Practice Development Unit

Technical Enablement

Pre-requisites

Student Guide

May 2017

© 2017 Microsoft Corporation. All rights reserved. This document is confidential and proprietary to Microsoft. Internal use only. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples are for illustration only and are fictitious. No real association is intended or inferred.

## Contents

[Pre-requisites 1](#_Toc483384371)

[Azure Subscription 3](#_Toc483384372)

[Install the Azure PowerShell module 5](#_Toc483384373)

[Install Windows 10 Linux subsystem and CLI 2.0 6](#_Toc483384374)

[Add Git to the Windows 10 Linux Subsystem 8](#_Toc483384375)

[Install Git Bash 8](#_Toc483384376)

[Install Visual Studio 2017 9](#_Toc483384377)

[Install Visual Studio Code 10](#_Toc483384378)

[Adding the Microsoft Azure SDK to Visual Studio 2015 11](#_Toc483384379)

[Creating a Visual Studio VM in Azure 12](#_Toc483384380)

[Configure Visual Studio for ARM Template Authoring 14](#_Toc483384381)

# Pre-requisites

## **IMPORTANT**

## PLEASE READ THIS SECTION FULLY - NOT ALL THE PRE-REQS WILL APPLY TO ALL SESSIONS OR OPERATING SYSTEMS, AND THEREFORE ONLY A SUBSET OF THE FOLLOWING INSTRUCTIONS NEED TO BE FOLLOWED

## Overview

All students must complete the required pre-requisites for the training prior to attendance, to maximise the time and value of the session itself.

Note that this student guide covers a number of varied technical enablement sessions.

Please refer to the communication for the session to determine if the requirementsfor that session are a subset of the list below.

## Always Required

An active **Azure subscription**. You will require **one** of the following:

* [Visual Studio subscription](#_Visual_Studio_Subscription) (preferred),
* [Internal usage rights](#_Internal_Usage_Rights) for silver and gold partners,
* [Free Azure trial account](#_Free_Azure_Trial)

## Optional Requirements

Refer to the requirements list specified by the trainer to determine whether any or all of the following optional pre-reqs are also a requirement for the workshop session.

* **PowerShell**
  + Ensure a recent version of PowerShell is installed and [add the Azure modules](#_Install_the_Azure)
* **Azure CLI 2.0** 
  + The Cloud Shell built into the Azure portal is the simplest option for most users as it has no install requirements
  + Power users on Windows 10 may wish to install the [Ubuntu Bash Subsystem](#_Install_Ubuntu_Bash), and then install Azure CLI 2.0 locally
* **Git**
  + For Windows 10 power users that have the Windows 10 Linux Subsystem installed, then [add git using apt-get](#_Add_Git_to)
  + If not, then [install Git Bash](#_Install_Git_Bash_1)
* **JSON template editor**. It is recommended to have **one** of the following:
  + [Visual Studio 2017](#_Install_Visual_Studio)
  + [Visual Studio Code](#_Install_Visual_Studio_1)

## Additional Information

* If Visual Studio 2015 is already installed, then this may be used instead for JSON editing if the [Azure SDK](#_Adding_the_Microsoft) is added
* For users that do not have full admin rights to their laptop, then there are instructions on how to [create a VM in Azure](#_Creating_a_Visual) that includes Visual Studio
* There are recommended instructions for [configuring the view in Visual Studio](#_Configure_Visual_Studio) for JSON editing

# Azure Subscription

## Visual Studio Subscription

The Visual Studio Enterprise subscription is the replacement for MSDN subscriptions, and should only be used for test/dev scenarios. A Microsoft partner with the Cloud Platform competency receives a number of Visual Studio 2017 Enterprise with MSDN subscriptions:

|  |  |
| --- | --- |
| Cloud Platform Competency | Internal Usage Rights |
| Silver | 10 |
| Gold | 35 |

(core benefit plus Cloud Platform competency benefit)

The Global Administrator for the [Partner Membership Center](https://partners.microsoft.com/) can assign the subscription under Requirements & Assets 🡪 Assign Privileges, as per the competency partner section of the [support page](https://support.microsoft.com/en-gb/help/4013871/microsoft-partner-network-mpn-visual-studio-subscriptions?tpqid=800-000036).

Visual Studio Enterprise subscribers are entitled to $150 of Azure credits per month.

