Azure Resource Manager

Hands-on lab step-by-step

December 2017

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2017 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

[Azure Resource Manager hands-on lab step-by-step 1](#_Toc506902620)

[Abstract and learning objectives 1](#_Toc506902621)

[Overview 1](#_Toc506902622)

[Requirements 1](#_Toc506902623)

[Exercise 1 : Install Visual Studio Code 2](#_Toc506902624)

[Task 1: RDP to the Management VM 2](#_Toc506902625)

[Task 2: Install Visual Studio Code 4](#_Toc506902626)

[Task 3: Install Git 4](#_Toc506902627)

[Exercise 2: Install JSON Extension 6](#_Toc506902628)

[Task1: Install 6](#_Toc506902629)

[Exercise 3: Clone the GC Blueprint repository 6](#_Toc506902630)

[Task 1: Clone the git repository 6](#_Toc506902631)

[Exercise 4: Update the ARM Template 8](#_Toc506902632)

[Task 1: Restrict traffic to the virtual machine 8](#_Toc506902633)

[Task 2: Change deployment location 8](#_Toc506902634)

[Exercise 5: Deploy the ARM Template (optional) 9](#_Toc506902635)

[Task 1: Install the Azure Resource Manager PowerShell Modules 9](#_Toc506902636)

[Task 2: Modify deploy.ps1 file with your subscription 9](#_Toc506902637)

Azure Resource Manager hands-on lab step-by-step

## Abstract and learning objectives

In this lab, attendees will learn how to setup an environment to author and manage an Azure Resource Manager (ARM) template that can be used to deploy infrastructure such as virtual machine, storage, and networking.

## Overview

Contoso has asked you to define an Azure Resource Manager (ARM) template that can deploy their application CloudShop and its associated database using Azure Virtual Machines. You just got a new Windows 10 laptop and you’d like to setup the tools necessary to author the templates and deploy them to Azure.

## Requirements

1. Windows 10 PC or VM. In this lab, we will use Windows Server 2016 Azure VMs
2. Administrator privileges in order to install software

## Exercise 1 : Install Visual Studio Code

Duration: 20 minutes

From your laptop, open a browser to <https://portal.azure.com> (we recommend you open an InPrivate browser window (Ctrl+Shift+P) to ensure no conflicts happen with existing Azure credentials you may have. If you are using Google Chrome, this is referred to as an Ingonito (CTRL+Shift+N) browser window.

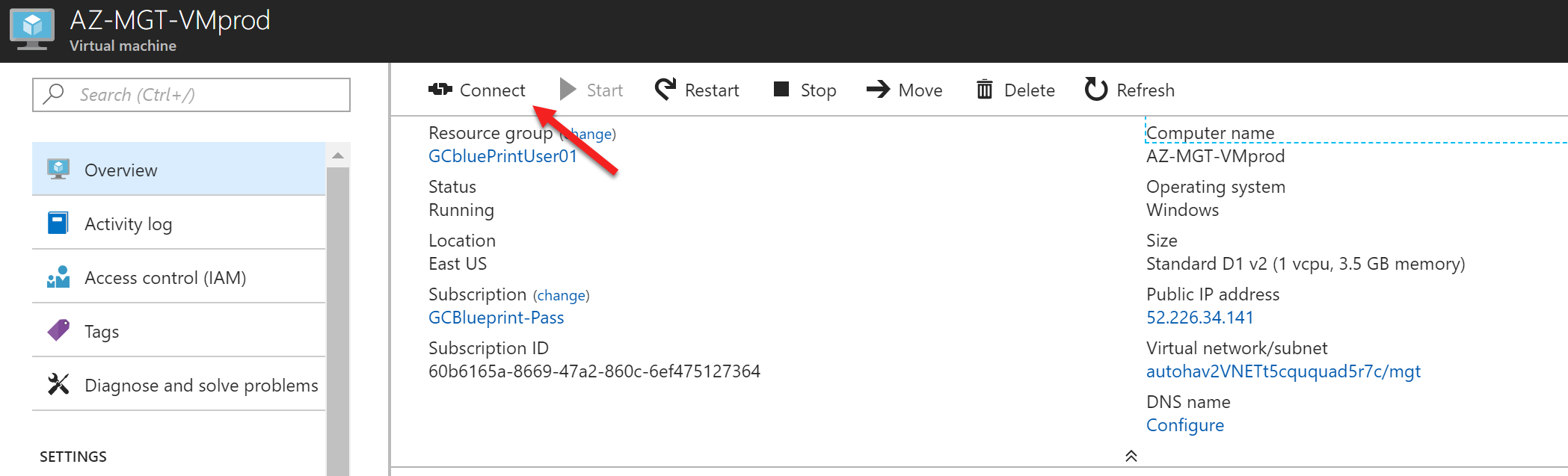
You login credentials are:

Username: adminuser0**?**@apottawaoutlook.onmicrosoft.com **(note: replace ? with your username)**

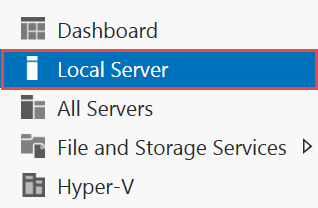
Password: **Azur3R0cks**

#### Task 1: RDP to the Management VM

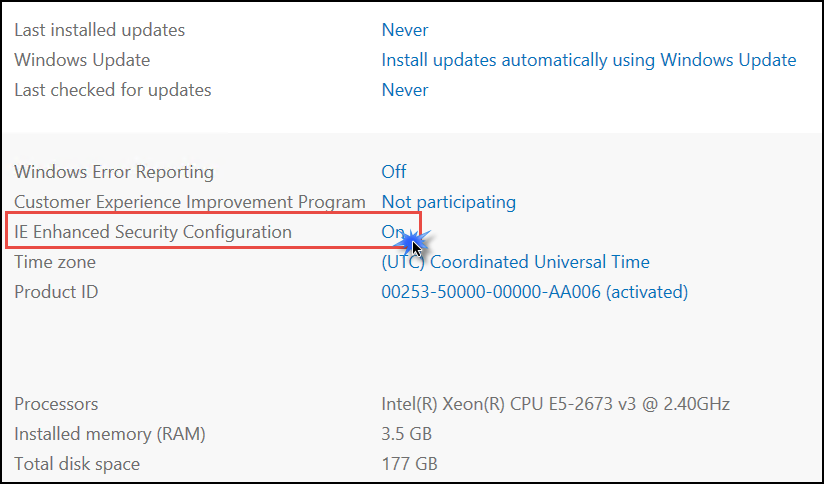
1. Launch a browser using Ingonito or InPrivate mode, and navigate to <https://portal.azure.com>. Once prompted, login with your Microsoft Azure credentials.
2. From the shortcuts blade on the left, click on **Resource Groups**. You should see the **GCBlueprintUser0? (again, replace ? with your username)** resource group.
3. Click on the **GCBlueprintUser01** resource group to see all the resources within that resource group.
4. From the list of resources, scroll down and find the **AZ-MGT-VMprod** VM and click it. You will be presented the **Overview** blade of the VM.
5. Click on the **Connect** button shown here. This will prompt you save an RDP connection file which has the public IP address of this VM. Save the file locally.



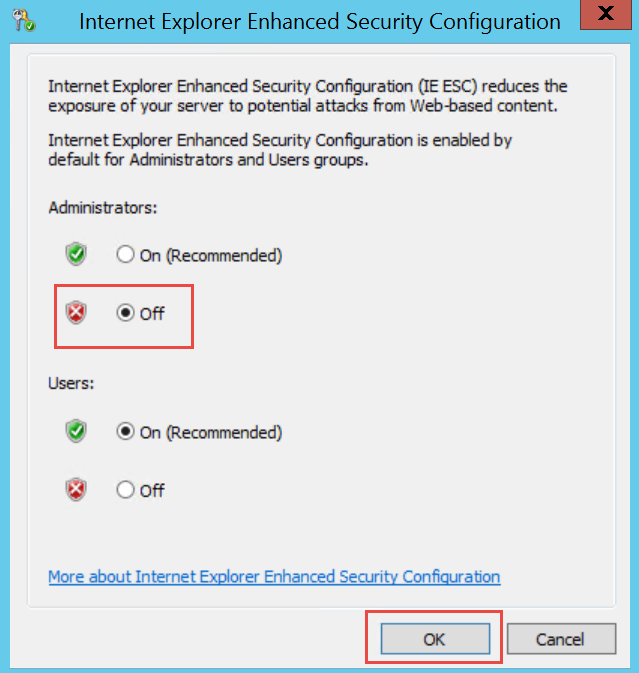
1. From your laptop, open an RDP session using the file you just downloaded by double-clicking it.
2. Use the following credentials to login:
   1. Username: **adminuser01@gcblueprint.local**
   2. Password: **P@ssw0rd1234567**
3. Once you logged in, notice the Server Manager opens by default. On the left, click **Local Server**.



