Microsoft Cloud Workshop

Building First Azure Resource Manager Template

Hands-on lab step-by-step

October 2018

Contents

Building First Azure Resource Manager Template **1**

Abstract and learning objectives 1

Networking References 2

Creating the first resource using Visual Studio 3

Building First Azure Resource Manager Template

Abstract and learning objectives

Students will deploy an ARM template via Visual Studio through guided screen shots and instructions. Once the deployment is complete, the student will check on the deployment and ensure there are no errors.

Prepared for Arthur J. Gallagher

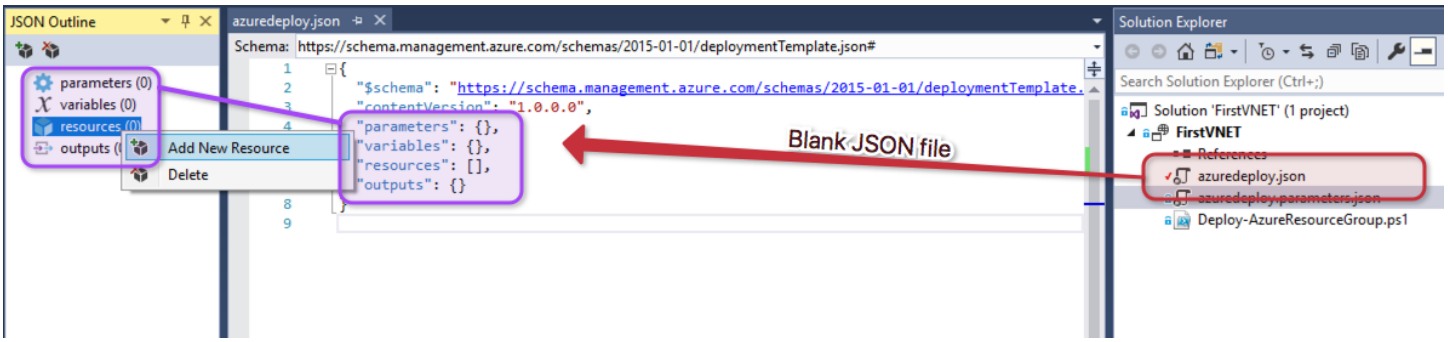
Networking References

1) Azure Virtual Network Overview
2) Azure Virtual Network FAQ
3) IP Addresses
    a. Public IP Addresses
    b. Internal IP Addresses
4) DNS
    a. Azure DNS
    b. Name Resolution for Azure VNets
5) Connectivity for Azure Virtual Networks
    a. Site-to-Site VPN
    b. VNet-to-VNet VPN
    c. Point-to-Site VPN
    d. Regional VNet Peering
    e. Global VNet Peering
    f. ExpressRoute Overview
6) Load Balancers
    a. Azure Load Balancer
    b. Azure Traffic Manager
    c. Azure Application Gateway
7) Network Security Strategies
    a. DMZ Between Azure and On-Premises
    b. DMZ Between Azure and the Internet
    c. Network Security Groups
    d. User Defined Routes
    e. Virtual Network Service Tunneling
    f. Web Application Firewall
    g. Service Endpoints
    h. Network Virtual Appliances
8) Monitoring
    a. Network Watcher
    b. Network Performance Monitor Overview & Solution
    c. ExpressRoute Monitor
    d. DNS Analytics
    e. Service Endpoint Monitoring
9) Infrastructure as Code
    a. Azure Quickstart Templates (searchable)
    b. Azure Quickstart Templates - GitHub
    c. Azure Resource Explorer
    d. ARM Template Reference Home
    e. ARM Template Functions Reference