To activate the benefit:

* Go to <https://my.visualstudio.com/benefits>
* Click on Activate on the Azure tile
* Follow the sign up wizard

No credit card is required for the activation. Your subscription will be disabled if you exceed the $150/mth limit, although it will be re-enabled the following month if usage is reduced.

## Internal Usage Rights

Microsoft Cloud Platform partners are also entitled to [internal usage rights](https://azure.microsoft.com/en-us/pricing/member-offers/mpn-benefits/) Azure credits.

|  |  |
| --- | --- |
| Cloud Platform Competency | Internal Usage Rights |
| Silver | $6,100 |
| Gold | $12,100 |

(core benefit plus Cloud Platform competency benefit)

The internal usage rights ([IUR](http://aka.ms/iur)) credits may be used for

* internal business use
* customer demo
* internal development and testing
* internal training for employees and customers

The internal usage rights may be activated at <http://aka.ms/ActivateIUR>.

## Free Azure Trial Account

An Azure trial account may be created. The trial account provides $200 of free credits, and may be used to try out any of the Azure services for 30 days.

Visit <https://azure.microsoft.com/en-gb/free> to set up the account.

Credit card details need to be provided for identity verification, but the spending limit is set to $0 to ensure that it remains free. The spending limit may be changed if the account is going to be used ongoing.

## Verification

Once the account is enabled, prove that it is working correctly by logging into the portal at <http://portal.azure.com>.

# Install the Azure PowerShell module

All current Windows desktop operating systems have PowerShell installed. Open up either the PowerShell prompt, or the PowerShell ISE (interactive scripting environment) as an Administrator. (Note that PowerShell is also available for Linux and MacOS.)

## Install PowerShellGet

PowerShellGet is included in Windows Management Framework 5, which includes PowerShell 5.x.

* Run Get-Module PowerShellGet -list | Select-Object Name,Version,Path to confirm that PowerShellGet is installed and the version is 1.0.0.1 or later
* If PowerShellGet is not installed, then install [WMF 5.0](https://www.microsoft.com/en-us/download/details.aspx?id=50395)

## Install or Update Azure PowerShell

Make sure that PowerShell is still open with admin privileges.

* Run Install-Module AzureRM
* If the AzureRM module is already installed, then update with Update-Module AzureRM
* The available versions may be listed using Get-Module AzureRM -ListAvailable
* Older versions may be uninstalled using Uninstall-Module AzureRM -RequiredVersion 3.3.0, where 3.3.0 is the version of the AzureRM module being uninstalled

## Verify the installation

Note that the AzureRM module isn’t imported by default.

* Import-Module AzureRM will import the module for use
* If there are multiple versions of the module side by side then the version may be specified using, for example, Import-Module AzureRM -RequiredVersion 3.8.0
* Get-Module AzureRM will confirm if the module is loaded and which version number
* Get-Command -Module AzureRM will list all the available AzureRM commands
* Type Login-AzureRmAccount and follow the dialog to log in andl show the subscription name

NB. Check your Execution Policy settings using Get-ExecutionPolicy -List | Format-Table -AutoSize if you receive a PowerShell execution error. The Set-ExecutionPolicy RemoteSigned command should set the correct policy.

The instructions are from <https://docs.microsoft.com/en-us/powershell/azure/install-azurerm-ps>.

# Install Windows 10 Linux subsystem and CLI 2.0

There are many options for [installing Azure CLI 2.0](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli). This section has **Windows 10** instructions for installing the Ubuntu Bash Linux subsystem on Windows 10 and configuring Azure CLI 2.0.

## Install Bash on Windows

This requires 64 bit Windows 10 with Anniversary Update (minimum), Creators Update (highly recommended.) (Settings 🡪 System 🡪 About, OS Build >= 14393.)

* Turn on Developer Mode
  + Settings 🡪 update and Security 🡪 For Developers
  + Click the radio button for Developer Mode
* Enable the Windows Subsystem for Linux, either
  + GUI: Select Windows Subsystem for Linux in “Turn Windows features on or off”
  + PowerShell (as Administrator):

Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

* + Restart Windows

## Initialise Bash

* Run Bash on Windows by typing bash in a Command Prompt
* Accept the licence and type y to continue (it will be installed to %localappdata%\lxss\)
* Create a UNIX username and password (separate to your Windows username and password)

## Modify the apt sources

* Add to the source list (single wrapped command)

echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ wheezy main" | sudo tee /etc/apt/sources.list.d/azure-cli.list

Note that there is a space before “wheezy”

* Add the key and https transport

sudo apt-key adv --keyserver packages.microsoft.com --recv-keys 417A0893

sudo apt-get install apt-transport-https

* Update the package list and then install Azure CLI 2.0

sudo apt-get update && sudo apt-get install azure-cli

## Optional: Change font and vi colours

The default colours for both the PS1 prompt and for vi and vim can be difficult to read. If you find that to be the case then follow the instructions below.