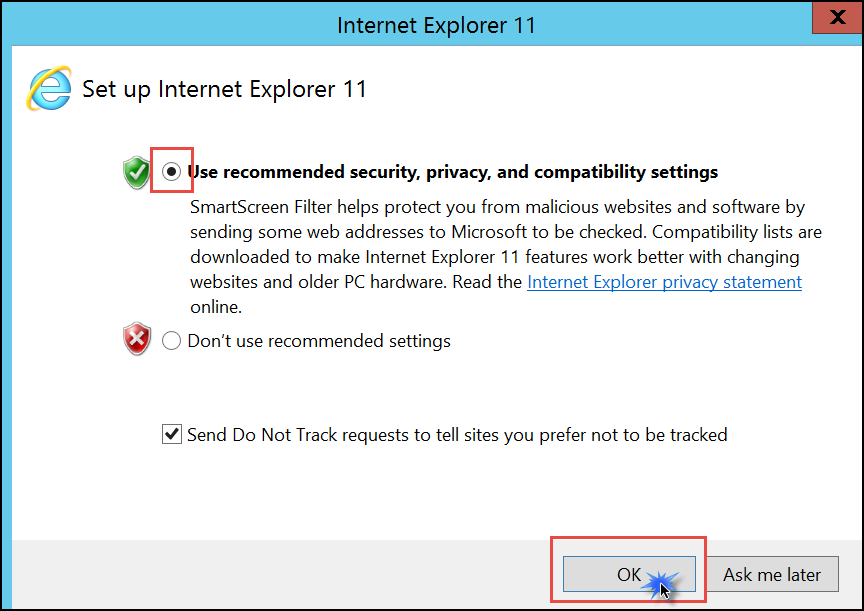
1. On the right side of the pane, click On by IE Enhanced Security Configuration.



1. Change to Off for Administrators, and click OK.



1. In the lower left corner, click Internet Explorer to open it. On first use, you will be prompted about security settings. Accept the defaults by clicking OK.



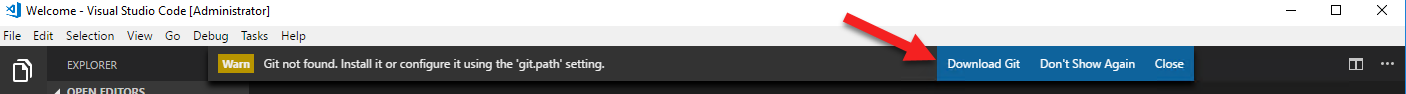
1. If prompted, click Don’t show this again regarding protected mode.
2. To download Visual Studio Code, open the URL <https://code.visualstudio.com/Download> and choose the version for Windows.
3. Download Visual Studio Code and save it locally. We will install it in the next task

#### Task 2: Install Visual Studio Code

1. Double click the installation file you download in the previous task. As of this writing, the file name is **VSCodeSetup-x64-1.20.1.exe**.
2. Once the **Setup – Visual Studio Code** window launches, click **Next**.
3. From the **License Agreement** window, click the **I accept the agreement** radio button and click **Next**.
4. From the **Select Destination Location** window, click **Next**.
5. From the **Select Start Menu Folder** window, click **Next**.
6. From the **Select Additional Tasks** window, select the **Register Code as an editor for supported file types** and click **Next**.
7. From the **Ready to Install** window, click **Install**.
8. From the **Completing the Visual Studio Code Setup Wizard**, click **Finish**.

#### Task 3: Install Git

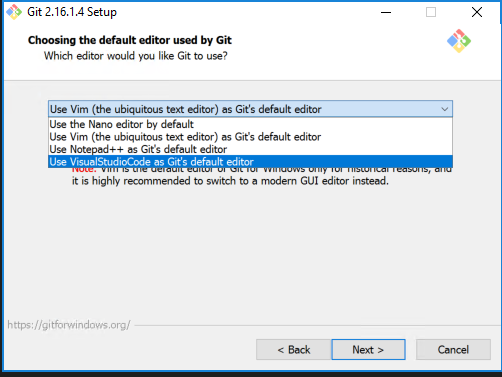
1. With Visual Studio Code running for the fist time click on the **Download Git** link at the top of the screen. This will open the browser window to the git download page.



