Microsoft®

Building Azure resources with Azure Resource Manager json template

Microsoft Azure

Hands-on lab

Building Azure resources with json template

Resource Groups allow you to manage all your resources in an application together. Azure Resource Manager allows you to manage these multiple resources together as a single resource group. In this hands on lab we will take you through how to build a json template and find all the providers that are available and how to register new providers. We will also take you through using a template to deploy a resource group. Finally we will take you through building out your own environment leveraging Visual Studio and the Azure SDK and how you can deploy your lab through PowerShell, Visual Studio and GitHub

Contents

Lab: Deploy Resources to Azure using Azure Resource Manager (ARM)

✏ In this lab, we will be creating some items that are global in nature, e.g. Storage Accounts. For this reason, we need to make sure everyone is using unique names for their resources. In the activities below, we have <yourinitials> in many places. Replace <yourinitials> with your actual initials, e.g 'vkp'. You can also use another random word in its place or a series of numbers and characters so long as it results in a globally unique name. If you try to create an item that is not unique, the deployment may fail and you will have to change the value to continue the exercise.

Pre-Requisites

The pre-requisites for this lab are already available on the VM. Below is a list of the pre-requisites and links to them in case they need to be reinstalled or you want to try this on another device.

- Azure PowerShell 1.0
- Azure SDK v2.9.1 for VS 2015

  Azure Account (Provided to you)

Azure Resource Manager introduces an entirely new way of thinking about your Azure resources. Instead of creating and managing individual resources, you begin by imagining an entire solution, such as a blog, a photo gallery, a SharePoint portal, or a wiki. You use a template -- a declarative representation of the solution -- to create a resource group that contains all of the resources you need to support the solution. Then, you manage and deploy that resource group as a logical unit

Exercise 1: View Azure Resource Providers

Introduction

Prior to Azure Resource Manager (ARM) being available for all to use, it required you to register new providers in order to use them. Since ARM is available now, when you leverage a new component such as a Virtual Machine v2, Storage Account v2, Virtual Network v2, these resource providers will be registered automatically. We still want to show you how to find these.

Objectives

After completing this lab, you will be able to:

- Connect to Azure
- Query available Resource Providers
- Register a Resource Provider

Estimated time to complete this exercise

10 minutes

Scenario

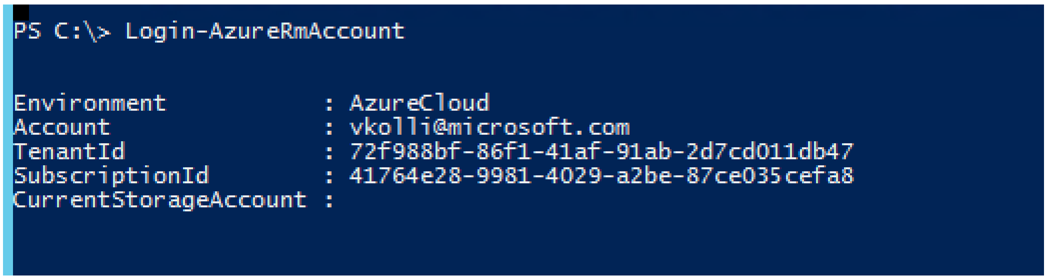
View and Register Azure resource providers.

Task 1: Sign in to Azure and list available ARM providers

In order to create a Storage Account, we must first sign into an Azure Account.

1. Click on the Start Menu and then type PowerShell ISE, right-click on the PowerShell ISE and select Run as Administrator.
2. You may get prompted if you want to allow the program to make changes, if you do click "Yes"
3. In the PowerShell window, type the following command:

```
↪     Login-AzureRmAccount
```

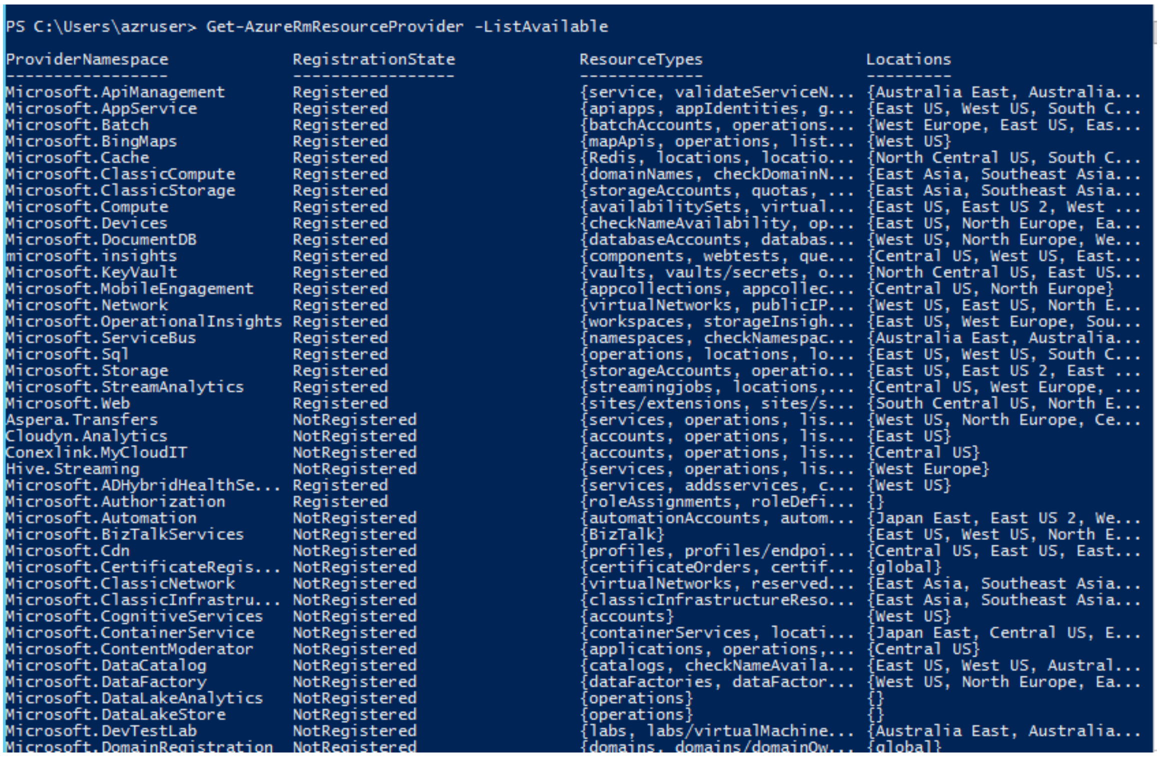


```
PS C:\> Login-AzureRmAccount


Environment           : AzureCloud
Account               : vkolli@microsoft.com
TenantId              : 72f988bf-86f1-41af-91ab-2d7cd011db47
SubscriptionId        : 41764e28-9981-4029-a2be-87ce035cefa8
CurrentStorageAccount :
```

4. The Sign in to Windows Azure PowerShell dialog box will open.
5. Enter your account and click “Continue”.
6. On the next page, enter your password or certificate and click “Ok”.
7. In the PowerShell window, type the following command:

```
↪ Get-AzureRmResourceProvider -ListAvailable
```

8. You will see the following results



```
PS C:\Users\azruser> Get-AzureRmResourceProvider -ListAvailable

ProviderNamespace             RegistrationState             ResourceTypes                 Locations
-----------------             -----------------             -------------                 ---------
Microsoft.ApiManagement       Registered                    {service, validateServiceN... {Australia East, Australia...
Microsoft.AppService          Registered                    {apiapps, appIdentities, g... {East US, West US, South C...
Microsoft.Batch               Registered                    {batchAccounts, operations... {West Europe, East US, Eas...
Microsoft.BingMaps            Registered                    {mapApis, operations, list... {West US}
Microsoft.Cache               Registered                    {Redis, locations, locatio... {North Central US, South C...
Microsoft.ClassicCompute      Registered                    {domainNames, checkDomainN... {East Asia, Southeast Asia...
Microsoft.ClassicStorage      Registered                    {storageAccounts, quotas, ... {East Asia, Southeast Asia...
Microsoft.Compute             Registered                    {availabilitySets, virtual... {East US, East US 2, West ...
Microsoft.Devices             Registered                    {checkNameAvailability, op... {East US, North Europe, Ea...
Microsoft.DocumentDB          Registered                    {databaseAccounts, databas... {West US, North Europe, We...
microsoft.insights            Registered                    {components, webtests, que... {Central US, West US, East...
Microsoft.KeyVault            Registered                    {vaults, vaults/secrets, o... {North Central US, East US...
Microsoft.MobileEngagement    Registered                    {appcollections, appcollec... {Central US, North Europe}
Microsoft.Network             Registered                    {virtualNetworks, publicIP... {West US, East US, North E...
Microsoft.OperationalInsights Registered                    {workspaces, storageInsigh... {East US, West Europe, Sou...
Microsoft.ServiceBus          Registered                    {namespaces, checkNamespac... {Australia East, Australia...
Microsoft.Sql                 Registered                    {operations, locations, lo... {East US, West US, South C...
Microsoft.Storage             Registered                    {storageAccounts, operatio... {East US, East US 2, East ...
Microsoft.StreamAnalytics     Registered                    {streamingjobs, locations,... {Central US, West Europe, ...
Microsoft.Web                 Registered                    {sites/extensions, sites/s... {South Central US, North E...
Aspera.Transfers              NotRegistered                 {services, operations, lis... {West US, North Europe, Ce...
Cloudyn.Analytics             NotRegistered                 {accounts, operations, lis... {East US}
Conexlink.MyCloudIT           NotRegistered                 {accounts, operations, lis... {Central US}
Hive.Streaming                NotRegistered                 {services, operations, lis... {West Europe}
Microsoft.ADHybridHealthSe... Registered                    {services, addsservices, c... {West US}
Microsoft.Authorization       Registered                    {roleAssignments, roleDefi... {}
Microsoft.Automation          NotRegistered                 {automationAccounts, autom... {Japan East, East US 2, We...
Microsoft.BizTalkServices     NotRegistered                 {BizTalk}                     {East US, West US, North E...
Microsoft.Cdn                 NotRegistered                 {profiles, profiles/endpoi... {Central US, East US, East...
Microsoft.CertificateRegis... NotRegistered                 {certificateOrders, certif... {global}
Microsoft.ClassicNetwork      NotRegistered                 {virtualNetworks, reserved... {East Asia, Southeast Asia...
Microsoft.ClassicInfrastru... NotRegistered                 {classicInfrastructureReso... {East Asia, Southeast Asia...
Microsoft.CognitiveServices   NotRegistered                 {accounts}                    {West US}
Microsoft.ContainerService    NotRegistered                 {containerServices, locati... {Japan East, Central US, E...
Microsoft.ContentModerator    NotRegistered                 {applications, operations,... {Central US}
Microsoft.DataCatalog         NotRegistered                 {catalogs, checkNameAvaila... {East US, West US, Austral...
Microsoft.DataFactory         NotRegistered                 {dataFactories, dataFactor... {West US, North Europe, Ea...
Microsoft.DataLakeAnalytics   NotRegistered                 {operations}                  {}
Microsoft.DataLakeStore       NotRegistered                 {operations}                  {}
Microsoft.DevTestLab          NotRegistered                 {labs, labs/virtualMachine... {Australia East, Australia...
Microsoft.DomainRegistration  NotRegistered                 {domains, domains/domainOw... {global}
```

Task 2: Register provider (Optional Step)

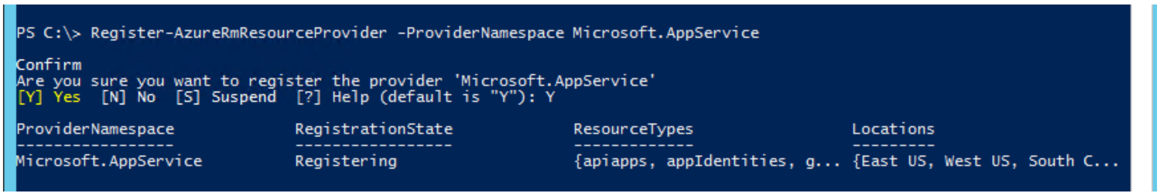
.

This Step is optional you can skip it and go to Exercise 2

Although it is not required at this point. If you needed to register a provider that was in preview you could register a provider using the following command.

```
↪ Register-AzureRmResourceProvider -ProviderNamespace
Microsoft.AppService
```

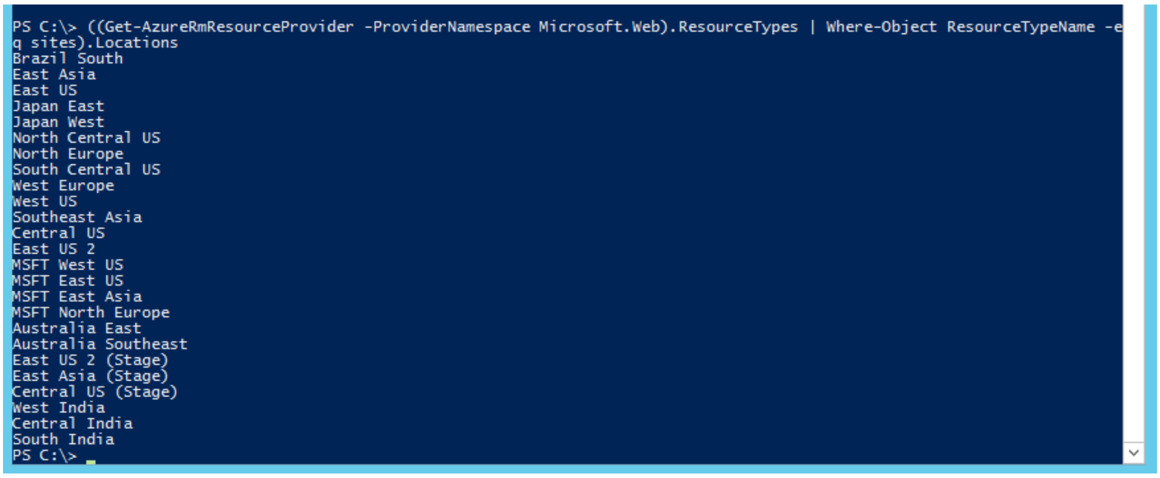
1. You will see the following results



```
PS C:\> Register-AzureRmResourceProvider -ProviderNamespace Microsoft.AppService

Confirm
Are you sure you want to register the provider 'Microsoft.AppService'
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y

ProviderNamespace            RegistrationState            ResourceTypes                Locations
-----------------            -----------------            -------------                ---------
Microsoft.AppService         Registering                  {apiapps, appIdentities, g... {East US, West US, South C...
```

2. To limit your output to the supported locations for a specific type of of resource, such as web sites, use:

```
↪ ((Get-AzureRmResourceProvider -ProviderNamespace
Microsoft.Web).ResourceTypes | Where-Object ResourceTypeName -eq
sites).Locations
```



```
PS C:\> ((Get-AzureRmResourceProvider -ProviderNamespace Microsoft.Web).ResourceTypes | Where-Object ResourceTypeName -e
q sites).Locations
Brazil South
East Asia
East US
Japan East
Japan West
North Central US
North Europe
South Central US
West Europe
West US
Southeast Asia
Central US
East US 2
MSFT West US
MSFT East US
MSFT East Asia
MSFT North Europe
Australia East
Australia Southeast
East US 2 (Stage)
East Asia (Stage)
Central US (Stage)
West India
Central India
South India
PS C:\>
```

Exercise 2: Create a Storage Account

Introduction

In order to complete some of the following exercises, a Storage Account is required. In this exercise, we will create the Storage Account to be used for the future exercises.

Objectives

After completing this lab, you will be able to:

- Create a Storage Account in Azure

Estimated time to complete this exercise

3 minutes

Scenario

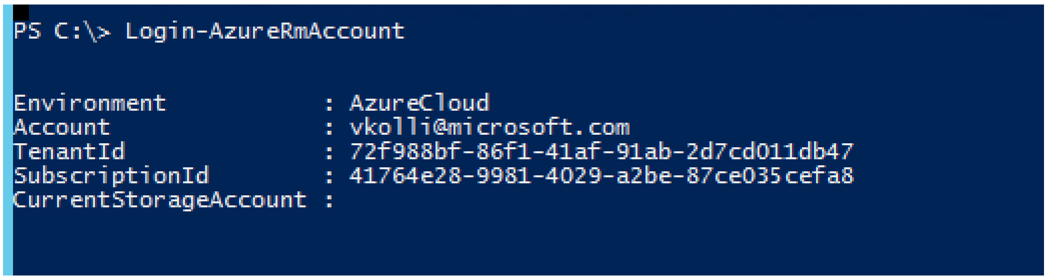
Create Storage account.

Task 1: Sign in to Azure

In order to create a Storage Account, we must first sign into an Azure Account.