* Edit ~/.bashrc (using nano, vi, or vim) and then scroll to the color\_prompt section.
  + The PS1 prompt colours are set in the sections that are in the format [01:34m\]. The 34 is light blue, which is hard to read. Changing the number from 34 to 36 (cyan) or 33 (yellow) will be more readable. (Info from [here](http://tldp.org/HOWTO/Bash-Prompt-HOWTO/x329.html).)
* For vi(m) users then creating a .vimrc file will also help to set a more readable colour scheme

umask 022

echo -e "colo murphy\nsyntax on" >> ~/.vimrc

## Verify the installation

* Type az to show the base commands
* Type az login and follow the instructions to log in to Azure
* Type az account list to show the subscription info in JSON output format

# 

# Add Git to the Windows 10 Linux Subsystem

* Run Bash on Windows by typing bash in a Command Prompt
* Ensure the package list is up to date and then install the basic Git tools

sudo apt-get update && sudo apt-get install git-all

* Verify by opening Git Bash from the Start Menu and typing git to see the base commands

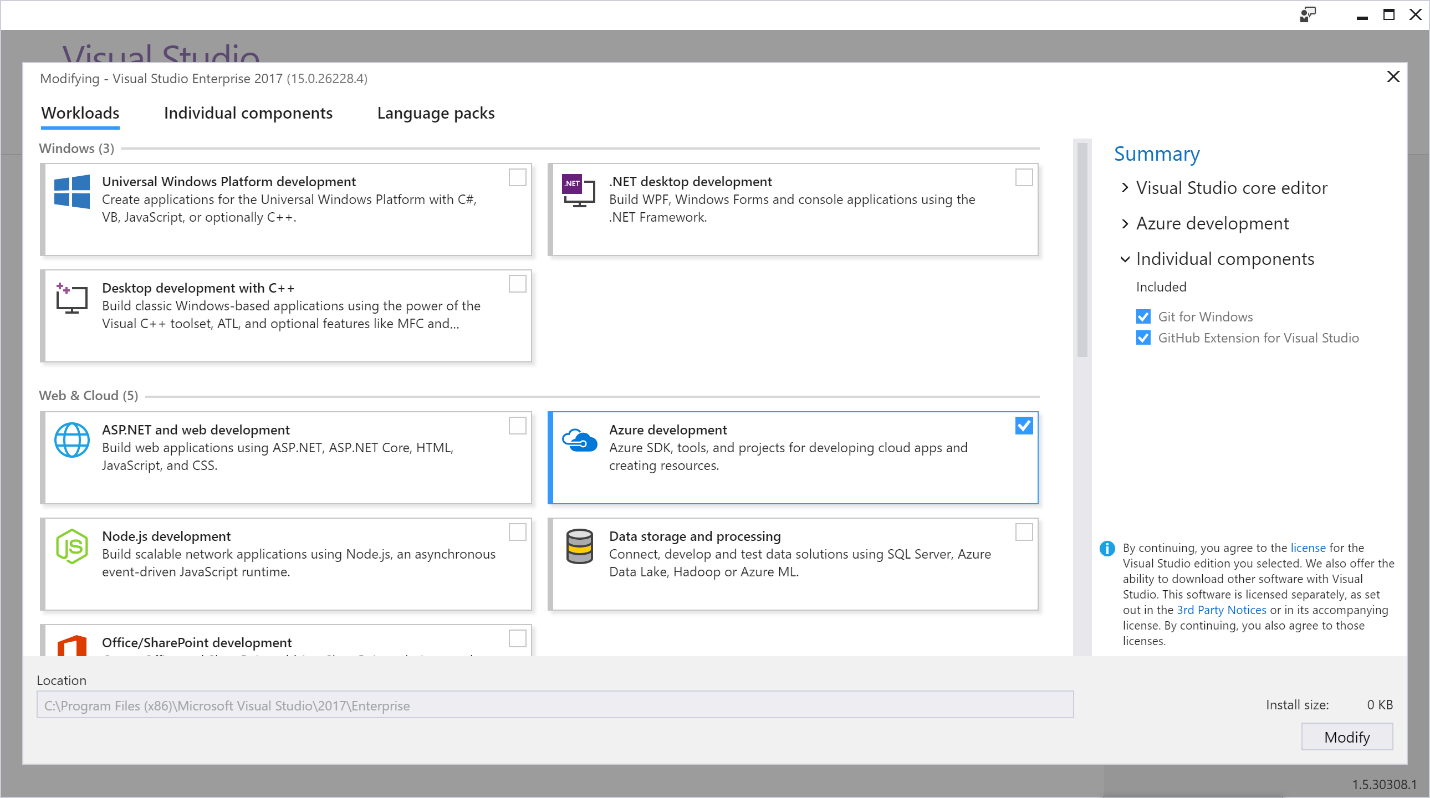
# Install Git Bash

* Download and install Git for Windows from <https://git-scm.com/download/win>
  + Use the defaults on the Select Components screen
  + Select Use Git from Git Bash to avoid PATH modification
  + Defaults for OpenSSH, OpenSSL, line endings, terminal emulator, extra options
* Verify by opening Git Bash from the Start Menu and typing git to see the base commands

# Install Visual Studio 2017

The installation steps for Visual Studio 2017 are the same for Enterprise, Professional, and Community editions. Enterprise has been assumed for the instructions.