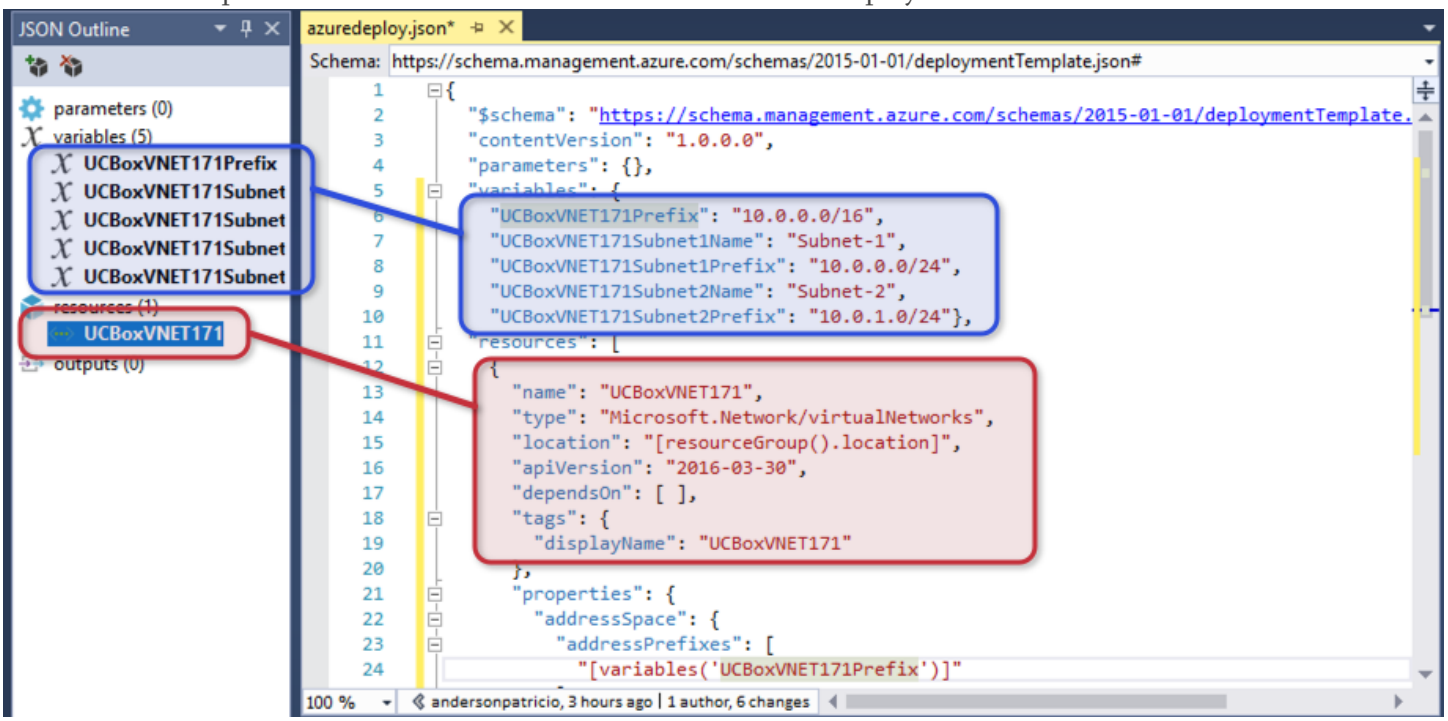
Creating the first resource using Visual Studio

A cloud engineer has several options to start an ARM template. We can get form a new deployment in Microsoft Azure Portal, some templates on the Internet (GitHub), or even from Visual Studio.

Since we are exploring Visual Studio, let's use the tool to create some basic resources to illustrate what we can do using Visual Studio. Right-click on **Resources** located on the JSON Outline tab, and click **Add New Resource**, from the list, select **Virtual Network** and label it. In our case we will use **UCBoxVNET171**, and click **OK**.