1. Launch Azure PowerShell from the icon on the Desktop if you don't already have it open.
2. You may get prompted if you want to allow the program to make changes, if you do click "Yes"
3. In the PowerShell window, type the following command:

```
↪ Login-AzureRmAccount
```

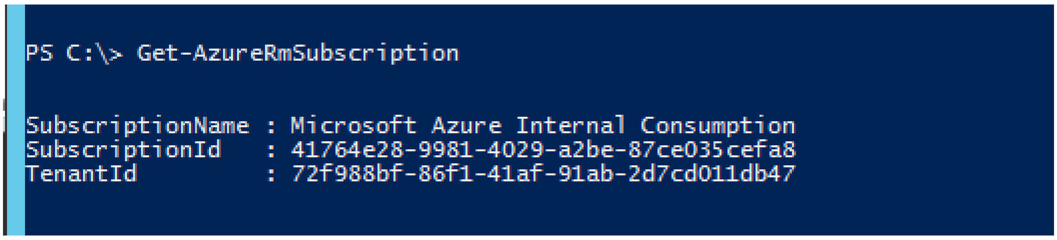


```
PS C:\> Login-AzureRmAccount


Environment           : AzureCloud
Account               : vkolli@microsoft.com
TenantId              : 72f988bf-86f1-41af-91ab-2d7cd011db47
SubscriptionId        : 41764e28-9981-4029-a2be-87ce035cefa8
CurrentStorageAccount :
```

4. The **Sign in to Windows Azure PowerShell** dialog box will open.
5. Enter your account and click "Continue".
6. On the next page, enter your password or certificate and click "Ok".
7. In the PowerShell window, type the following command:

```
↪     Get-AzureRmSubscription
```

8. Your results should look similar to the results below:

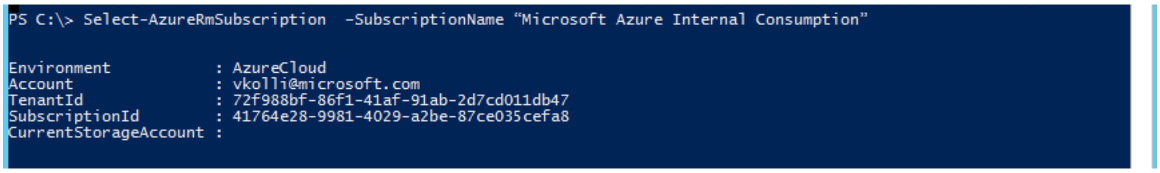


```
PS C:\> Get-AzureRmSubscription


SubscriptionName : Microsoft Azure Internal Consumption
SubscriptionId   : 41764e28-9981-4029-a2be-87ce035cefa8
TenantId         : 72f988bf-86f1-41af-91ab-2d7cd011db47
```

9. In the powershell wndow type the following command, take the subscription name as in above step

```
↪ Select-AzureRmSubscription  –SubscriptionName “Microsoft Azure Internal Consumption”
```

10. Your results should look similar to the results below:



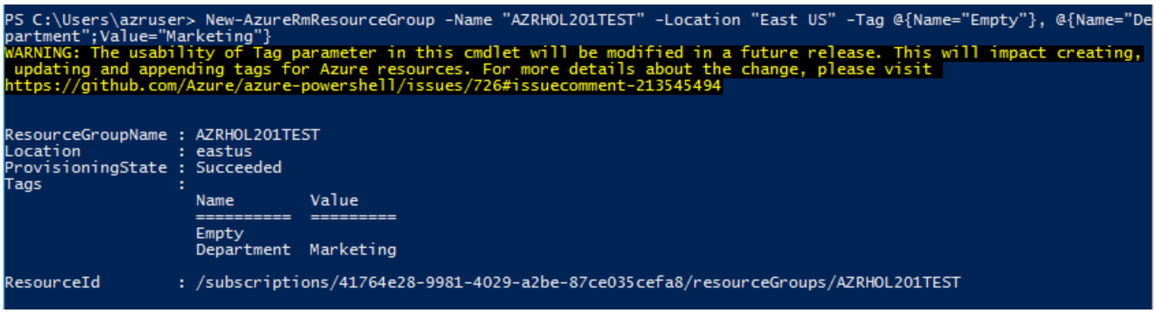
```
PS C:\> Select-AzureRmSubscription  -SubscriptionName “Microsoft Azure Internal Consumption”


Environment           : AzureCloud
Account               : vkolli@microsoft.com
TenantId              : 72f988bf-86f1-41af-91ab-2d7cd011db47
SubscriptionId        : 41764e28-9981-4029-a2be-87ce035cefa8
CurrentStorageAccount :
```

Task 2: Create Storage Account

Now that we have our subscription selected, it is time to create our Storage Account. This Storage Account will be used to store temporary artifacts needed to deploy our Website in a later exercise.

1) Create resource group

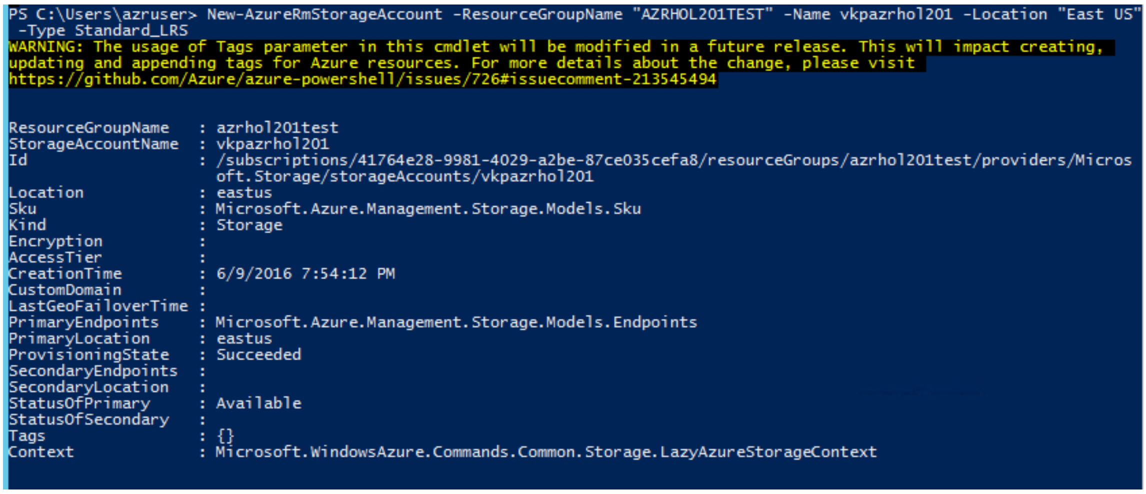
```
↪ New-AzureRmResourceGroup -Name "AZRHOL201TEST" -Location "East US" -Tag @{Name="Empty"}, @{Name="Department";Value="Marketing"}
```



```
PS C:\Users\azruser> New-AzureRmResourceGroup -Name "AZRHOL201TEST" -Location "East US" -Tag @{Name="Empty"}, @{Name="De
partment";Value="Marketing"}
WARNING: The usability of Tag parameter in this cmdlet will be modified in a future release. This will impact creating,
 updating and appending tags for Azure resources. For more details about the change, please visit
https://github.com/Azure/azure-powershell/issues/726#issuecomment-213545494


ResourceGroupName : AZRHOL201TEST
Location          : eastus
ProvisioningState : Succeeded
Tags              :
                    Name        Value
                    ==========  =========
                    Empty
                    Department  Marketing

ResourceId        : /subscriptions/41764e28-9981-4029-a2be-87ce035cefa8/resourceGroups/AZRHOL201TEST
```

2) In the PowerShell window, type the following command, storage account name has to be all lower case letters:

```
↪New-AzureRmStorageAccount -ResourceGroupName "AZRHOL201TEST" -Name
vkpazrhol201 -Location "East US" -Type Standard_LRS
```

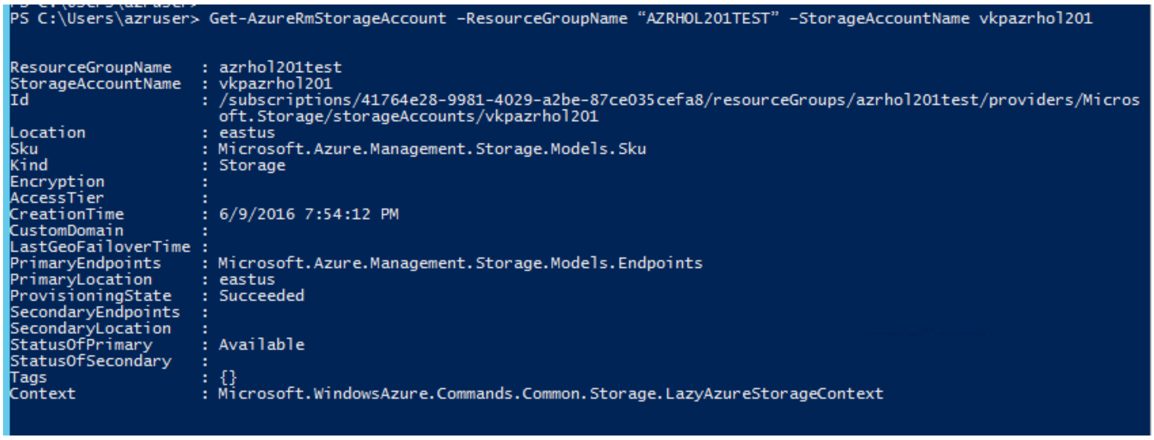


```
PS C:\Users\azruser> New-AzureRmStorageAccount -ResourceGroupName "AZRHOL201TEST" -Name vkpazrhol201 -Location "East US"
 -Type Standard_LRS
WARNING: The usage of Tags parameter in this cmdlet will be modified in a future release. This will impact creating,
updating and appending tags for Azure resources. For more details about the change, please visit
https://github.com/Azure/azure-powershell/issues/726#issuecomment-213545494


ResourceGroupName   : azrhol201test
StorageAccountName  : vkpazrhol201
Id                  : /subscriptions/41764e28-9981-4029-a2be-87ce035cefa8/resourceGroups/azrhol201test/providers/Micros
                      oft.Storage/storageAccounts/vkpazrhol201
Location            : eastus
Sku                 : Microsoft.Azure.Management.Storage.Models.Sku
Kind                : Storage
Encryption          :
AccessTier          :
CreationTime        : 6/9/2016 7:54:12 PM
CustomDomain        :
LastGeoFailoverTime :
PrimaryEndpoints    : Microsoft.Azure.Management.Storage.Models.Endpoints
PrimaryLocation     : eastus
ProvisioningState   : Succeeded
SecondaryEndpoints  :
SecondaryLocation   :
StatusOfPrimary     : Available
StatusOfSecondary   :
Tags                : {}
Context             : Microsoft.WindowsAzure.Commands.Common.Storage.LazyAzureStorageContext
```

3) Now that you have created a Storage Account, you are done with this exercise. If you want to verify it was successfully created, you can run the following PowerShell command:

```
Get-AzureRmStorageAccount –ResourceGroupName “AZRHOL201TEST” –
StorageAccountName <yourinitials>azrhol200
```

4) Your results should look similar to the results below:



```
PS C:\Users\azruser> Get-AzureRmStorageAccount -ResourceGroupName “AZRHOL201TEST” -StorageAccountName vkpazrhol201


ResourceGroupName   : azrhol201test
StorageAccountName  : vkpazrhol201
Id                  : /subscriptions/41764e28-9981-4029-a2be-87ce035cefa8/resourceGroups/azrhol201test/providers/Micros
                      oft.Storage/storageAccounts/vkpazrhol201
Location            : eastus
Sku                 : Microsoft.Azure.Management.Storage.Models.Sku
Kind                : Storage
Encryption          :
AccessTier          :
CreationTime        : 6/9/2016 7:54:12 PM
CustomDomain        :
LastGeoFailoverTime :
PrimaryEndpoints    : Microsoft.Azure.Management.Storage.Models.Endpoints
PrimaryLocation     : eastus
ProvisioningState   : Succeeded
SecondaryEndpoints  :
SecondaryLocation   :
StatusOfPrimary     : Available
StatusOfSecondary   :
Tags                : {}
Context             : Microsoft.WindowsAzure.Commands.Common.Storage.LazyAzureStorageContext
```

Exercise 3: Deploy VMs to Azure via Visual Studio

Introduction

In this exercise, we will be deploying Virtual Machines to Azure with a Resource Group.

Objectives

After completing this lab, you will be able to:

- Create Virtual Machines in Azure
- Have a better understanding of the updated Compute, Storage, and Networking concepts

Estimated time to complete this exercise

20 minutes

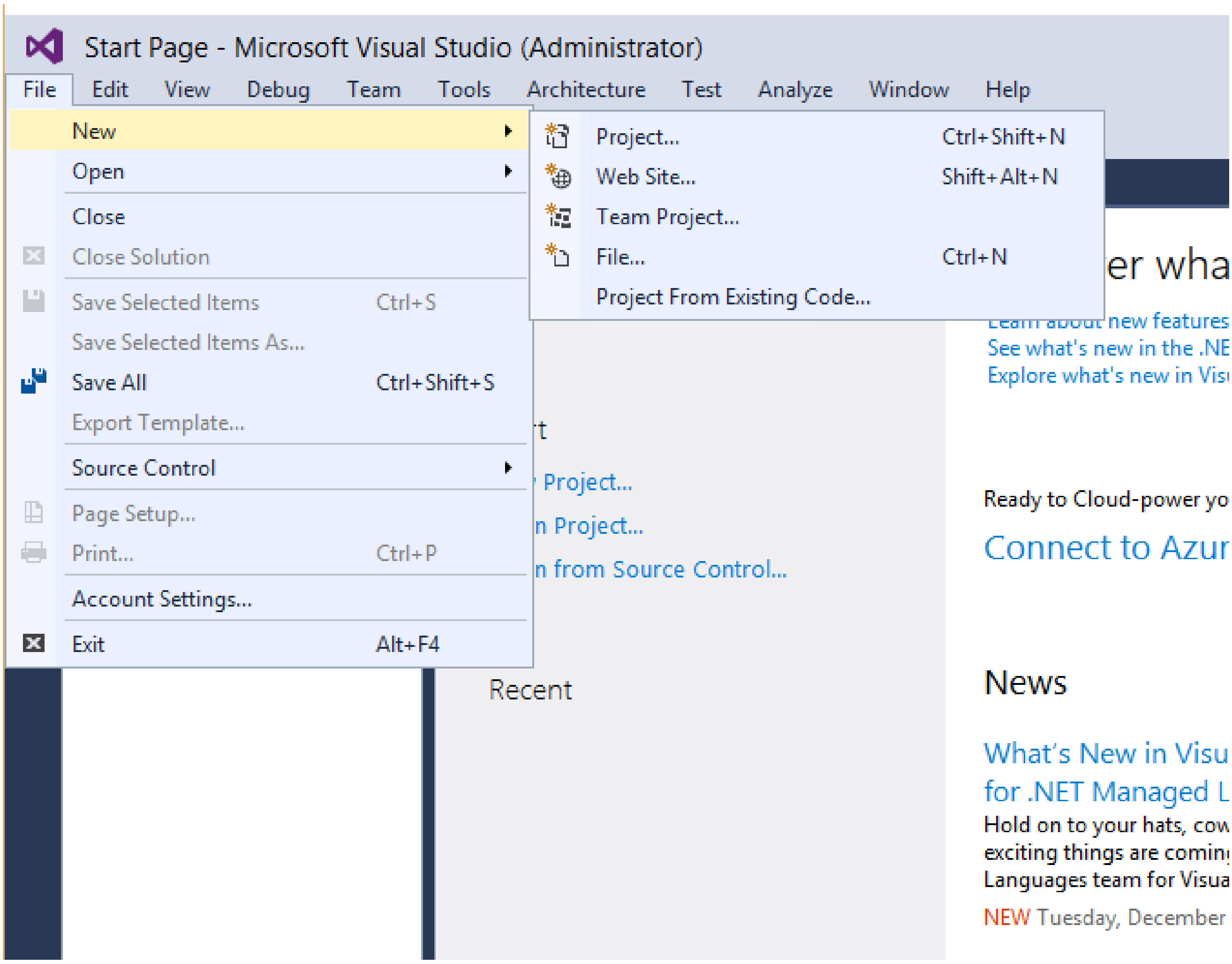
Scenario

Resources Groups and Azure Resource Manager help speed the deployment of Virtual Machines to Azure.