* Download the Visual Studio Enterprise 2017 Installer from <https://my.visualstudio.com/benefits>
* Running the installer may prompt an update to itself
* Whilst installing Visual Studio Enterprise 2017, check the “Azure development” workload from the Web & Cloud section.



If Visual Studio is already installed, then you may rerun the installer to add the Azure workload:

* Click Start and then the Visual Studio Installer icon
* Click Update in the Visual Studio Enterprise 2017 tile if shown
* Click Modify in the Visual Studio Enterprise 2017 tile
  + If Modify is not shown then you may find it in the options icon denoted by three horizontal lines
* Select the Azure development workload and install
* [Configure Visual Studio for ARM Template Authoring](#_Configure_Visual_Studio) once it has installed

# Install Visual Studio Code

Visual Studio Code is a free alternative source code editor to Visual Studio. It is a smaller, more lightweight, and cross platform application. Visual Studio Code is optimized for building and debugging modern web and cloud applications but may also be used for ARM template authoring.

Whilst VS Code is not as Azure integrated as Visual Studio, it does includes IntelliSense code completion and colour coding for JSON templates. It can be integrated further by adding the Azure Resource Manager Tools extension. And you can then make use of the Azure quickstart templates on GitHub and the ARM template references for the various Azure service providers on the Azure docs site.

