --location "$location\_name" -o table

az sql list-usages --location "$location" -o table

az sql db list-editions -l "$location" -o table

az resource show --resource-group ... --name ... --resource-type Microsoft.DataFactory/factories

az group export --resource-group ... --resource-ids ...

Connect-AzAccount -Tenant "$tenant\_id" -SubscriptionId "$subscription\_id"