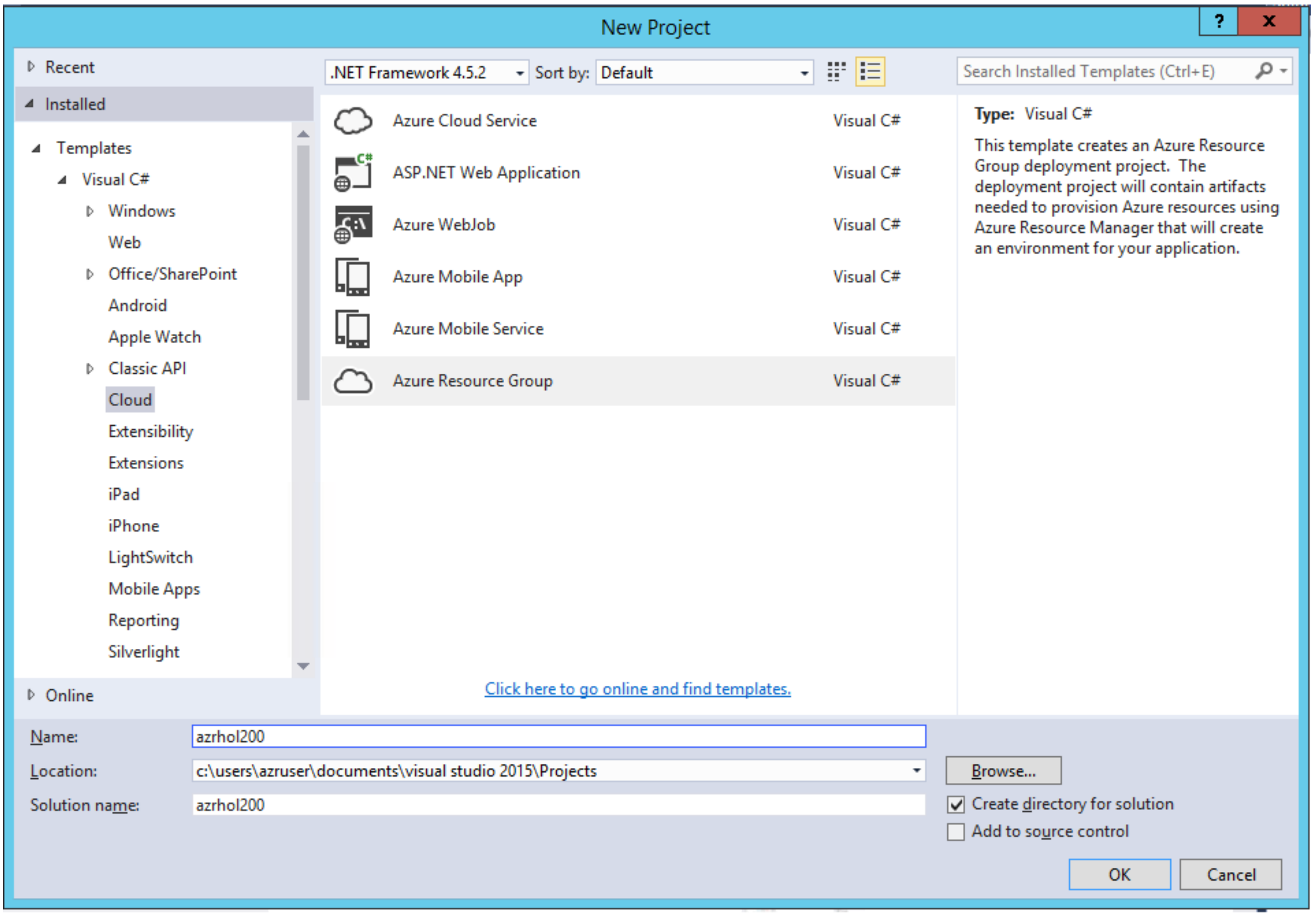
Task 1: Create a Project in Visual Studio

We will add a new project using **File, New, Project...**

1) Open Visual Studio using the icon in the taskbar.
2) Click New Project…



3) Give the project the name azr200vm, make sure **Azure Resource Group** is selected under Visual C# > **Cloud** and click "**Ok**".


New Project

.NET Framework 4.5.2 | Sort by: Default

Azure Cloud Service	Visual C#
ASP.NET Web Application	Visual C#
Azure WebJob	Visual C#
Azure Mobile App	Visual C#
Azure Mobile Service	Visual C#
Azure Resource Group	Visual C#

Type: Visual C#

This template creates an Azure Resource Group deployment project. The deployment project will contain artifacts needed to provision Azure resources using Azure Resource Manager that will create an environment for your application.

Name: azrhol200

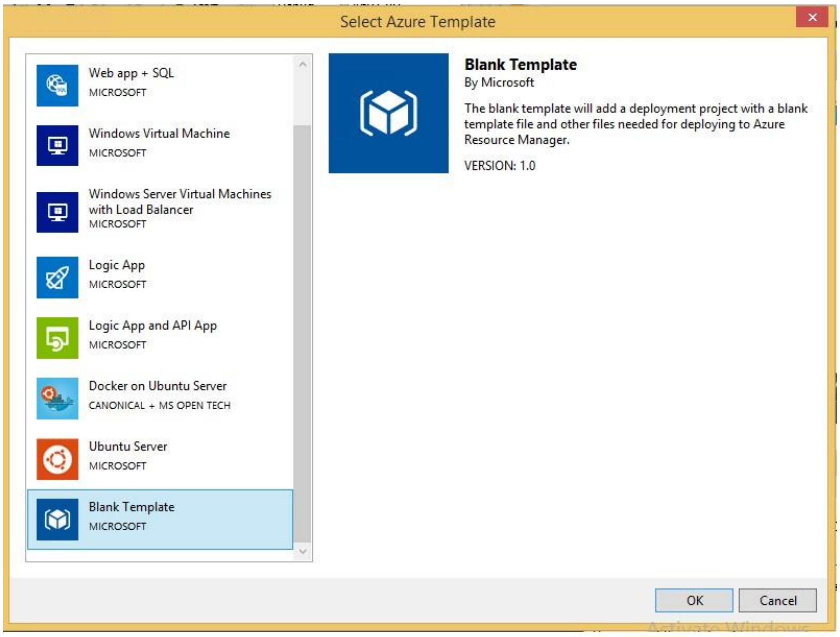
Location: c:\users\azruser\documents\visual studio 2015\Projects

Solution name: azrhol200

☑ Create directory for solution

□ Add to source control

4) For the sake of the lab we will choose Blank Template again, then click “OK”.


Select Azure Template

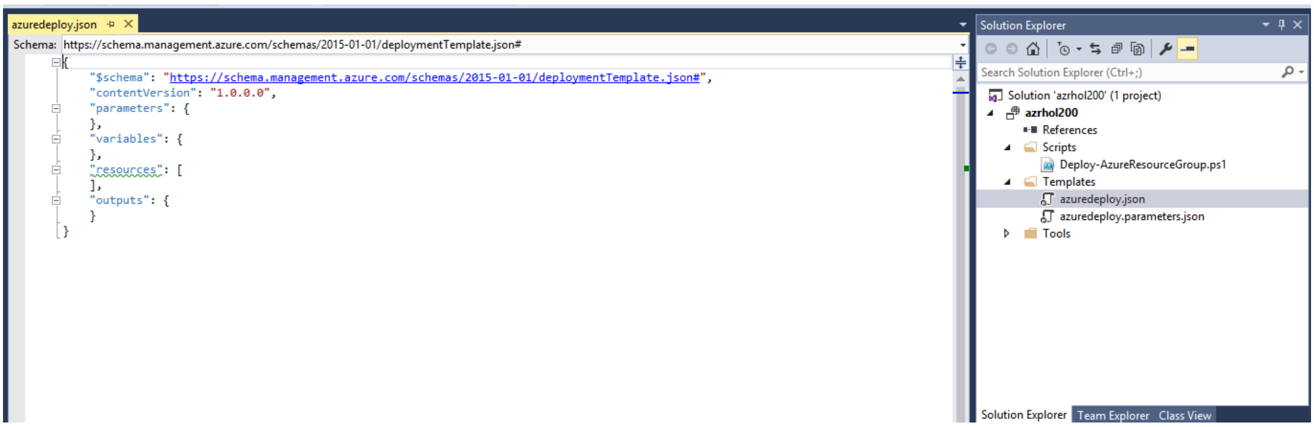
Web app + SQL	MICROSOFT
Windows Virtual Machine	MICROSOFT
Windows Server Virtual Machines with Load Balancer	MICROSOFT
Logic App	MICROSOFT
Logic App and API App	MICROSOFT
Docker on Ubuntu Server	CANONICAL + MS OPEN TECH
Ubuntu Server	MICROSOFT
Blank Template	MICROSOFT

Blank Template
By Microsoft

The blank template will add a deployment project with a blank template file and other files needed for deploying to Azure Resource Manager.

VERSION: 1.0

5) Expand the new project, then the Templates folder and finally open the **AzureDeploy.json** file. It will be empty to start.



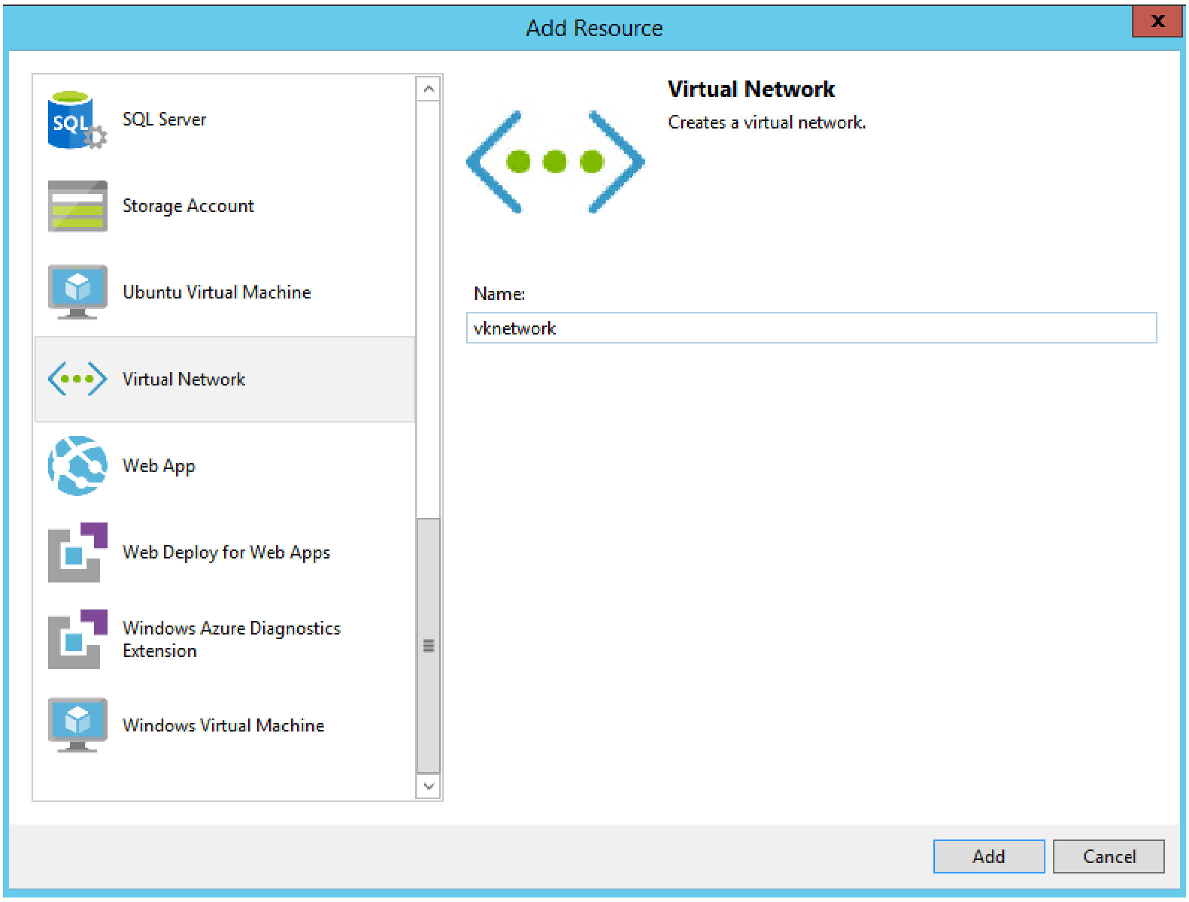
Schema: https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
    },
    "variables": {
    },
    "resources": [
    ],
    "outputs": {
    }
}
```

Task 2: Create Virtual Network

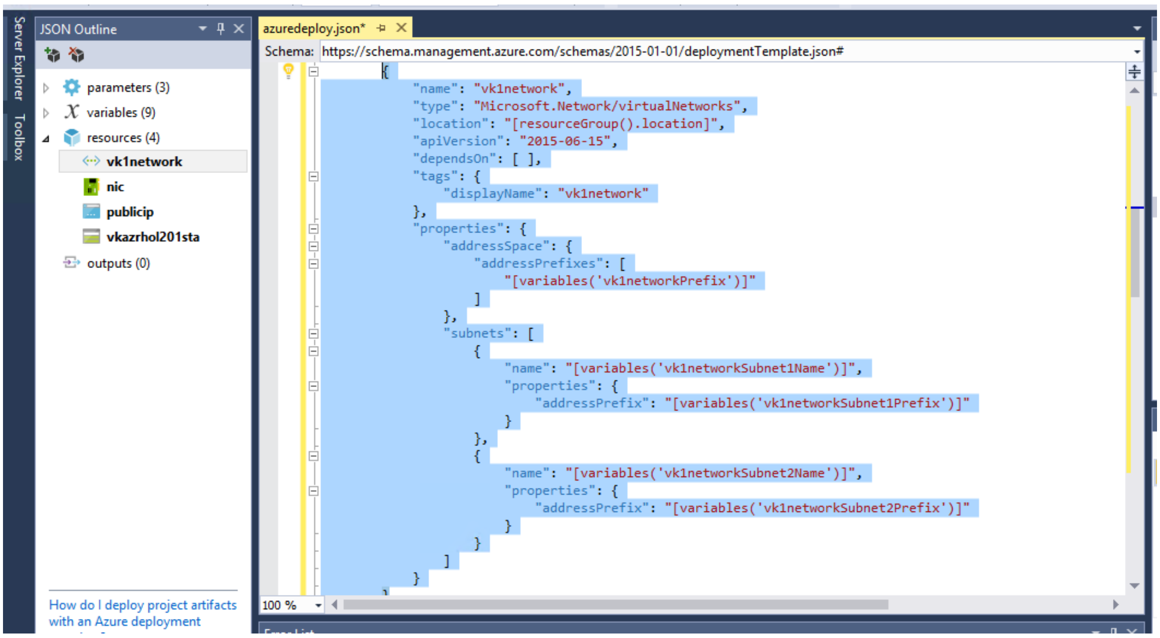
In this Task, we are going to create a Virtual Network to be used for our deployment.

1. Click the box with a plus sign on top left to add your next element. , you may need to unpin Cloud Explorer to see the JSON Outline
2. The first thing we will add is a Virtual Network. The Virtual Network is the one item that isn't dependent on any other items and so must be the first one we add. Give the Virtual Network the name <yourinitials>network and click "**Add**".



Name: vknetwork

3. If you expand out the variables node in the JSON Outline section, you will see that by default they setup a network prefix for you and 2 subnets by default. You can change this information, but for this exercise we will leave it alone. We will not be leveraging the second subnet in this exercise, but it would be removed manually by removing the references to it that the template creates.



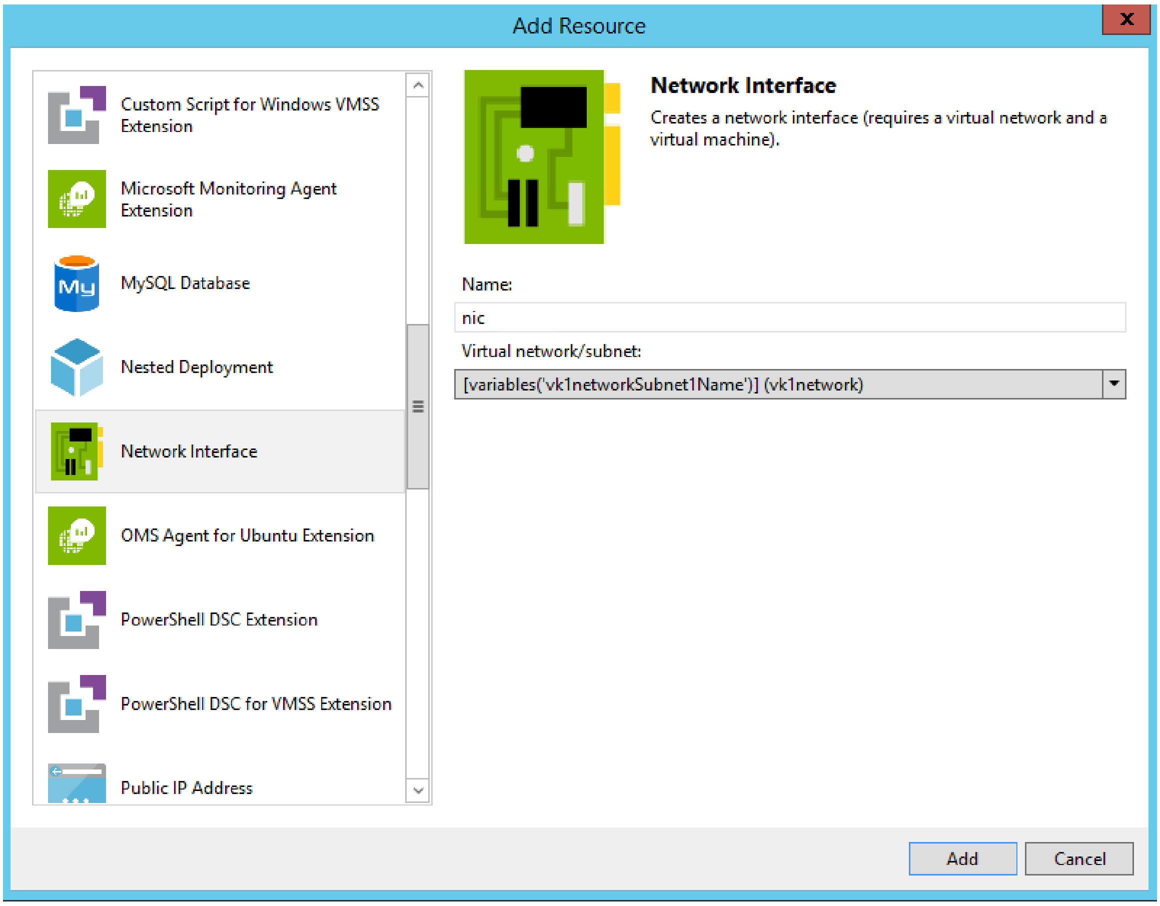
Schema: https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#