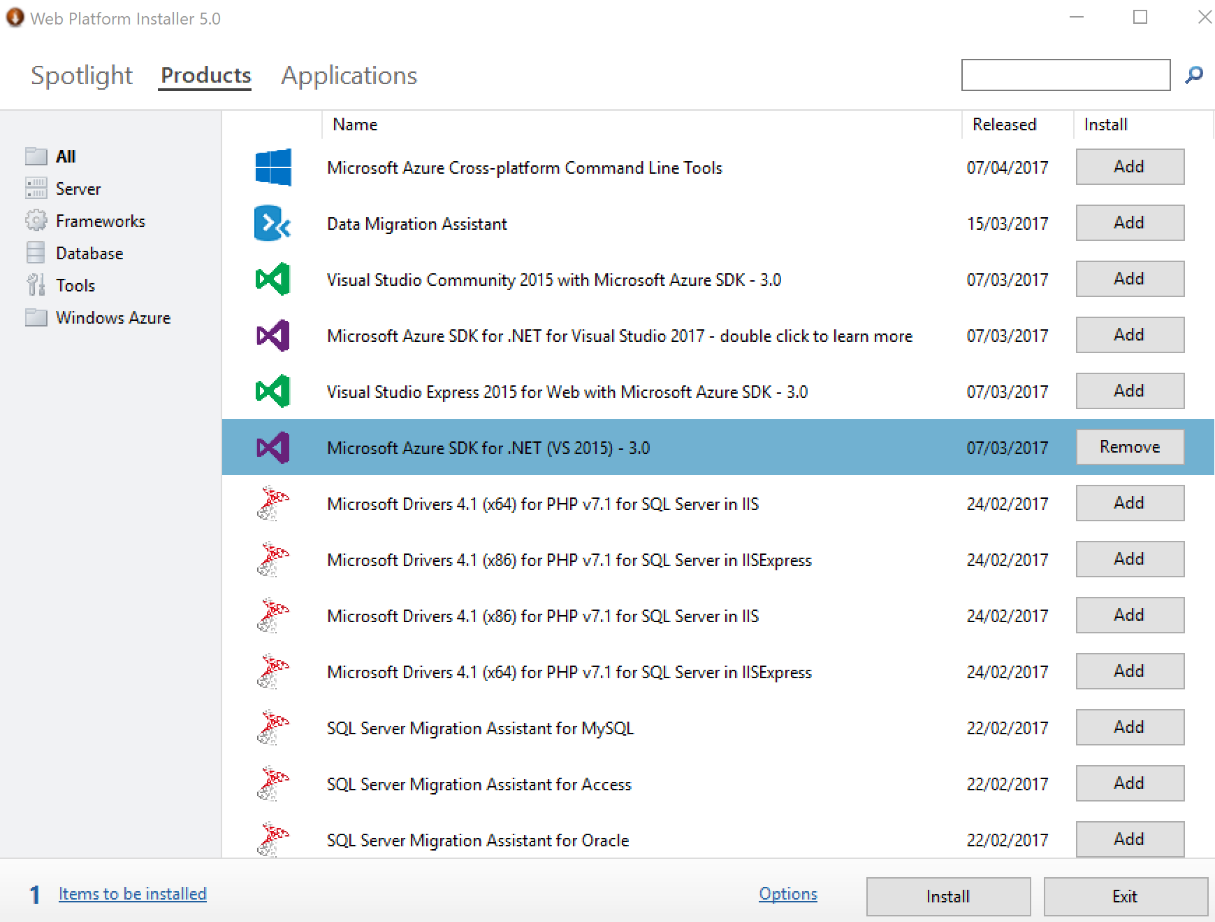
* Install VS Code from <https://code.visualstudio.com>.
* Install the Azure Resource Manager Tools extension
  + Quick Open (Ctrl+P)
  + Enter ext install azuretoolsforvscode
  + Clicking on the green install button
* Reload VS Code when prompted to enable the extension
* Type Ctrl-Shift-P to open the Command Palette. Examples:
  + Typing azure shows all of the available extension commands
  + Typing login will bring up the Azure login
  + Typing active will set the default Azure datacentre
  + Typing search will bring up the Azure Quickstart template search
* Whilst typing in the main body of an Azure Resource Manager JSON template, the editor will show IntelliSense for the available commands, and will also show snippets that can be quickly inserted

# Adding the Microsoft Azure SDK to Visual Studio 2015

If you have not yet installed Visual Studio then it is recommended to install Visual Studio 2017. If you have an existing installation of Visual Studio 2015 then it is possible to run both versions side by side.

Alternatively, you can configure Visual Studio 2015 to work with Azure. Visual Studio 2015 does not have the same concept of workloads as Visual Studio 2017. Instead you need to add the most recent version of the Microsoft Azure SDK for .NET.

* Install the [Microsoft Web Platform Installer 5.0](https://www.microsoft.com/web/downloads/platform.aspx) (Web PI)
* Run the Web PI and click on Products at the top
* Add the Microsoft Azure SDK for .NET (VS 2015)