1. With Internet Explorer open to the git website, click on the download link. Save the link when prompted.



1. Launch the git installer by double-clicking on the file you just downloaded.
2. From the **Information** window, click **Next**.
3. From the **Select Destination Location**, click **Next**.
4. From the **Select Components**, click **Next**.
5. From the **Select Start Menu Folder**, click **Next**.
6. From the **Choosing the default editor used by Git**, select **Use VisualStudio Code as Git’s default editor** and click **Next.**



1. From the **Adjusting your PATH environment** window, click **Next**.
2. From the **Choosing HTTPS transport backend** window, click **Next**.
3. From the **Configuring the line ending conversions** window, click **Next**.
4. From the **Configuring the terminal emulator to use with Git Bash** window, click **Next**.
5. From the **Configuring extra options** window, click **Next**.
6. From the **Completing the Git Setup Wizard** window, uncheck **View Release Notes** and click **Finish**.
7. If Visual Studio Code is open close it and re-open so the installation of Git takes effect.

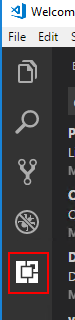
## Exercise 2: Install JSON Extension

Duration: 5 minutes

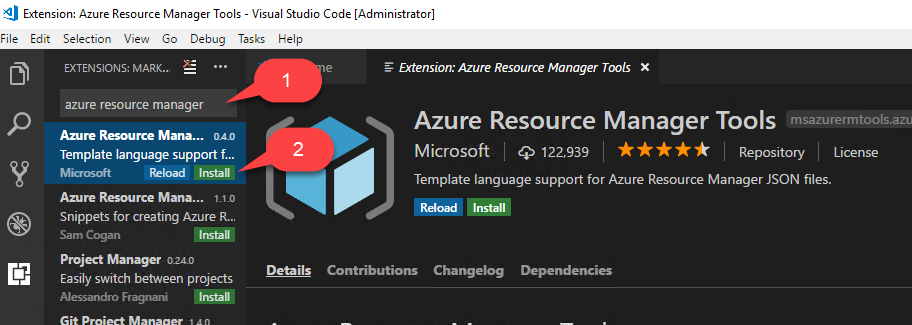
In this exercise, you will install a Visual Studio Code extension which helps you to create and manage JSON files.

#### Task1: Install

1. With Visual Studio Code open, click on the Extensions icon on the left or press the CTRL+Shift+X keyboard shortcut.



1. From the search text box, type **Azure Resource Manager Tools** and click **Install**



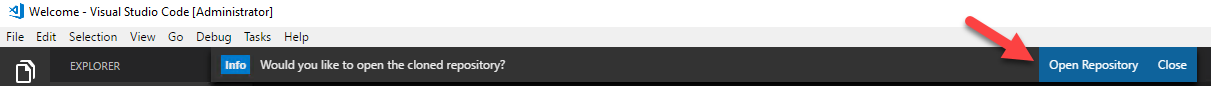
## Exercise 3: Clone the GC Blueprint repository

Duration: 15 minutes

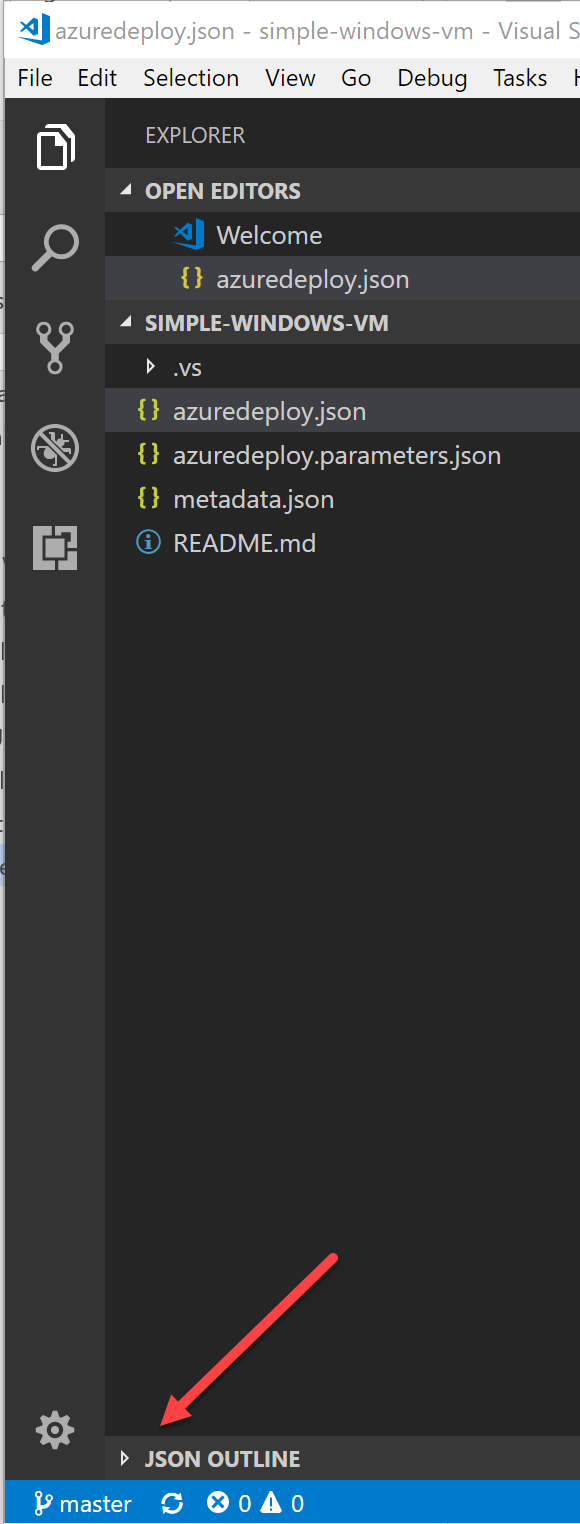
In this exercise, you will clone a git repository and edit some of the content locally on your VM. You will clone a simple repository containing the ARM templates for deploying a simple Windows Server VM.