```
{
    "name": "vk1network",
    "type": "Microsoft.Network/virtualNetworks",
    "location": "[resourceGroup().location]",
    "apiVersion": "2015-06-15",
    "dependsOn": [ ],
    "tags": {
        "displayName": "vk1network"
    },
    "properties": {
        "addressSpace": {
            "addressPrefixes": [
                "[variables('vk1networkPrefix')]"
            ]
        },
        "subnets": [
            {
                "name": "[variables('vk1networkSubnet1Name')]",
                "properties": {
                    "addressPrefix": "[variables('vk1networkSubnet1Prefix')]"
                }
            },
            {
                "name": "[variables('vk1networkSubnet2Name')]",
                "properties": {
                    "addressPrefix": "[variables('vk1networkSubnet2Prefix')]"
                }
            }
        ]
    }
```

Task 3: Create Network Interfaces

v2 Networks in Azure are different than v1 Networks. You can see the difference between the networks in the new Azure Portal. With the new model for networks, you create Network Interfaces that are connected to the Virtual Network and connect those NICs to the Virtual Machines.

1. Click the box with a plus sign on top left to add your next element.
2. The next item(s) we will create is a network interfaces. Choose Network Interface, make sure that the Virtual Network/subnet has Subnet1 selected as seen in the image, give the NIC a name of nic and select subnet1name and click “Add”.


Add Resource
Custom Script for Windows VMSS Extension
Microsoft Monitoring Agent Extension
MySQL Database
Nested Deployment
Network Interface
OMS Agent for Ubuntu Extension
PowerShell DSC Extension
PowerShell DSC for VMSS Extension
Public IP Address
Add
Cancel

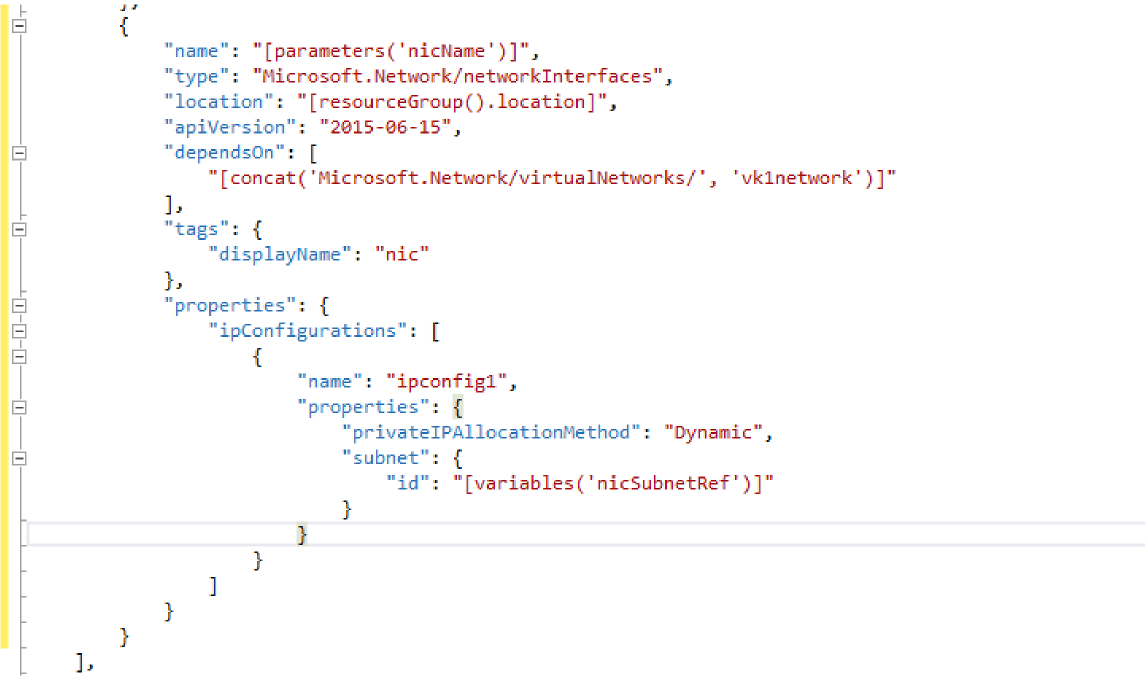
Network Interface

Creates a network interface (requires a virtual network and a virtual machine).

Name: nic

Virtual network/subnet: [variables('vk1networkSubnet1Name')] (vk1network)

3. This section of the **AzureDeploy.json** file should now look like below:



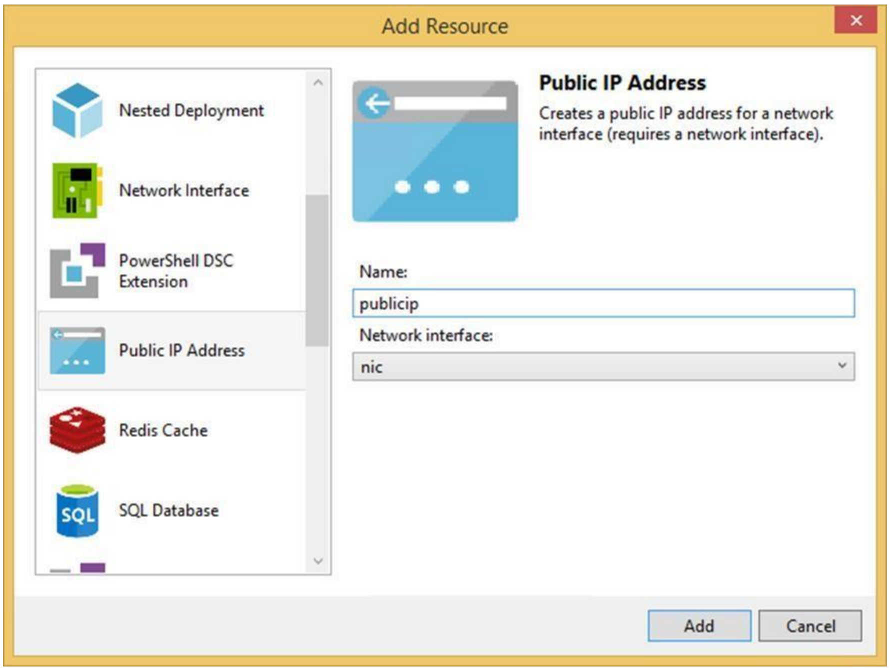
```
        },
        {
            "name": "[parameters('nicName')]",
            "type": "Microsoft.Network/networkInterfaces",
            "location": "[resourceGroup().location]",
            "apiVersion": "2015-06-15",
            "dependsOn": [
                "[concat('Microsoft.Network/virtualNetworks/', 'vk1network')]"
            ],
            "tags": {
                "displayName": "nic"
            },
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Dynamic",
                            "subnet": {
                                "id": "[variables('nicSubnetRef')]"
                            }
                        }
                    }
                ]
            }
        }
    ],
```

4. We will modify the public IPs in the next exercise

Task 4: Create Public IPs

The next elements we will create are our Public IP addresses. Public IP addresses are dependent on NIC already being created.

1. Click the box with a plus sign on top left to add your next element.
2. Choose Public IP Address, give it a name of publicip and click “Add”. You may notice that the Network Interface that is selected shows nic.



Add Resource

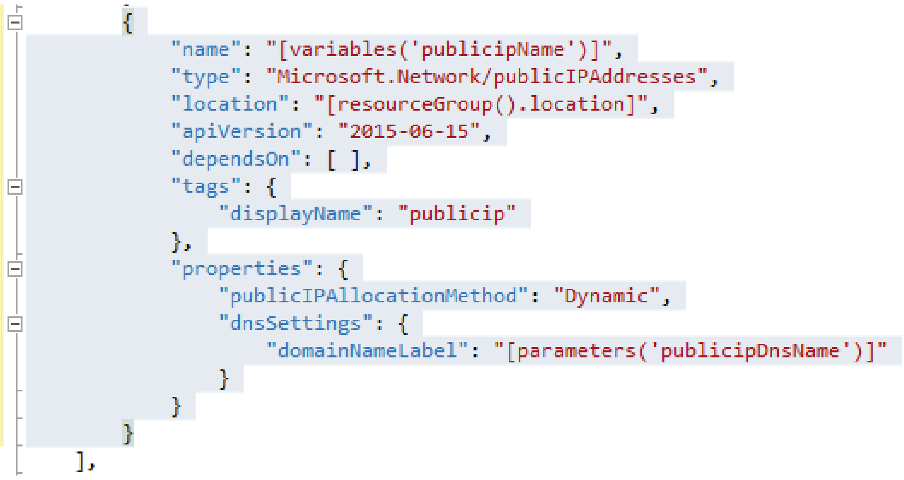
Public IP Address

Creates a public IP address for a network interface (requires a network interface).

Name: publicip

Network interface: nic

3. Your final Public IP resource section should look like below:



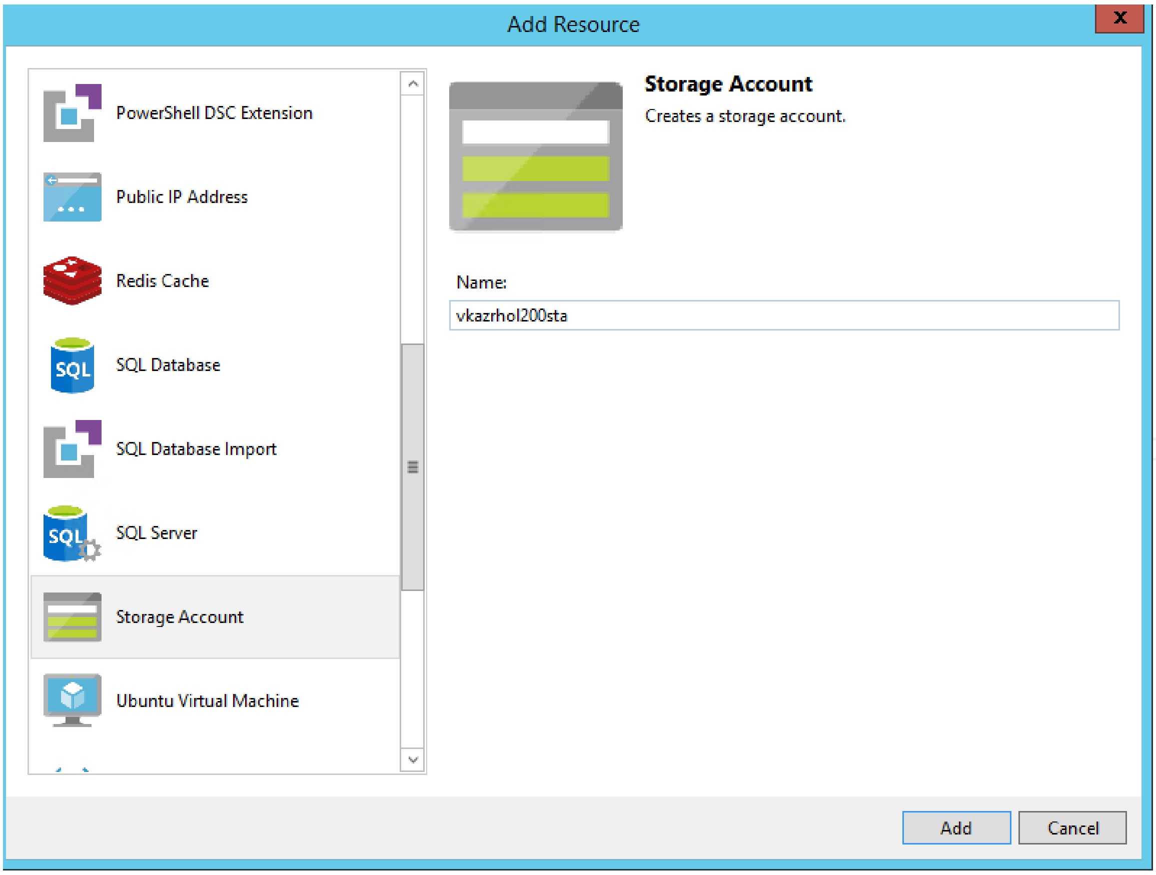
```
        {
            "name": "[variables('publicipName')]",
            "type": "Microsoft.Network/publicIPAddresses",
            "location": "[resourceGroup().location]",
            "apiVersion": "2015-06-15",
            "dependsOn": [ ],
            "tags": {
                "displayName": "publicip"
            },
            "properties": {
                "publicIPAllocationMethod": "Dynamic",
                "dnsSettings": {
                    "domainNameLabel": "[parameters('publicipDnsName')]"
                }
            }
        }
    ],
```

Task 5: Create a Storage Account

The next element we will be creating is the Storage Account. The Storage Account can be created at any point prior to creating the VM as it has no dependencies.

1. Click the box with a plus sign on top left to add your next element.
2. Choose Storage Account from the list, enter the name <yourinitials>azr200sta and click

   "**Add**".

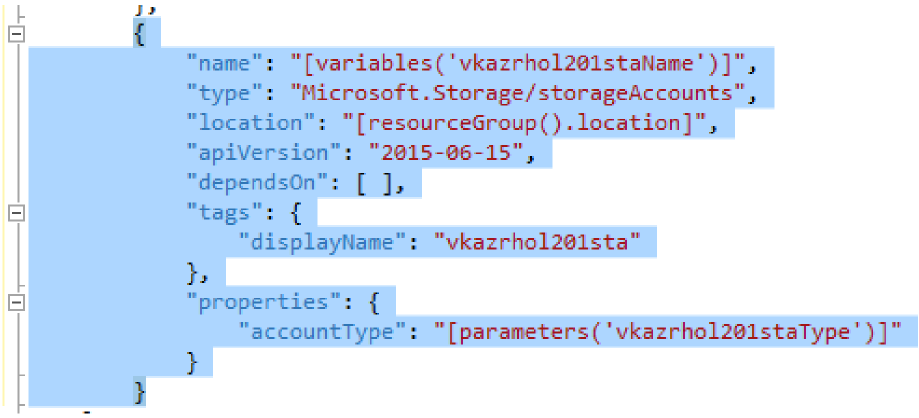


Storage Account

Creates a storage account.

Name: vkazrhol200sta

3. Your final code look like below



```
        },
        {
            "name": "[variables('vkazrhol201staName')]",
            "type": "Microsoft.Storage/storageAccounts",
            "location": "[resourceGroup().location]",
            "apiVersion": "2015-06-15",
            "dependsOn": [ ],
            "tags": {
                "displayName": "vkazrhol201sta"
            },
            "properties": {
                "accountType": "[parameters('vkazrhol201staType')]"
            }
        }
```

4. Change the variable <yourinitials>azrhol200staName to remove unique string. Search for concat and find this line