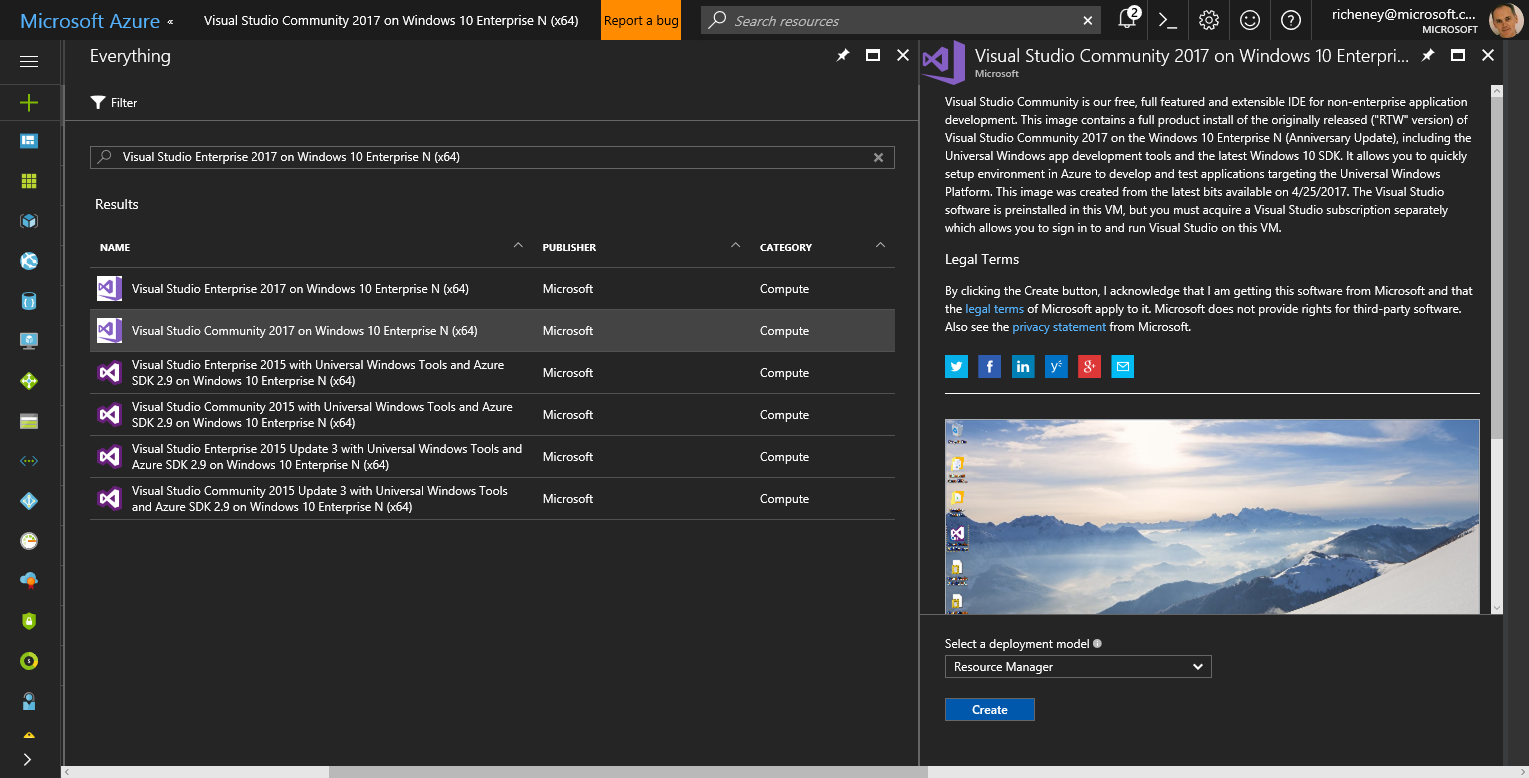


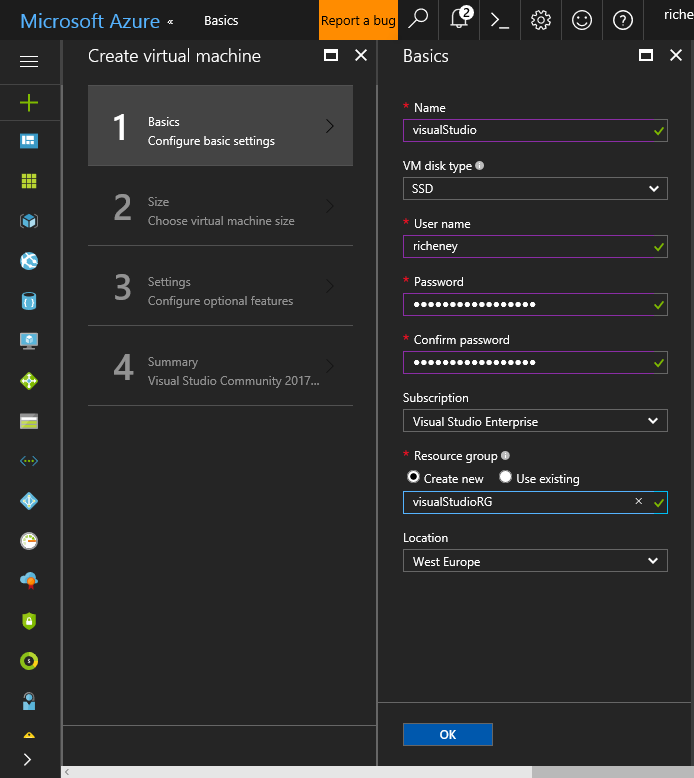
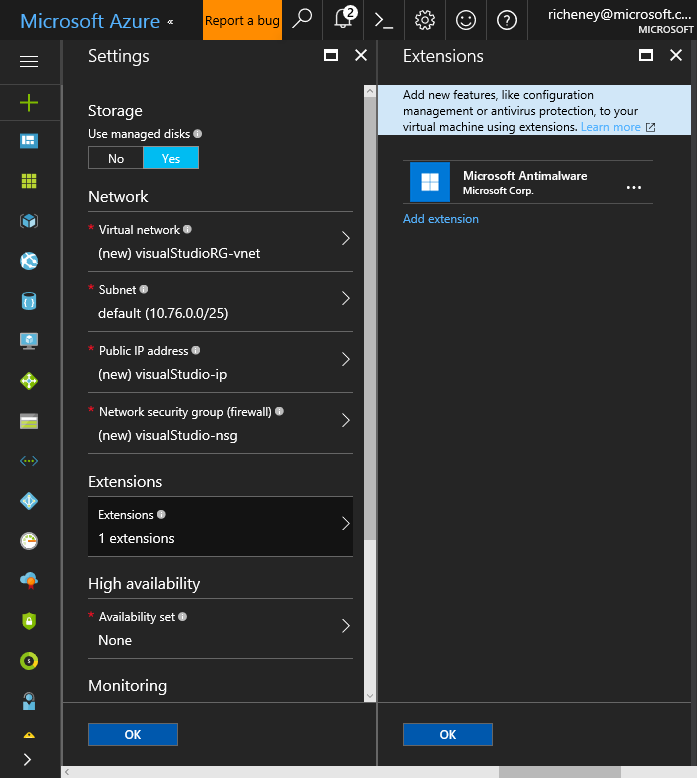
* [Configure Visual Studio for ARM Template Authoring](#_Configure_Visual_Studio) after the Azure SDK has installed

# Creating a Visual Studio VM in Azure

As an alternative to installing Visual Studio on to your own machine, you can spin up a Windows 10 virtual machine in your Azure subscription. The Azure Marketplace includes prebuilt images that include both the OS and the installed Visual Studio application.

