Automating Azure Data Platform Deployments with Terraform

1. Introduction

Hi, there's a few things you need to install in preparation for the workshop, some are optional but recommended. You will find them all listed below.

If you have any questions, please email falekmiah@hotmial.co.uk & wykesanna32@gmail.com.

2. Prerequisite

2.1. Applications

Download and install the following on laptop:

- Visual Studio Code
- Terraform CLI (Latest version greater than v1.1.2)
- Azure CLI
- Terraform Extension in Azure DevOps

2.2. Accounts

Active **Azure** account

- We recommend you create a free tier account using a valid email account to avoid any company policies preventing Terraform actions.
- You will need to be 'owner' or 'contributor' of the subscription

Active **Azure DevOps** project

- We recommend you pre-clone your repository
- Azure DevOps Service Connection that uses a Service Principal, we recommend the Service Principal have full access to the subscription, so 'owner' or 'contributor'. We also recommend you add the Service Principal ID and Secret to your Key Vault (listed in section 1.3 Azure Resource) in advance.

  The following Azure CLI snippet will help you create a Service Principal:

```
$subscriptionId = "00000000-0000-0000-0000-000000000000"
$appName        = "<service principal name>

az ad sp create-for-rbac `
   --name $appName `
   --role="<Owner/Contributor>" `
   --scopes="/subscriptions/$subscriptionId"
```

2.3. Azure Resources

- Resource Group with:
  - Key Vault
  - Storage Account

The Key Vault must contain the following secrets:

Secret Name	Populate with
ARM-CLIENT-ID	Service Principal ID (same SP as used in DevOps Service Connection)
ARM-CLIENT-SECRET	Service Principal Secret (same SP as used in DevOps Service Connection)
ARM-TENANT-ID	Azure AD Tenant ID
ARM-SUBSCRIPTION-ID	Azure Subscription ID

3. Optional

3.1. Remote-Container

To ensure you have a clean local environment you recommend you use a remote container, this is optional.

If you have remote container then ensure you have Azure CLI and Terraform features enabled for the latest.

If you need to install Remote-Container, then the below are some steps to help create a remote container for Terraform:

- Install "Remote - Containers" extension in Visual Studio Code
  https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-containers

- You will need to install Docker for Remote-Containers:
  - Install from website directly: https://docs.docker.com/desktop/windows/install/
  - Alternativity, the following chocolatey command in an administrator PowerShell will install it:

```
choco install docker-desktop
```

- Install WSL (Windows Subsystem for Linux) and Ubuntu.
  - https://docs.microsoft.com/en-gb/windows/wsl/install
  - https://docs.microsoft.com/en-gb/windows/wsl/install-manual

- Select "Open folder in container" and select the directory.
  - If first time you will prompt to "Add Development Container Configure File" to your directory, include Azure CLI and Terraform definitions. This will create the .devcontainer folder and devcontainer.json & Dockerfile files.