```
↪ "vkazrhol201staName": "[concat('vkazrhol201sta',
uniqueString(resourceGroup().id))]",
```

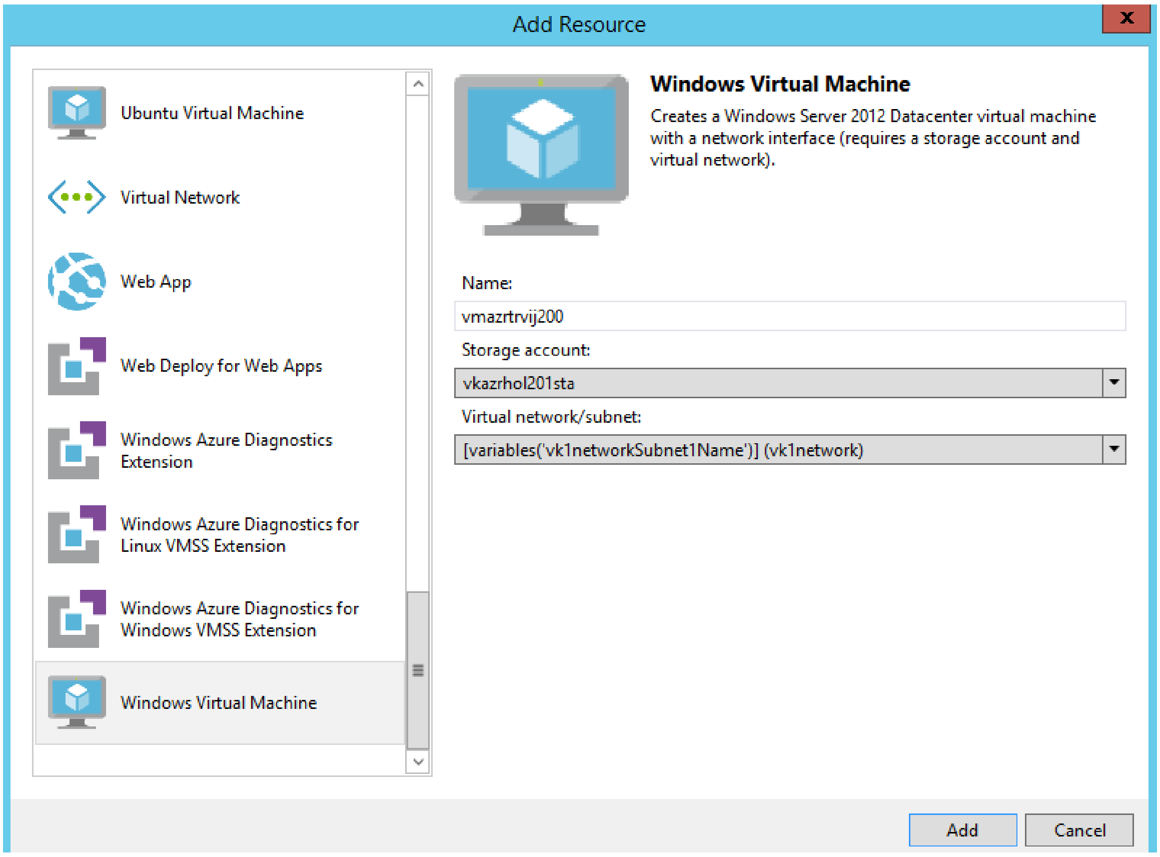
5. Change the above line to as below:

```
↪ "vkazrhol201staName": "[concat('vkazrhol200sta', 'tr')]",
```

Task 6: Create Virtual Machines

The final element to be created will be the Virtual Machines.

1. Click the box with a plus sign on top left to add your next element.
2. Click Windows Virtual Machine, fill out the name as vm1, choose the Storage Account you just created, and choose the Subnet1 from your virtual network and click “Add”.



Add Resource

- Ubuntu Virtual Machine
- Virtual Network
- Web App
- Web Deploy for Web Apps
- Windows Azure Diagnostics Extension
- Windows Azure Diagnostics for Linux VMSS Extension
- Windows Azure Diagnostics for Windows VMSS Extension
- Windows Virtual Machine

Windows Virtual Machine

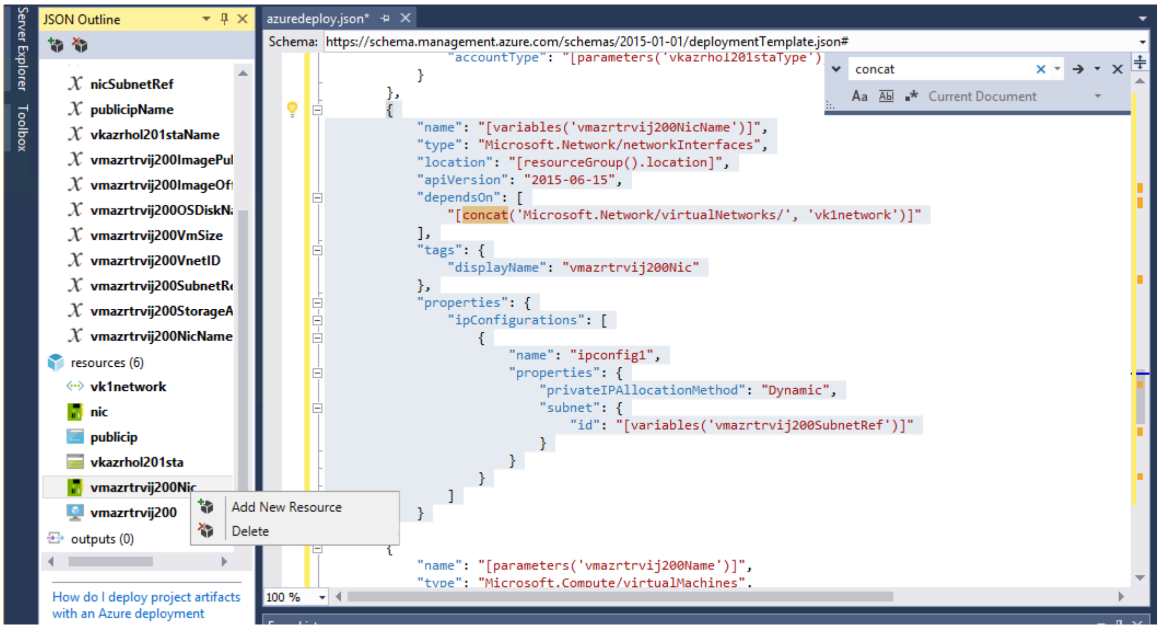
Creates a Windows Server 2012 Datacenter virtual machine with a network interface (requires a storage account and virtual network).

Name: vmazrtrvij200

Storage account: vkazrhol201sta

Virtual network/subnet: [variables('vk1networkSubnet1Name')] (vk1network)

3. After adding the Virtual Machine resource, more variables and parameters will appear.
4. Additionally, a new Network Interface will be created. We will right click that interface and choose “Delete”, as we will be using the one we made earlier.



JSON Outline:

- nicSubnetRef
- publicipName
- vkazrhol201staName
- vmazrtrvij200ImagePul
- vmazrtrvij200ImageOf
- vmazrtrvij200OSDiskN
- vmazrtrvij200VmSize
- vmazrtrvij200VnetID
- vmazrtrvij200SubnetR
- vmazrtrvij200StorageA
- vmazrtrvij200NicName
- resources (6)
  - vk1network
  - nic
  - publicip
  - vkazrhol201sta
  - vmazrtrvij200Nic
  - vmazrtrvij200
- outputs (0)

Schema: https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#

```
                "accountType": "[parameters('vkazrhol201staType')]
            }
        },
        {
            "name": "[variables('vmazrtrvij200NicName')]",
            "type": "Microsoft.Network/networkInterfaces",
            "location": "[resourceGroup().location]",
            "apiVersion": "2015-06-15",
            "dependsOn": [
                "[concat('Microsoft.Network/virtualNetworks/', 'vk1network')]"
            ],
            "tags": {
                "displayName": "vmazrtrvij200Nic"
            },
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Dynamic",
                            "subnet": {
                                "id": "[variables('vmazrtrvij200SubnetRef')]"
                            }
                        }
                    }
                ]
            }
        },
        {
            "name": "[parameters('vmazrtrvij200Name')]",
            "type": "Microsoft.Compute/virtualMachines",
```

5. Find the following lines of code which define what elements the VM depends on.

```
↪ "dependsOn": [
  "[concat('Microsoft.Storage/storageAccounts/', variables('
  vkazrhol201staName'))]",
↪
  "[concat('Microsoft.Network/networkInterfaces/',
  variables('vm1NicName'))]"
↪       ],
```

6. Update the code with the lines found below. This will make sure that we reference the cloned NICs created earlier.

```
↪ "dependsOn": [
↪  "[concat('Microsoft.Storage/storageAccounts/', variables('
  vkazrhol201staName'))]",
↪
  "[concat('Microsoft.Network/networkInterfaces/',
  parameters('nicName'))]"
↪         ],
```

7. Next, we will be updating the NICs defined in the Network Profile for the VMs. Find the following lines of code:

```
↪ "networkProfile": {
↪   "networkInterfaces": [
↪     {
   "id": "[resourceId('Microsoft.Network/networkInterfaces',
  variables('vm1NicName'))]"
↪                         }
```

8. Replace the above lines of code with the lines below. This will ensure that a new NIC is used for each Virtual Machine.

```
↪ "networkProfile": {
↪     "networkInterfaces": [
↪                       {
↪   "id": "[resourceId('Microsoft.Network/networkInterfaces',
  parameters('nicName'))]"
↪                         }
```

9. Finally, your VM resource should look like the following:

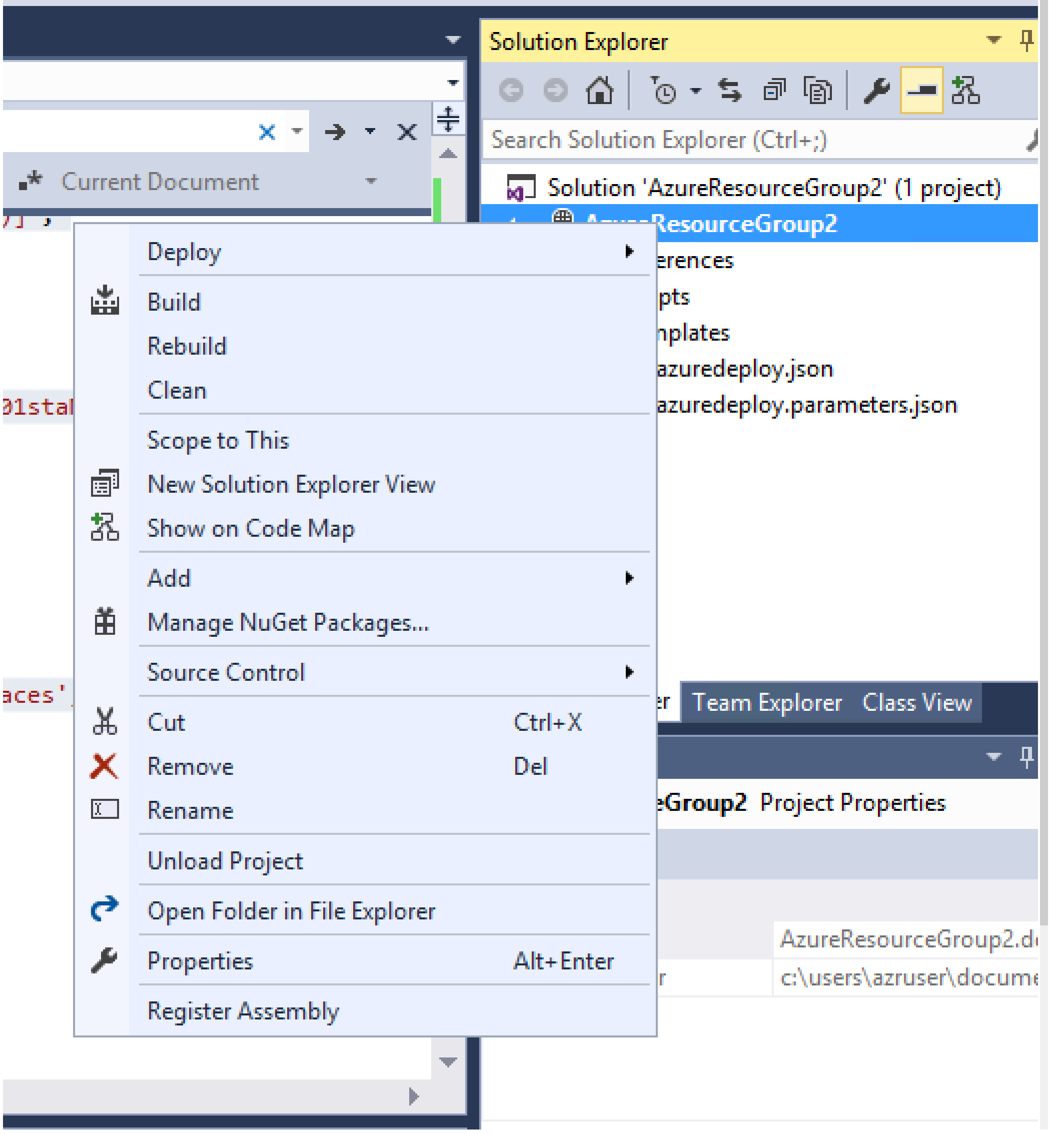
```
↪ {
↪         {
↪             "name": "[parameters('vmazrtrvij200Name')]",
↪             "type": "Microsoft.Compute/virtualMachines",
↪             "location": "[resourceGroup().location]",
↪             "apiVersion": "2015-06-15",
↪           "dependsOn": [
↪             "[concat('Microsoft.Storage/storageAccounts/',
  variables('vkazrhol201staName'))]",
↪             "[concat('Microsoft.Network/networkInterfaces/',
  parameters('nicName'))]"
↪           ],
↪             "tags": {
↪                 "displayName": "vmazrtrvij200"
↪             },
↪             "properties": {
↪                 "hardwareProfile": {
↪                     "vmSize":
  "[variables('vmazrtrvij200VmSize')]"
↪                 },
↪                 "osProfile": {
↪                     "computerName":
  "[parameters('vmazrtrvij200Name')]",
↪                     "adminUsername":
  "[parameters('vmazrtrvij200AdminUsername')]",
↪                     "adminPassword":
  "[parameters('vmazrtrvij200AdminPassword')]"
↪                 },
↪                 "storageProfile": {
↪                     "imageReference": {
```