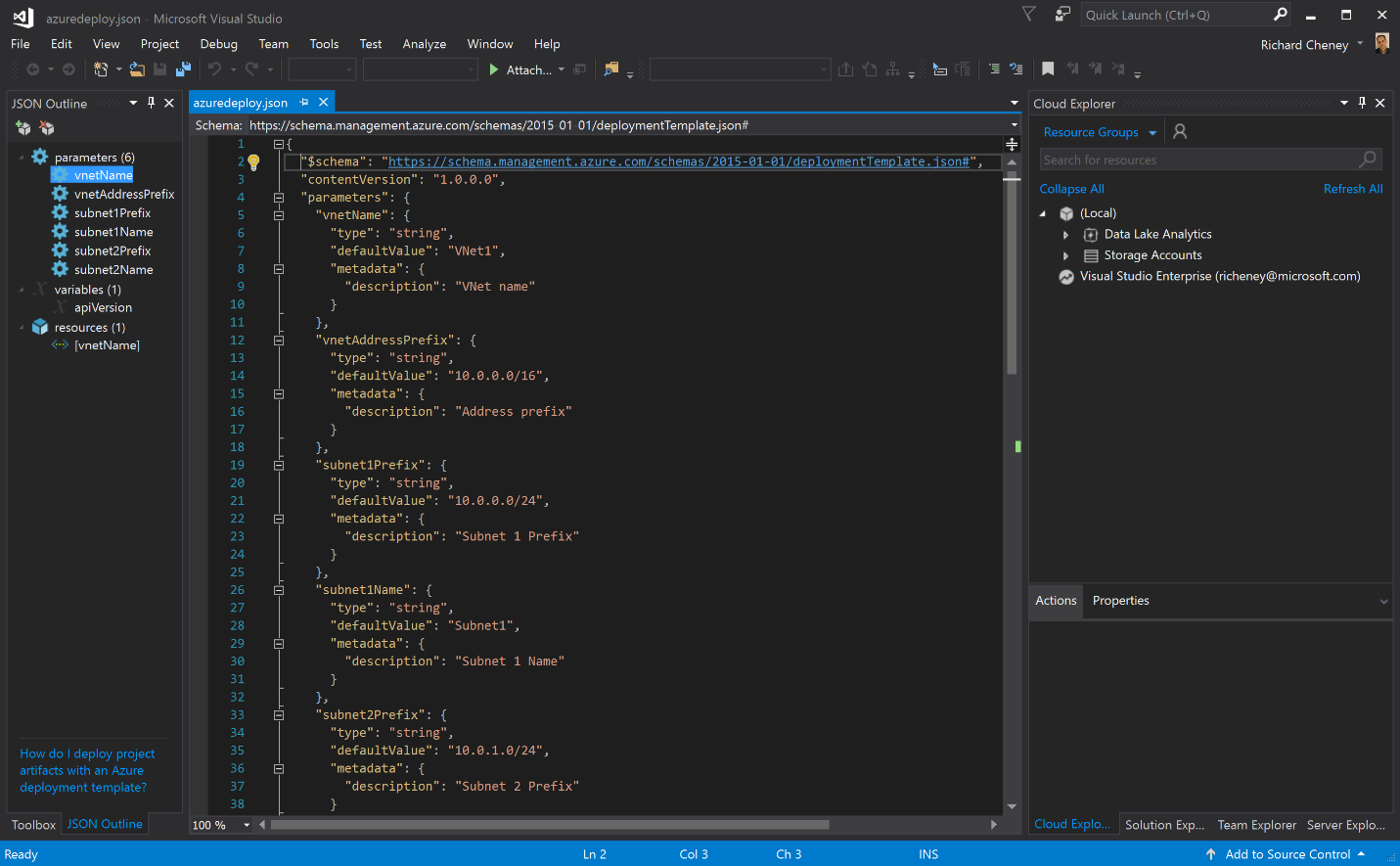
* Open the [portal](http://portal.azure.com) and click on New
* Type “Visual Studio” to filter the Marketplace offerings
* Select the Visual Studio Community 2017 on Windows 10 Enterprise image and click on Create



* Basic blade:
  + Set the name of your VM e.g. visualStudio
  + Keep the default VM disk type, SSD
  + Set you username and a valid password for the VM
  + Create a new resource group, e.g. visualStudioRG
  + Set the location to West Europe
* Size blade:
  + Select the DS2\_v2 VM size from the recommended list
* Settings blade:
  + Set the disk to managed
  + Add Antimalware as an extension to the VM, using the defaults
  + If required, change the address space and subnet range for the virtual network and subnet
  + Ensure that the VM has both a public IP (PIP) and network security group (NSG), as per the defaults
* Summary blade:
  + Review the summary and then create
  + The portal will revert to the dashboard, with a pinned tile showing the deployment
  + Once complete, the portal will switch to the virtual machine blade
* Virtual Machine blade:
  + Click on the Connect icon to download an RDP file
  + Remote Desktop will start and connect to the public IP
  + Open the Visual Studio shortcut on the desktop
* [Configure Visual Studio for ARM Template Authoring](#_Configure_Visual_Studio)

# Configure Visual Studio for ARM Template Authoring

* Click **View** 🡪 **Other Windows** 🡪 **JSON Outline** in order to set the left hand pane to show the JSON Outline
* Click **View** 🡪 **Cloud Explorer** to configure the Cloud Explorer in the right hand pane
* If required, authenticate your Azure subscription
* The authentication may be refreshed by clicking on the user icon in the Cloud Explorer pane
  + This dialog also allows subscriptions to be selected and filtered
* Verify the configuration by opening (**File** 🡪 **Open** 🡪 **File**, or CTRL+O) an example JSON file, e.g. https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-vnet-two-subnets/azuredeploy.json
* The screen should like similar to the one below, with the coloured JSON file in the middle pane, and the parameters, variables and resources listed in the JSON Outline pane on the left



* If the JSON Outline is not showing correctly, then try closing the file and reopening it.