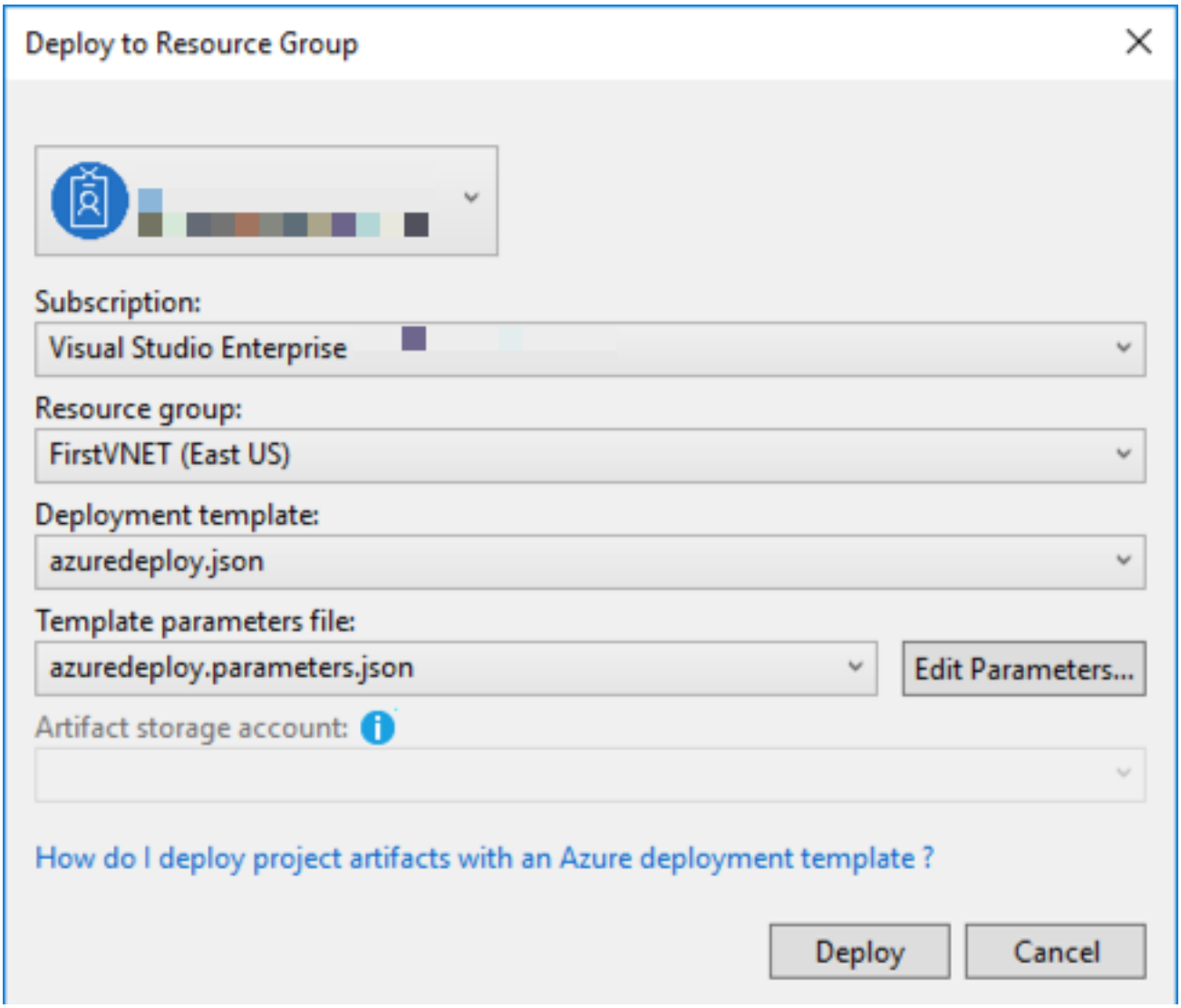


The result of that simple addition will be some code added to the **variables** and **resources** areas of the JSON file. By default, the subnets come with their IP ranges and names. We can change that on the JSON file to fit our environment. Keep in mind that information there is all we need to deploy a Virtual Network in Azure.