```
↪                         "publisher":
  "[variables('vmazrtrvij200ImagePublisher')]",
↪                         "offer":
  "[variables('vmazrtrvij200ImageOffer')]",
↪                         "sku":
  "[parameters('vmazrtrvij200WindowsOSVersion')]",
↪                         "version": "latest"
↪                     },
↪                     "osDisk": {
↪                         "name": "vmazrtrvij200OSDisk",
↪                         "vhd": {
↪                             "uri": "[concat('http://',
  variables('vkazrhol201staName'), '.blob.core.windows.net/',
  variables('vmazrtrvij200StorageAccountContainerName'), '/',
  variables('vmazrtrvij200OSDiskName'), '.vhd')]"
↪                         },
↪                         "caching": "ReadWrite",
↪                         "createOption": "FromImage"
↪                     }
↪                 },
↪                 "networkProfile": {
↪                     "networkInterfaces": [
↪                       {
↪                         "id":
  "[resourceId('Microsoft.Network/networkInterfaces',
  parameters('nicName'))]"
↪                       }
↪                     ]
↪                 }
↪             }
↪         }
↪     ],
```

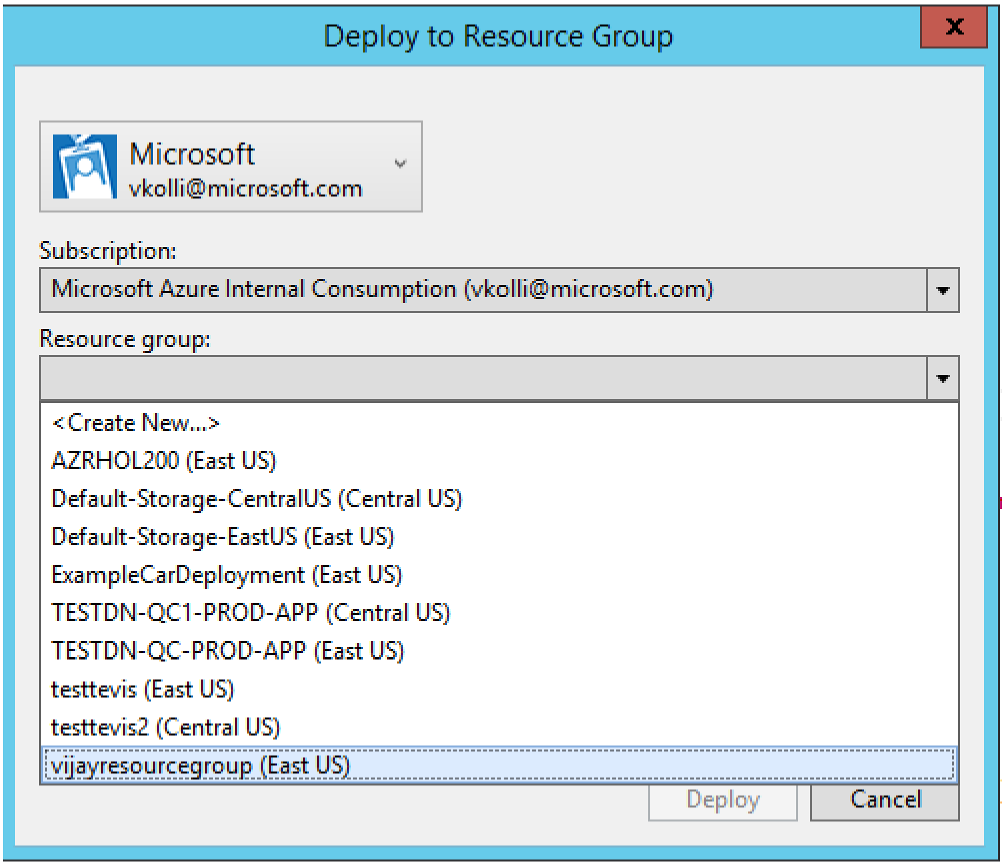
Task 7: Deploy the Resources

Now that all the resources are in the Deployment Template it is time to deploy them to Azure.

1. Ensure you are logged into your Azure Subscription. In order to connect Visual Studio to Azure, click Server Explorer. Right-click on the current connection and select Connect to Microsoft Azure Subscription
2. Right click the Project and choose Deploy and then New Deployment…



3. On the logged in Account dropdown, select your account (Visual Studio may have the lab author account listed, after logging in your account should appear in the dropdown and will take a moment to load your information). On the Deploy to Resource Group Dialog Window, pull down the Resource Group dropdown and choose <Create New…>



Deploy to Resource Group

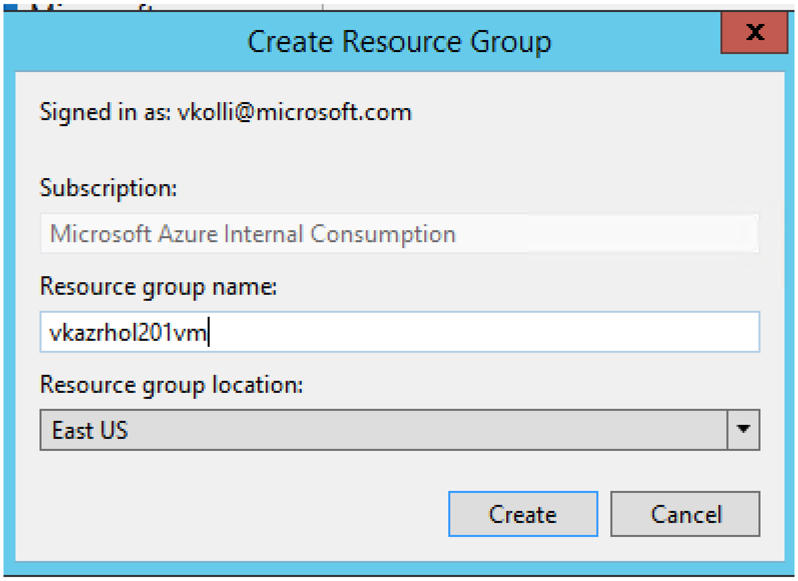
Microsoft
vkolli@microsoft.com

Subscription: Microsoft Azure Internal Consumption (vkolli@microsoft.com)

Resource group:

- <Create New...>
- AZRHOL200 (East US)
- Default-Storage-CentralUS (Central US)
- Default-Storage-EastUS (East US)
- ExampleCarDeployment (East US)
- TESTDN-QC1-PROD-APP (Central US)
- TESTDN-QC-PROD-APP (East US)
- testtevis (East US)
- testtevis2 (Central US)
- vijayresourcegroup (East US)

4. Put in the resource group name of <yourintials>azrhol201vm and choose East US as your Resource Group Location and click “Create”.



Create Resource Group

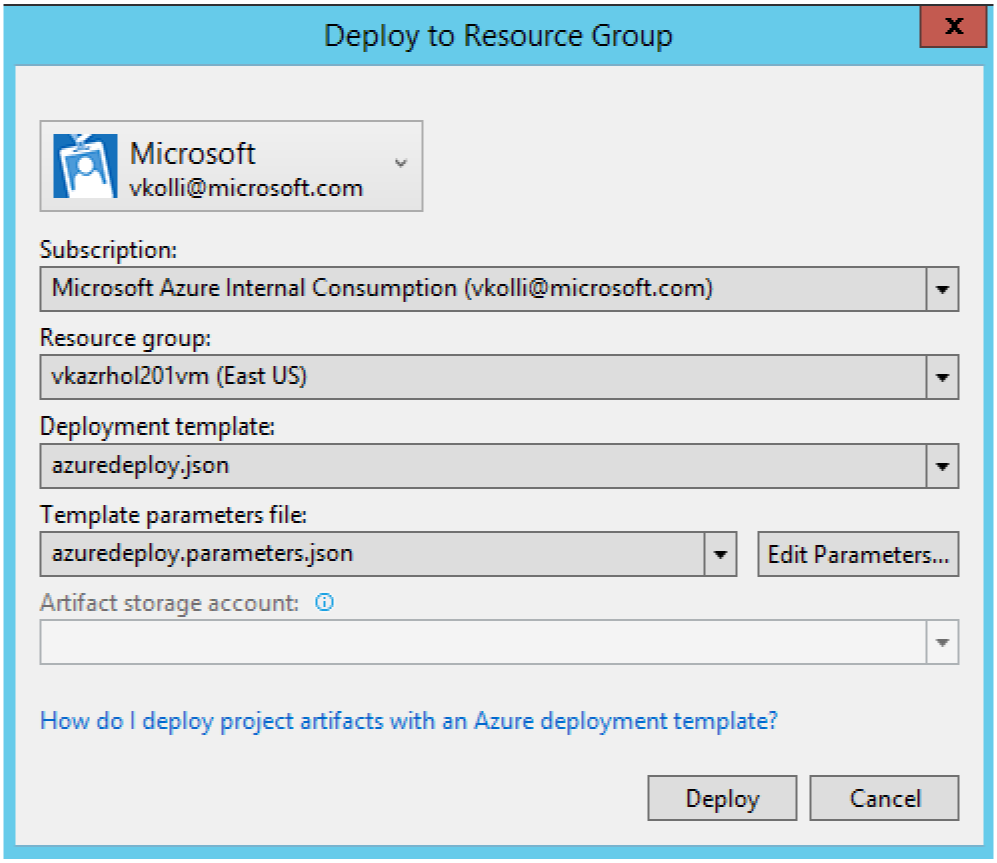
Signed in as: vkolli@microsoft.com

Subscription: Microsoft Azure Internal Consumption

Resource group name: vkazrhol201vm

Resource group location: East US

5. Click the Edit Parameters button and fill out the parameters and click “Save”.



Deploy to Resource Group

Microsoft
vkolli@microsoft.com

Subscription: Microsoft Azure Internal Consumption (vkolli@microsoft.com)

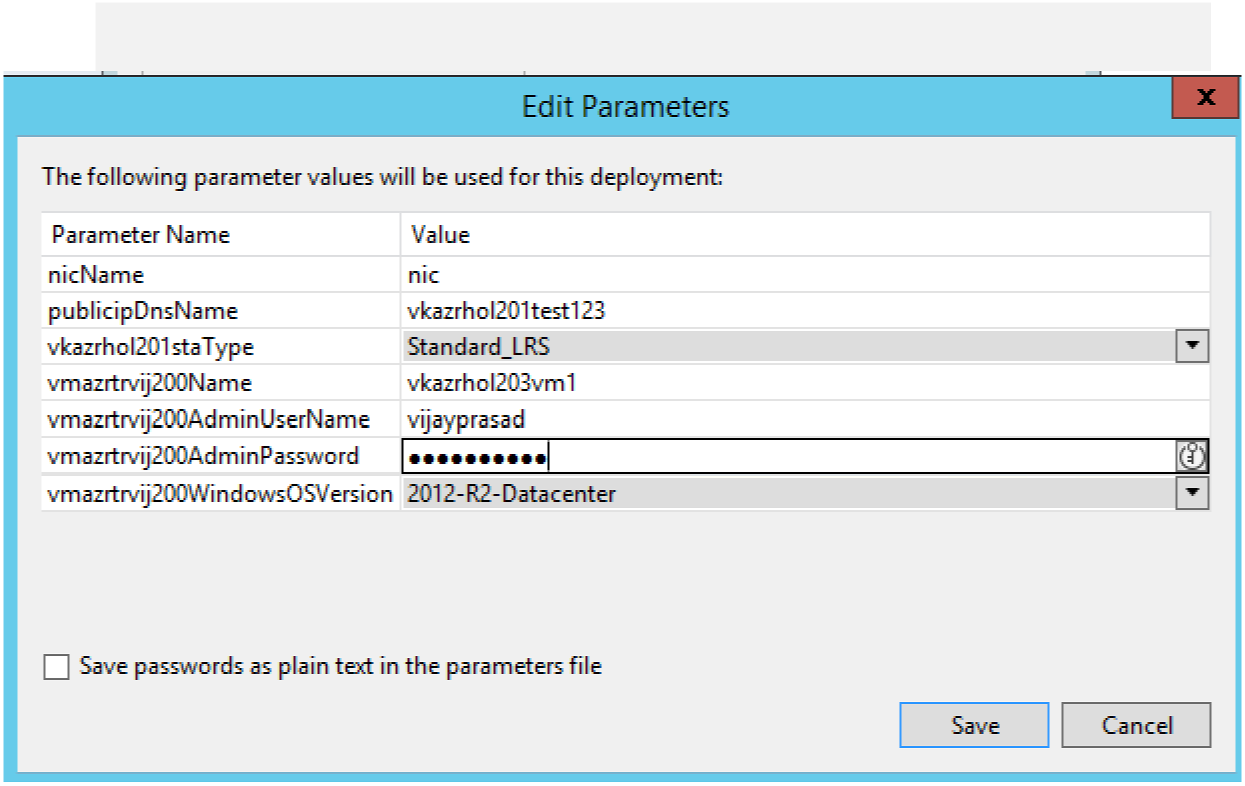
Resource group: vkazrhol201vm (East US)

Deployment template: azuredeploy.json

Template parameters file: azuredeploy.parameters.json

Artifact storage account:

How do I deploy project artifacts with an Azure deployment template?



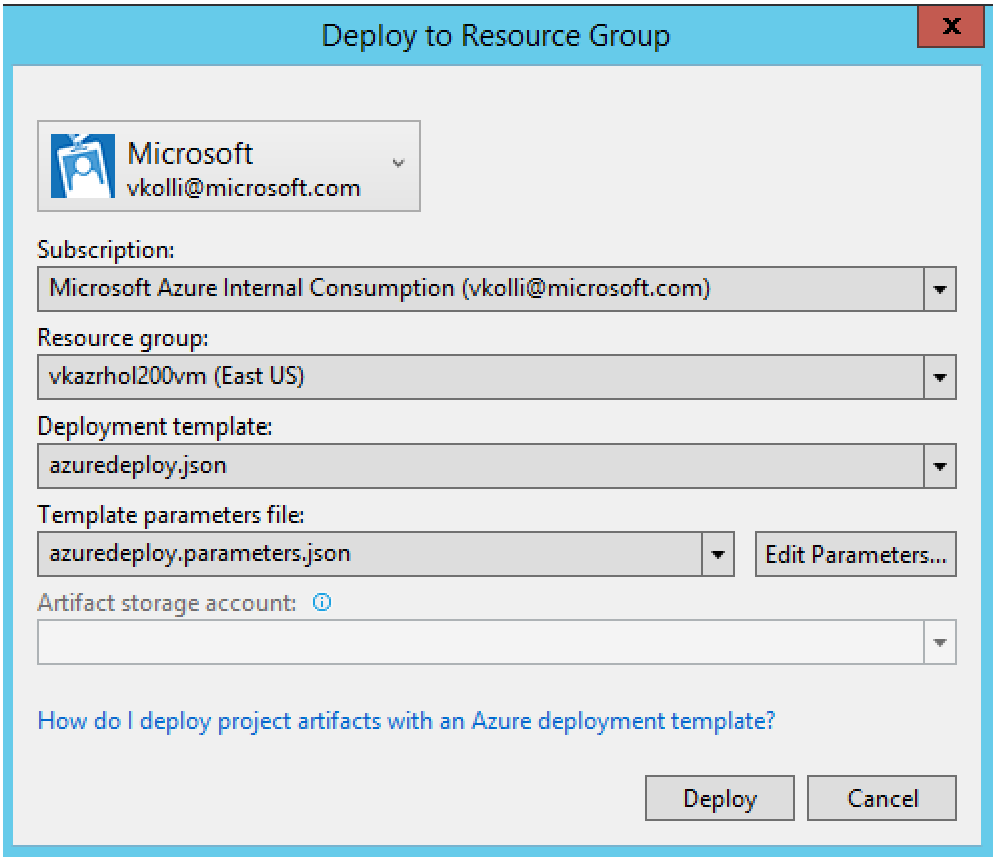
Edit Parameters

The following parameter values will be used for this deployment:

Parameter Name	Value
nicName	nic
publicipDnsName	vkazrhol201test123
vkazrhol201staType	Standard_LRS
vmazrtrvij200Name	vkazrhol203vm1
vmazrtrvij200AdminUserName	vijayprasad
vmazrtrvij200AdminPassword	●●●●●●●●●●●
vmazrtrvij200WindowsOSVersion	2012-R2-Datacenter

☐ Save passwords as plain text in the parameters file

6. Click “Deploy” and the deployment will begin and it will ask you for VM1 password.



Microsoft
vkolli@microsoft.com

Subscription: Microsoft Azure Internal Consumption (vkolli@microsoft.com)

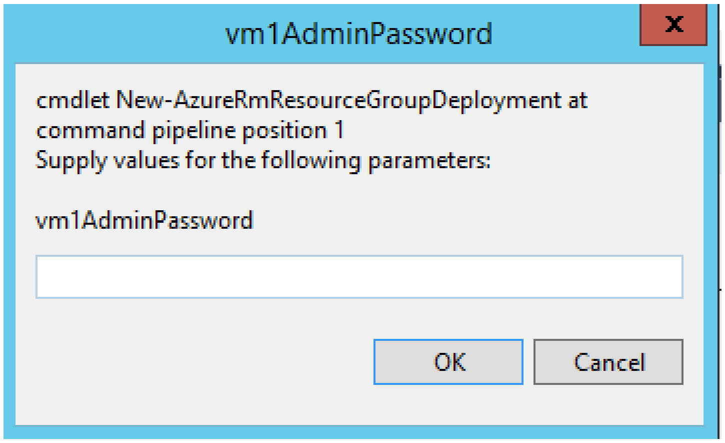
Resource group: vkazrhol200vm (East US)

Deployment template: azuredeploy.json

Template parameters file: azuredeploy.parameters.json

Artifact storage account:

How do I deploy project artifacts with an Azure deployment template?



cmdlet New-AzureRmResourceGroupDeployment at command pipeline position 1
Supply values for the following parameters:

vm1AdminPassword

7. If everything was successful, your results should look like below.

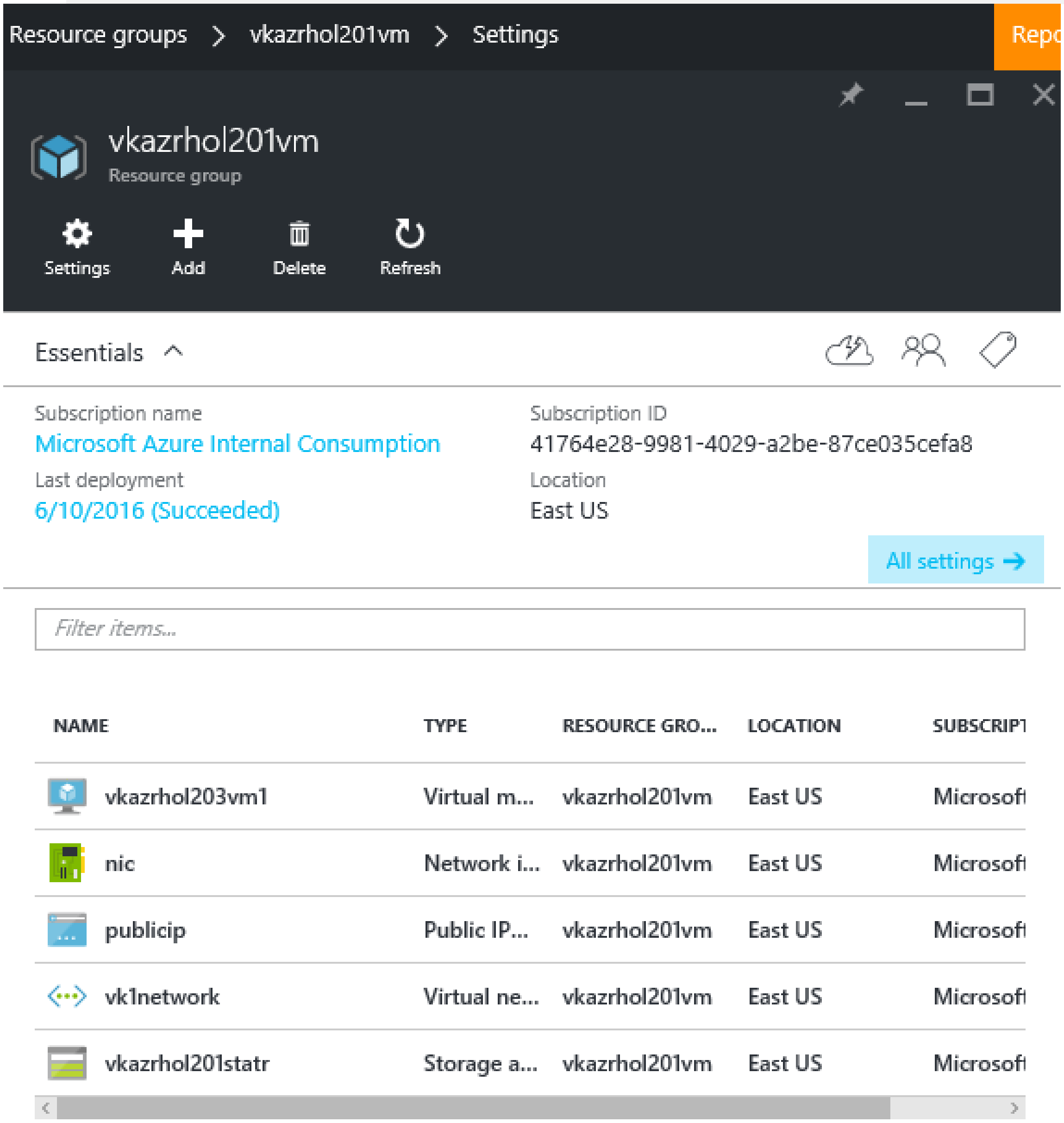
```
20:39:07 - Build started.
20:39:08 - Project "AzureResourceGroup2.deployproj" (StageArtifacts target(s)):
20:39:08 - Project "AzureResourceGroup2.deployproj" (ContentFilesProjectOutputGroup
target(s)):
20:39:08 - Done building project "AzureResourceGroup2.deployproj".
20:39:08 - Done building project "AzureResourceGroup2.deployproj".
20:39:08 - Build succeeded.
```

