# Install Terraform

This guide will show you how to install Terraform on your workstation with access through the zscaler proxy to allow terraform to download providers.

The cloud platform team take a different approach where they use a docker image with terraform and providers pre-installed. Contact the cloud platform team if you want to know more about their approach.

## Installation on Developer PC

1. Install Visual Studio Code (or your editor of choice): https://code.visualstudio.com/download (use system installer, 64 bit)
2. Install Azure CLI: https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest
3. Install Terraform: https://www.terraform.io/downloads.html
    1. For Windows download AMD64
    2. Unzip to C:\Programs\Terraform or C:\Bin (i.e. a location you like to keep command line tools)
4. Update the path environment variable to include the path to Terraform, easiest way is to:
    1. Open File Explorer > My PC > Properties > Advanced system settings > Environment Variables
    2. Find the "Path" setting under System variables section and click edit
    3. Click the new button
    4. Enter the folder where you have installed Terraform, eg C:\Programs\Terraform
5. Configure the proxy so that Azure CLI can get through the Westpac Zscaler proxy and verify the connection

    Open command line run the following two commands:

    ```
    SETX HTTPS_PROXY http://whitelist.proxy.prod.westpac.co.nz:8080
    SETX NO_PROXY .westpac.co.nz,.westpactrust.co.nz
    ```

    > The proxy settings can affect a number of application so if an application fails to work correctly this is one of the first places to look. For example you may need to add localhost,127.0.0.1 to your NO_PROXY setting for connecting to services running on your machine for development.
    >
    > For more information on Wesptac proxies see Proxies.

    To prevent certificate errors when running "terraform init":

    a. Download azure-management.pem and place it somewhere on your PC

    b. From command-line run:

    ```
    SETX REQUESTS_CA_BUNDLE C:\<path>\azure-management.pem
    ```

    Close CMD window and reopen
6. Optionally, start Visual Studio Code and install the Terraform extension
7. From command line (use either cmd window or terminal within Visual Studio Code) run "az login", a browser window should open, enter your credentials for the azure dev/test subscription. This will automatically close after 10 seconds and take you back to the command window where you will see Json output with your user name and the Azure subscription and tenant you've connected to
8. Run "terraform version" to ensure terraform is setup and available in the path
9. Request "Restricted-UAC-Bitbucket-Developer" entitlement in IIQ if you don't already have it (this is required to access the stash based terraform modules).

> When working from home you will need to authenticate with the zscaler proxy before being able to run "terraform init" or other commands. This is a simple as navigating to an external web page such as https://www.stuff.co.nz/.

## Set Plugin Cache

To prevent Terraform from downloading providers from the internet every time you run "terraform init"...

1. Edit / add configuration file: C:\Users\<your salary id>\AppData\Roaming\terraform.rc
2. Add the line: plugin_cache_dir = "$APPDATA/terraform.d/plugin-cache"
    One way of doing steps 1 and 2 in Windows machines: open cmd and run this command from:

    ```
    C:\Users\<your salary id>\AppData\Roaming>echo plugin_cache_dir = "$APPDATA/terraform.d/plugin-cache" > terraform.rc
    ```
3. Create the directory: C:\Users\<your salary id>\AppData\terraform.d\plugin-cache" (you'll probably find you already have the terraform.d directory)

See https://www.terraform.io/docs/commands/cli-config.html for more information.

## Configure Powershell

If your Terraform configurations are targeting D365 then you'll need to ensure that Powershell is setup correctly:

1. Open Powershell running as administrator.
2. Run `Get-PSRepository` and ensure the output looks something like this, if not you will need to run `Register-PSRepository -Default`

   ```
   Name       InstallationPolicy    SourceLocation
   -------    ------------------    --------------
   PSGallery  Untrusted             https://www.powershellgallery.com/api/v2
   ```

3. Run `Get-PackageProvider -Name NuGet` and ensure the output looks something like this, if not you will need to run `Install-PackageProvider -Name "NuGet"`

   ```
   Name    Version     DynamicOptions

   ----    -------     --------------
   NuGet   2.8.5.208   Destination, ExcludeVersion, Scope, SkipDependencies, Headers, FilterOnTag,
   Contains, AllowPrereleaseVersions, ConfigFile, SkipValidate
   ```

The D365 Terraform modules are designed to automatically install any Powershell modules that they need, however to do that the first time you run them you must either:

1. Run `terraform apply` as administrator
2. Run `Install-Module -Name Microsoft.Xrm.Data.PowerShell -Force -Verbose -Scope AllUsers -RequiredVersion 2.8.11 -ErrorAction Stop` NOTE: the version the D365 modules are using may change, but this is what was in use at the time these instructions were written.