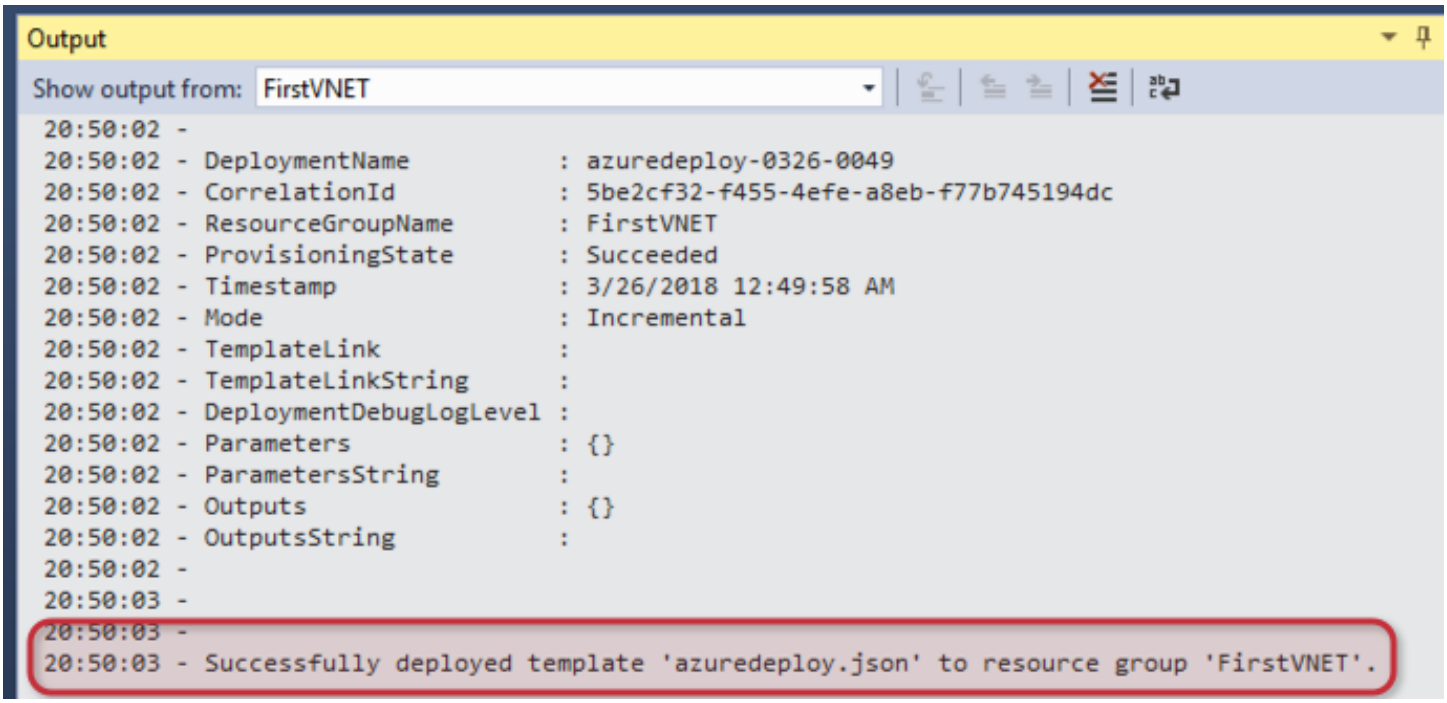


A simple way to deploy that JSON file in Azure is to right-click on the Project/Solution, located on the *Solution Explorer* tab, then click on **Deploy**, **New…**

In the new window, we have to define the subscription, the Resource Group (it can be an existent or create a new one), and define the files (the template, which is azuredeploy.json, and the parameters, which by default is azuredeploy.parameters.json). Click on **Deploy**.



The entire operation will be logged in the *output* area, and we can check the *azuredeploy.json* file was deployed successfully to the Resource Group that we created in the previous step.



Although a *Successfully deployed template* messages bring a lot of confidence, it is always good to check the results on the Azure Portal or using PowerShell. In the image below, we can validate that the Virtual Network (subnets are not being shown here but they were deployed) were created successfully in Microsoft Azure.