#### Task 1: Clone the git repository

1. Open Visual Studio Code.
2. From the **View** menu, select **Command Palette…**
3. From the Command Palette, type **Git:Clone**. This will prompt you to enter the repository URL of <https://github.com/aqasrawi/simple-windows-vm>
4. When prompted for a local folder, accept the default folder (ex. C:\Users\%username%)
5. Once the repository has been cloned, you will a message to prompt you to open. Click **Open Repository**.



1. Open the **azuredeploy.json** by double-clicking it.
2. Notice that the JSON OUTLINE feature appears in the Explorer view. Expand it.



1. Expand the **resources** element in the JSON.

## Exercise 4: Update the ARM Template

Duration: 20 minutes

In this exercise, you will deploy a network security group to restrict the network attack surface for the deployment.

#### Task 1: Restrict traffic to the virtual machine

1. With the **azuredeploy.json** file open, Add the following at the beginning of the JSON template as the first item under the **resources** node. This will deploy the network security group resource and add a rule. Therefore, the only port open on the Public IP is port 3389.

{

"name": "[parameters('networkSecurityGroupName')]",

"type": "Microsoft.Network/networkSecurityGroups",

"apiVersion": "2017-06-01",

"location": "[parameters('location')]",

"properties": {

"securityRules": [

{

"name": "default-allow-rdp",

"properties": {

"priority": 1000,

"protocol": "Tcp",

"access": "Allow",

"direction": "Inbound",

"sourceAddressPrefix": "\*",

"sourcePortRange": "\*",

"destinationAddressPrefix": "\*",

"destinationPortRange": "3389"

}

}

]

}

},

1. Now when the VM gets deployed, the VM will have an NSG rule to limit inbound traffic flow.

#### Task 2: Change deployment location

1. With the **azuredeploy.parameters.json** file open. You will change the **location** parameter to have the VM deploy in the Canada East region instead of East US.
2. Find the **location**  attribute. Replace the **eastus** value with **canadacentral**. Save the file.

"parameters": {

"location": {

"value": "eastus"

},

## Exercise 5: Deploy the ARM Template (optional)

Duration: 60 minutes

In this exercise, you will deploy the ARM template solution to Azure.

#### Task 1: Install the Azure Resource Manager PowerShell Modules

1. In order to deploy the ARM template, you will need to install the AzureRM Powershell module. To do so, open a PowerShell command prompt and type the following command: Install-Module -Name AzureRM
2. If prompted to install NuGet Provider, type Yes
3. If prompted to accept the package from an untrusted repository, type Yes

This will take several minutes while the AzureRM Modules get installed on your system.

#### Task 2: Modify deploy.ps1 file with your subscription

To deploy the solution, you need to execute the deploy.ps1 script from a PowerShell window. This script requires the input of an Azure Subscription ID where the VM will get deployed. Open the deploy.ps1 file with Visual Studio code and replace SUBID with your Subscription’s ID:

$subscriptionId = 'SUBID'

You can find your subscription ID from the Azure Portal under Subscriptions.