```
20:39:08 - The following parameter values will be used for this deployment:
20:39:08 -     nicName: nic
20:39:08 -     publicipDnsName: vkazrhol201test123
20:39:08 -     vkazrhol201staType: Standard_LRS
20:39:08 -     vmazrtrvij200Name: vkazrhol203vm1
20:39:08 -     vmazrtrvij200AdminUserName: vijayprasad
20:39:08 -     vmazrtrvij200AdminPassword: <securestring>
20:39:08 -     vmazrtrvij200WindowsOSVersion: 2012-R2-Datacenter
20:39:08 - Launching deployment PowerShell script with the following command:
20:39:08 - 'c:\users\azruser\documents\visual studio
2015\projects\azureresourcegroup2\azureresourcegroup2\Scripts\Deploy-
AzureResourceGroup.ps1' -StorageAccountName '' -ResourceGroupName 'vkazrhol201vm' -
ResourceGroupLocation 'eastus' -TemplateFile 'c:\users\azruser\documents\visual studio
2015\projects\azureresourcegroup2\azureresourcegroup2\templates\azuredeploy.json' -
TemplateParametersFile 'c:\users\azruser\documents\visual studio
2015\projects\azureresourcegroup2\azureresourcegroup2\templates\azuredeploy.parameters.js
on' -ArtifactStagingDirectory '..\bin\Debug\staging'
20:39:13 - 
20:39:13 - 
20:39:13 - Environment           : AzureCloud
20:39:13 - Account               : vkolli@microsoft.com
20:39:13 - TenantId              : 72f988bf-86f1-41af-91ab-2d7cd011db47
20:39:13 - SubscriptionId        : 41764e28-9981-4029-a2be-87ce035cefa8
20:39:13 - SubscriptionName      : Microsoft Azure Internal Consumption
20:39:13 - CurrentStorageAccount : 
20:39:13 - 
20:39:15 - [WARNING] The usability of Tag parameter in this cmdlet will be modified in a
future release. This will impact creating, updating and appending tags for Azure
resources. For more details about the change, please visit
https://github.com/Azure/azure-powershell/issues/726#issuecomment-213545494
20:39:16 - [VERBOSE] 8:39:16 PM - Created resource group 'vkazrhol201vm' in location
'eastus'
20:39:16 - 
20:39:16 - ResourceGroupName : vkazrhol201vm
20:39:16 - Location          : eastus
20:39:16 - ProvisioningState : Succeeded
20:39:16 - Tags              : {}
20:39:16 - TagsTable         : 
20:39:16 - ResourceId        : /subscriptions/41764e28-9981-4029-a2be-
87ce035cefa8/resourc
20:39:16 -                     eGroups/vkazrhol201vm
20:39:16 - 
20:39:16 - cmdlet New-AzureRmResourceGroupDeployment at command pipeline position 1
Supply values for the following parameters:
20:39:33 - vmazrtrvij200AdminPassword: 
20:39:34 - [VERBOSE] 8:39:34 PM - Template is valid.
20:39:34 - [VERBOSE] 8:39:34 PM - Create template deployment 'azuredeploy-0609-2039'
20:39:34 - [VERBOSE] 8:39:34 PM - Checking deployment status in 5 seconds
20:39:39 - [VERBOSE] 8:39:39 PM - Resource Microsoft.Storage/storageAccounts
'vkazrhol201statr' provisioning status is running
20:39:39 - [VERBOSE] 8:39:39 PM - Resource Microsoft.Network/publicIPAddresses 'publicip'
provisioning status is running
20:39:39 - [VERBOSE] 8:39:39 PM - Resource Microsoft.Network/virtualNetworks 'vk1network'
provisioning status is running
20:39:39 - [VERBOSE] 8:39:39 PM - Checking deployment status in 10 seconds
```

```
20:39:49 - [VERBOSE] 8:39:49 PM - Resource Microsoft.Network/virtualNetworks 'vk1network'
provisioning status is succeeded
20:39:49 - [VERBOSE] 8:39:49 PM - Checking deployment status in 15 seconds
20:40:05 - [VERBOSE] 8:40:05 PM - Resource Microsoft.Network/networkInterfaces 'nic'
provisioning status is succeeded
20:40:05 - [VERBOSE] 8:40:05 PM - Resource Microsoft.Storage/storageAccounts
'vkazrhol201statr' provisioning status is succeeded
20:40:05 - [VERBOSE] 8:40:05 PM - Resource Microsoft.Network/publicIPAddresses 'publicip'
provisioning status is succeeded
20:40:05 - [VERBOSE] 8:40:05 PM - Checking deployment status in 20 seconds
20:40:25 - [VERBOSE] 8:40:25 PM - Resource Microsoft.Compute/virtualMachines
'vkazrhol203vm1' provisioning status is running
20:40:25 - [VERBOSE] 8:40:25 PM - Checking deployment status in 25 seconds
20:40:50 - [VERBOSE] 8:40:50 PM - Checking deployment status in 30 seconds
20:41:20 - [VERBOSE] 8:41:20 PM - Checking deployment status in 35 seconds
20:41:56 - [VERBOSE] 8:41:56 PM - Checking deployment status in 40 seconds
20:42:36 - [VERBOSE] 8:42:36 PM - Checking deployment status in 45 seconds
20:43:21 - [VERBOSE] 8:43:21 PM - Checking deployment status in 50 seconds
20:44:11 - [VERBOSE] 8:44:11 PM - Resource Microsoft.Compute/virtualMachines
'vkazrhol203vm1' provisioning status is succeeded
20:44:11 -
20:44:11 - DeploymentName          : azuredeploy-0609-2039
20:44:11 - CorrelationId           : c6591865-1563-4b6b-bf78-3cf4d7ea330b
20:44:11 - ResourceGroupName       : vkazrhol201vm
20:44:11 - ProvisioningState       : Succeeded
20:44:11 - Timestamp               : 6/9/2016 8:43:22 PM
20:44:11 - Mode                    : Incremental
20:44:11 - TemplateLink            :
20:44:11 - TemplateLinkString      :
20:44:11 - DeploymentDebugLogLevel :
20:44:11 - Parameters              : {[nicName,
Microsoft.Azure.Commands.ResourceManager.C
20:44:11 -                           mdlets.SdkModels.DeploymentVariable],
20:44:11 -                           [publicipDnsName,
Microsoft.Azure.Commands.ResourceMa
20:44:11 -                           nager.Cmdlets.SdkModels.DeploymentVariable],
20:44:11 -                           [vkazrhol201staType,
Microsoft.Azure.Commands.Resourc
20:44:11 -                           eManager.Cmdlets.SdkModels.DeploymentVariable],
20:44:11 -                           [vmazrtrvij200Name,
Microsoft.Azure.Commands.Resource
20:44:11 -                           Manager.Cmdlets.SdkModels.DeploymentVariable]...}
20:44:11 - ParametersString        :
20:44:11 -                           Name             Type                       Value
20:44:11 -
20:44:11 -                           ===============  =========================
20:44:11 -                           ==========
20:44:11 -                           nicName          String                     nic
20:44:11 -
20:44:11 -                           publicipDnsName  String
20:44:11 -                           vkazrhol201test123
20:44:11 -                           vkazrhol201staType  String
20:44:11 -                           Standard_LRS
20:44:11 -                           vmazrtrvij200Name  String
20:44:11 -                           vkazrhol203vm1
```

```
20:44:11 -                           vmazrtrvij200AdminUserName  String
20:44:11 -                             vijayprasad
20:44:11 -                           vmazrtrvij200AdminPassword  SecureString
20:44:11 -
20:44:11 -                           vmazrtrvij200WindowsOSVersion  String
20:44:11 -                                2012-R2-Datacenter
20:44:11 -
20:44:11 - Outputs                 : {}
20:44:11 - OutputsString           :
20:44:11 -
20:44:11 -
20:44:11 -
20:44:11 -
20:44:11 - Successfully deployed template 'c:\users\azruser\documents\visual studio
2015\projects\azureresourcegroup2\azureresourcegroup2\templates\azuredeploy.json' to
resource group 'vkazrhol201vm'.
```

8. You can then go to http://portal.azure.com and look at your Resource Group deployment and see all your components are there.



Subscription name
Microsoft Azure Internal Consumption

Subscription ID
41764e28-9981-4029-a2be-87ce035cefa8

Last deployment
6/10/2016 (Succeeded)

Location
East US

Filter items...

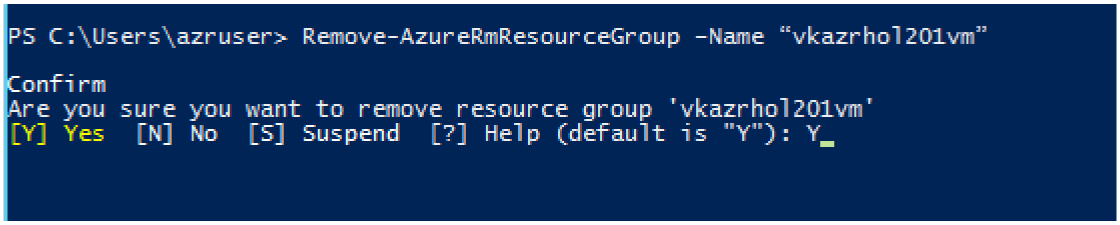
NAME	TYPE	RESOURCE GRO...	LOCATION	SUBSCRIPT
vkazrhol203vm1	Virtual m...	vkazrhol201vm	East US	Microsof
nic	Network i...	vkazrhol201vm	East US	Microsof
publicip	Public IP...	vkazrhol201vm	East US	Microsof
vk1network	Virtual ne...	vkazrhol201vm	East US	Microsof
vkazrhol201statr	Storage a...	vkazrhol201vm	East US	Microsof

Exercise 4: Delete Resource Group

Once we are done and we no longer require the environment, it's time to delete. The following is a great way to cleanup. It deletes everything that is created within the resource group. One thing to keep in mind is this cannot be undo.

1.

```
Remove-AzureRmResourceGroup –Name "vkazrhol201vm"
```



```
PS C:\Users\azruser> Remove-AzureRmResourceGroup -Name "vkazrhol201vm"

Confirm
Are you sure you want to remove resource group 'vkazrhol201vm'
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
```

Wait for a few minutes and the resource group and all the resources will be deleted

Exercise 5: Deploy a VM to Azure using GitHub

Introduction

In this exercise, we will be deploying a VM to Azure with a Resource Group and leveraging GitHub as our source for the Resources instead of Visual Studio.

Objectives

After completing this lab, you will be able to:

- Create a VM in Azure
- Understand how GitHub and Azure work together as it related to ARM

Estimated time to complete this exercise

15 minutes

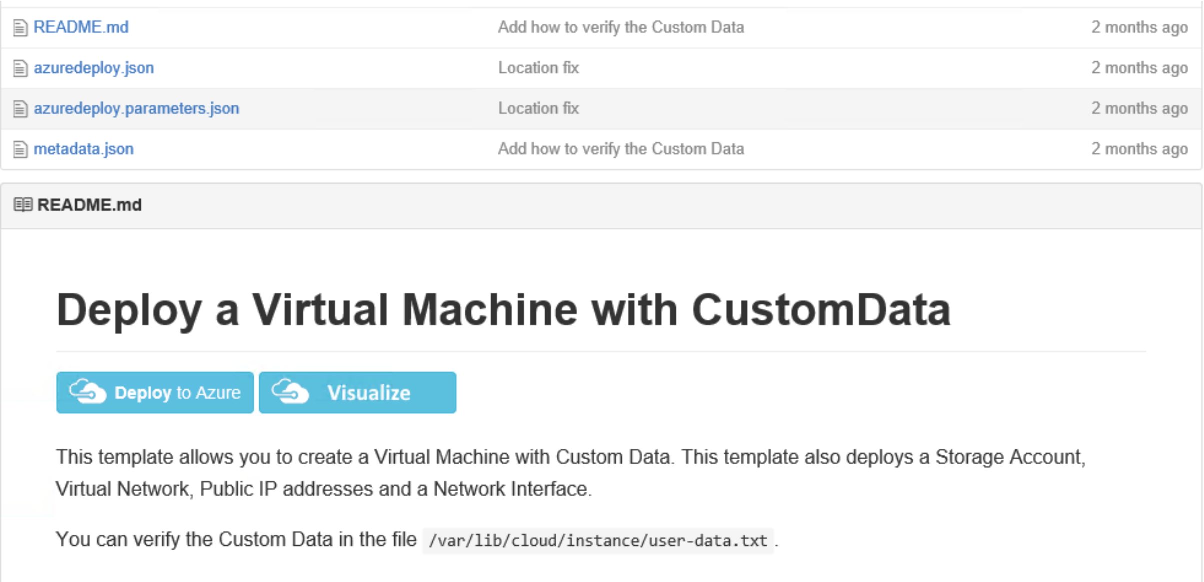
Scenario

Creating Resources in Azure leveraging Azure Resource Manager and GitHub help speed the deployment of Websites to Azure. Additionally, this allows us to share code and work openly and collaboratively with others.

Task 1: Navigate to GitHub

For this exercise, pre-configured templates have been hosted in GitHub. There is also the Azure QuickStart Templates that can be found at https://github.com/Azure/azure-quickstart-templates.

1. https://github.com/Azure/azure-quickstart-templates/tree/master/101-vm-customdata in your browser.



README.md	Add how to verify the Custom Data	2 months ago
azuredeploy.json	Location fix	2 months ago
azuredeploy.parameters.json	Location fix	2 months ago
metadata.json	Add how to verify the Custom Data	2 months ago

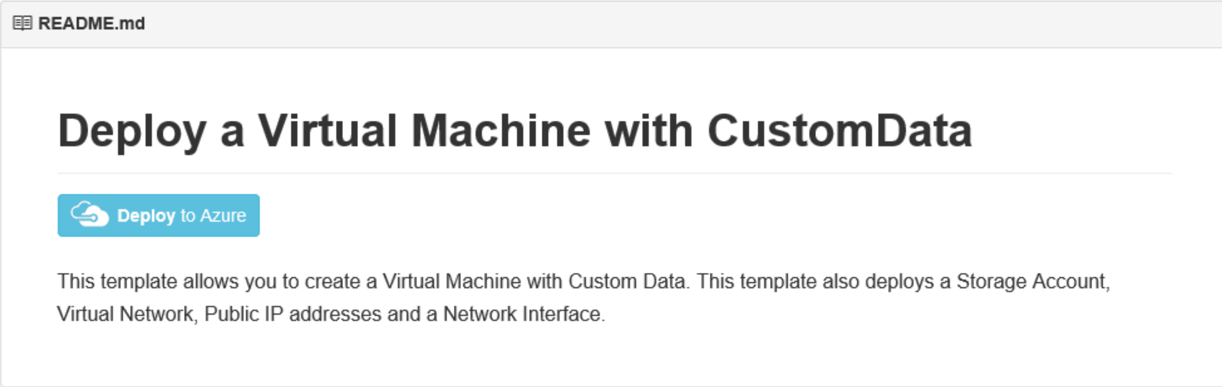
README.md

Deploy a Virtual Machine with CustomData

This template allows you to create a Virtual Machine with Custom Data. This template also deploys a Storage Account, Virtual Network, Public IP addresses and a Network Interface.

You can verify the Custom Data in the file `/var/lib/cloud/instance/user-data.txt`.

2. This example has 3 files: ReadMe.md, azuredeploy.json and metadata.json. These files are used to help us deploy the example to Azure. The text you see in the window below is the contents of the ReadMe.md file. In this file there is a hyperlink that calls to Azure and passes the azuredeploy.json file to it.



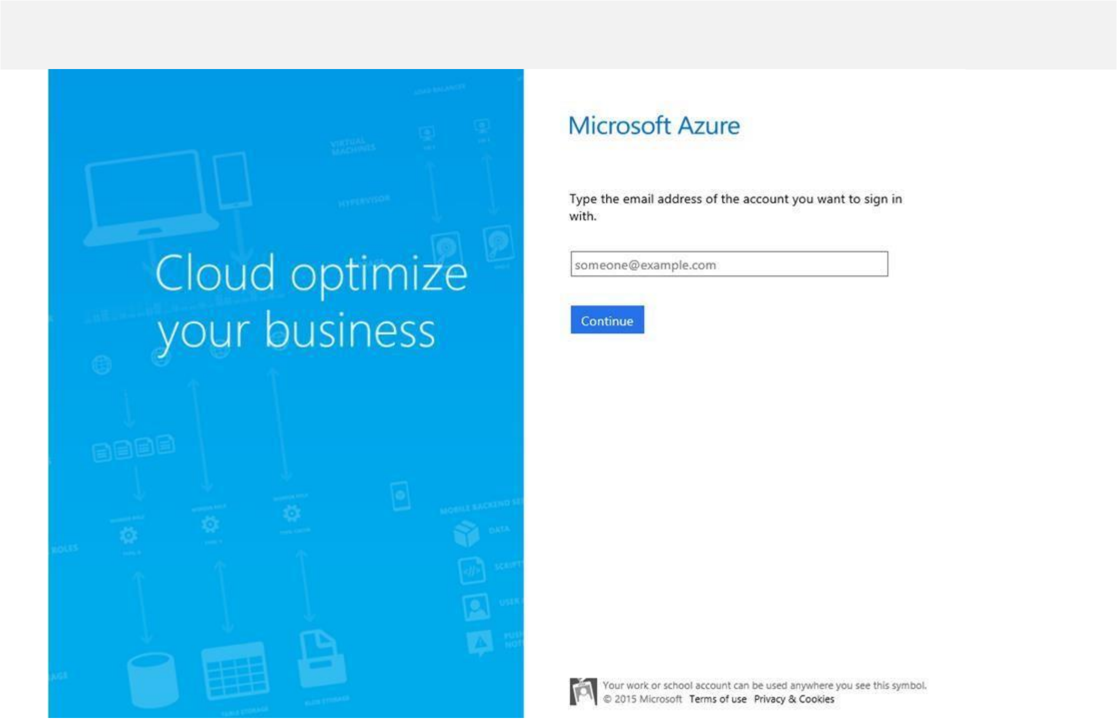
README.md

Deploy a Virtual Machine with CustomData

This template allows you to create a Virtual Machine with Custom Data. This template also deploys a Storage Account, Virtual Network, Public IP addresses and a Network Interface.

Task 2: Deploy Azure Resources through Azure Portal

3. Click the **Deploy to Azure** button. When prompted for your credentials, put in your Azure credentials.

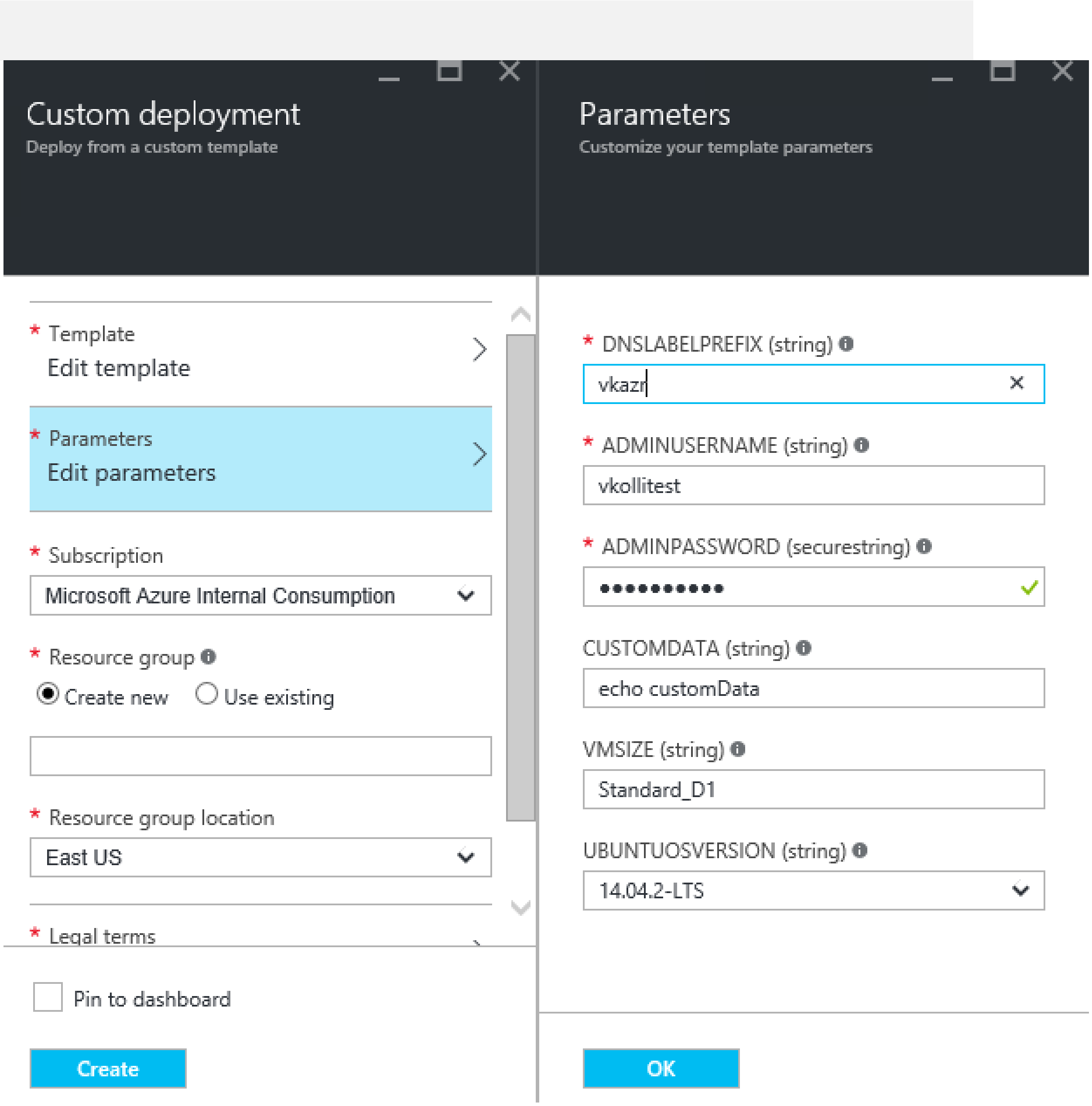


Type the email address of the account you want to sign in with.

Your work or school account can be used anywhere you see this symbol.
© 2015 Microsoft

4. Now you will be redirected to the new Azure Portal. The current window will be the **Edit Template** window. There are no edits we need to do so click the “**Save**” button.

5. We do have parameters that we need to fill out. On the **Custom Deployment** screen, click in the box that says PARAMETERS Edit Parameters



Custom deployment
Deploy from a custom template

* Template
Edit template

* Parameters
Edit parameters

* Subscription
Microsoft Azure Internal Consumption

* Resource group
● Create new ○ Use existing

* Resource group location
East US

* Legal terms

□ Pin to dashboard

Create

Parameters
Customize your template parameters

* DNSLABELPREFIX (string)
vkazr

* ADMINUSERNAME (string)
vkollitest

* ADMINPASSWORD (securestring)
••••••••••• ✓

CUSTOMDATA (string)
echo customData

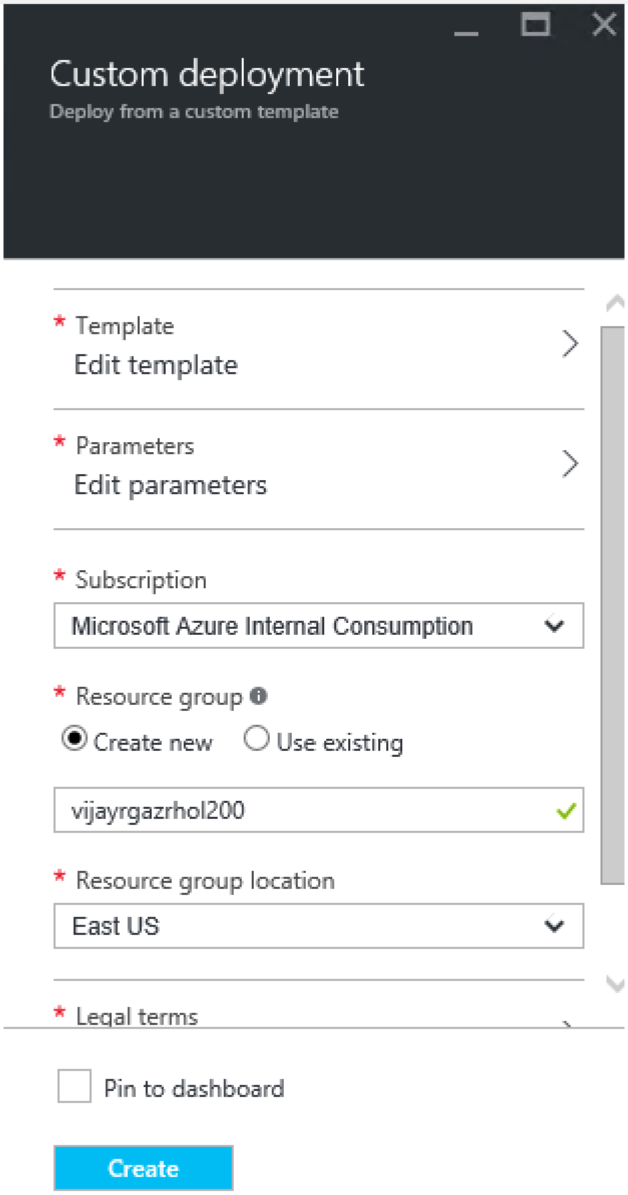
VMSIZE (string)
Standard_D1

UBUNTUOSVERSION (string)
14.04.2-LTS

OK

6. A new blade will open for the parameters. Fill out the parameters according to the right table above and then click “OK”.

7. Next click the “**Or create new**” link in the RESOURCE GROUP Select a resource group section.
8. Type in the resource group name <yourinitials>rg
9. Click **RESOURCE GROUP LOCATION** and choose EAST US



Custom deployment

Deploy from a custom template

* Template

Edit template

* Parameters

Edit parameters

* Subscription

Microsoft Azure Internal Consumption

* Resource group

● Create new ○ Use existing

vijayrgazrhol200 ✓

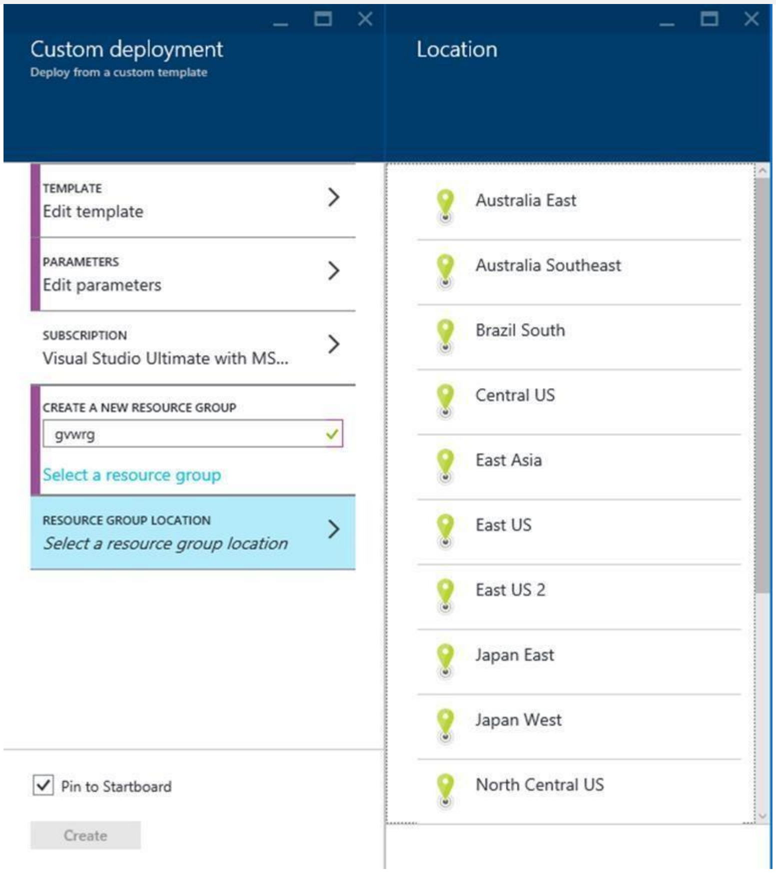
* Resource group location

East US

* Legal terms

□ Pin to dashboard

Revuiew Legal Terms



TEMPLATE
Edit template

PARAMETERS
Edit parameters

SUBSCRIPTION
Visual Studio Ultimate with MS...

CREATE A NEW RESOURCE GROUP
gvwrg ✓

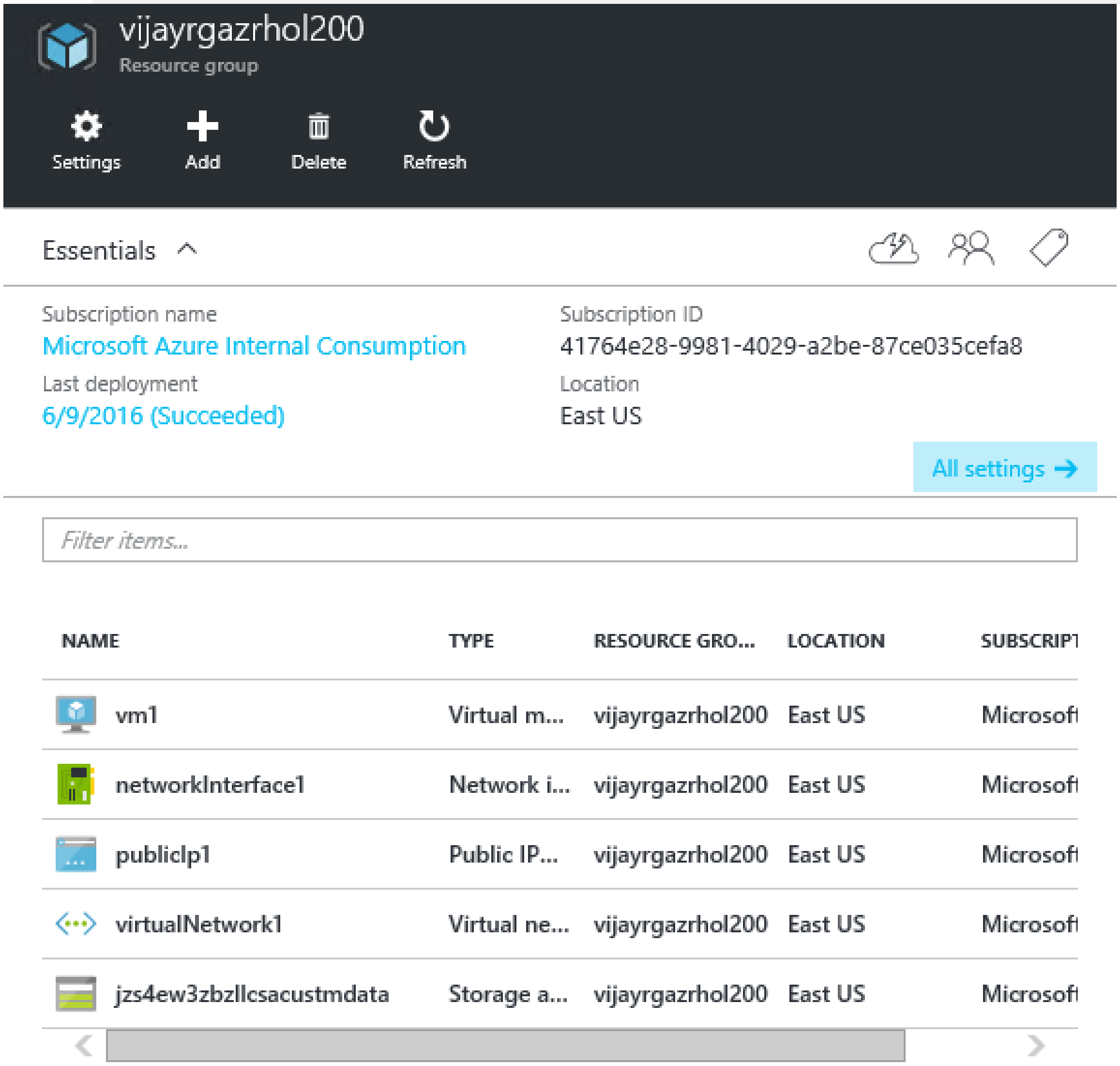
Select a resource group

RESOURCE GROUP LOCATION
Select a resource group location

10. Finally, click “**Create**”.

11. When done, should open with the successful creation of your VM.

If there is an error click redeploy or retry.



vijayrgazrhol200

Resource group

Subscription name
Microsoft Azure Internal Consumption

Subscription ID
41764e28-9981-4029-a2be-87ce035cefa8

Last deployment
6/9/2016 (Succeeded)

Location
East US

NAME	TYPE	RESOURCE GRO...	LOCATION	SUBSCRIPT
vm1	Virtual m...	vijayrgazrhol200	East US	Microsoft
networkInterface1	Network i...	vijayrgazrhol200	East US	Microsoft
publiclp1	Public IP...	vijayrgazrhol200	East US	Microsoft
virtualNetwork1	Virtual ne...	vijayrgazrhol200	East US	Microsoft
jzs4ew3zbzllcsacustmdata	Storage a...	vijayrgazrhol200	East US	Microsoft