Azure ARM Template in Action

Earlier we discussed azure Arm Template and Terraform basics, now let's observe Azure resource Manager (ARM) template in action.

So let's explore it further.

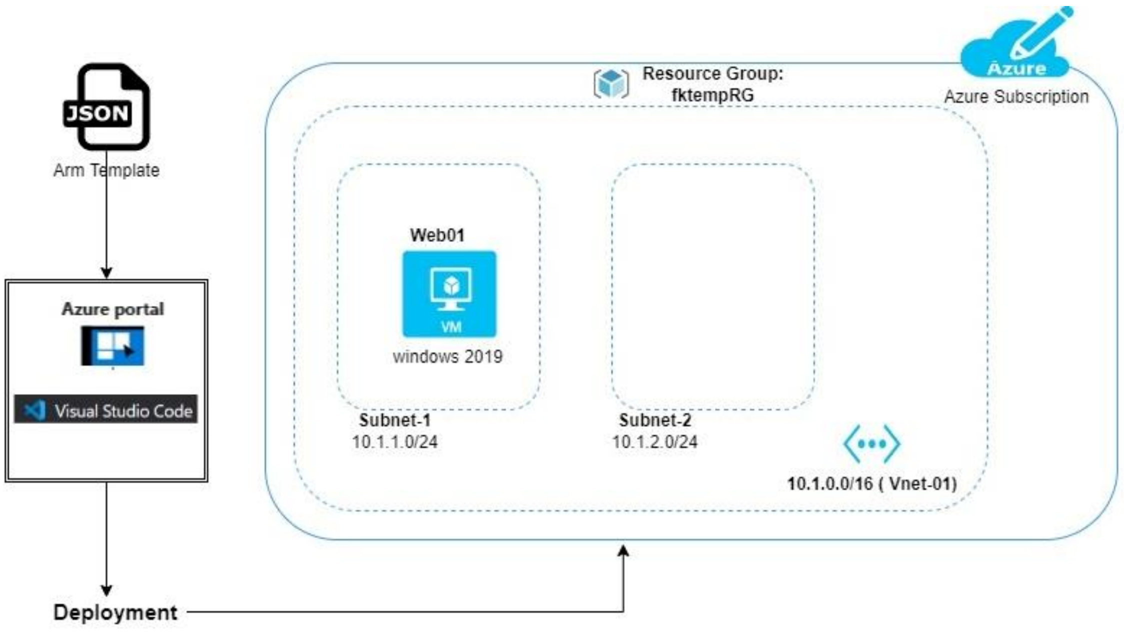
ARM Templates

We discussed earlier that ARM templates are a form of infrastructure as code for azure solutions, a concept where you define the infrastructure that needs to be deployed. The template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project.

We also covered the various Template components (overview) and options from where the template can be deployed.

In this section we will use Azure portal & Visual studio code for template creation and editing. We will use azure portal for template deployment.

We will start with a basic setup, the following Azure resources will be deployed using ARM Template.



Get the Template file (JSON) from below link.

https://tinyurl.com/fkarm1vmtempl

feel free to use and modify the template as per your requirement.

Using the Template, the following resources will be deployed in the created or specified Azure Resource Group.

1. 1 Virtual Machine (Windows 2019- Datacenter) (Vm size- DS2_V2).
2. 1 Virtual Network (10.1.0.0/16) and with 2 associated subnets (10.1.1.0/24 & 10.1.2.0/24).

We can create the Json template by the following available tools.

- **Azure Portal:**

  Method 1: Deploy a resources using Azure portal

  - Access the deployed Template from the following
    - **Navigate** to your created Resource Group.
    - **Click** on Deployments
    - **Click** on the deployment Name of the deployed resource.
    - On the Left pane **click** on template.
    - **Access** the template in Json format.
    - You have an option to **Download** or **Deploy** the template.

  Method 2: Create template using Azure portal Custom Template.
    - On the azure portal, **search** for templates and then **click** Deploy custom template.
    - **Deploy** a quick start template using an existing template (Community created)
      OR
    - **Build** your own template in the editor.

- **Visual Studio Code (Editor):**
  The Azure Resource Manager Tools for Visual Studio Code provide language support, resource snippets, and resource autocompletion. These tools help create and validate Azure Resource Manager templates (ARM templates).

    - **Download** Visual Studio Code (Windows / Mac/ Linux) Platforms.
      https://code.visualstudio.com/Download

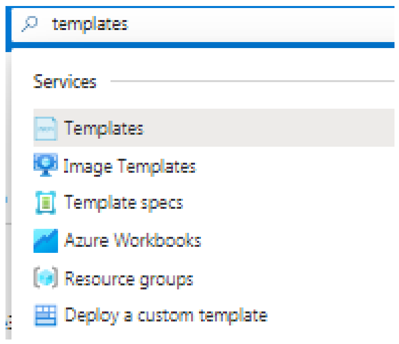
    - **Install** the Azure Resource Manager tools Extensions on your VS code console.
      https://marketplace.visualstudio.com/items?itemName=msazurermtools.azurerm-vscode-tools

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

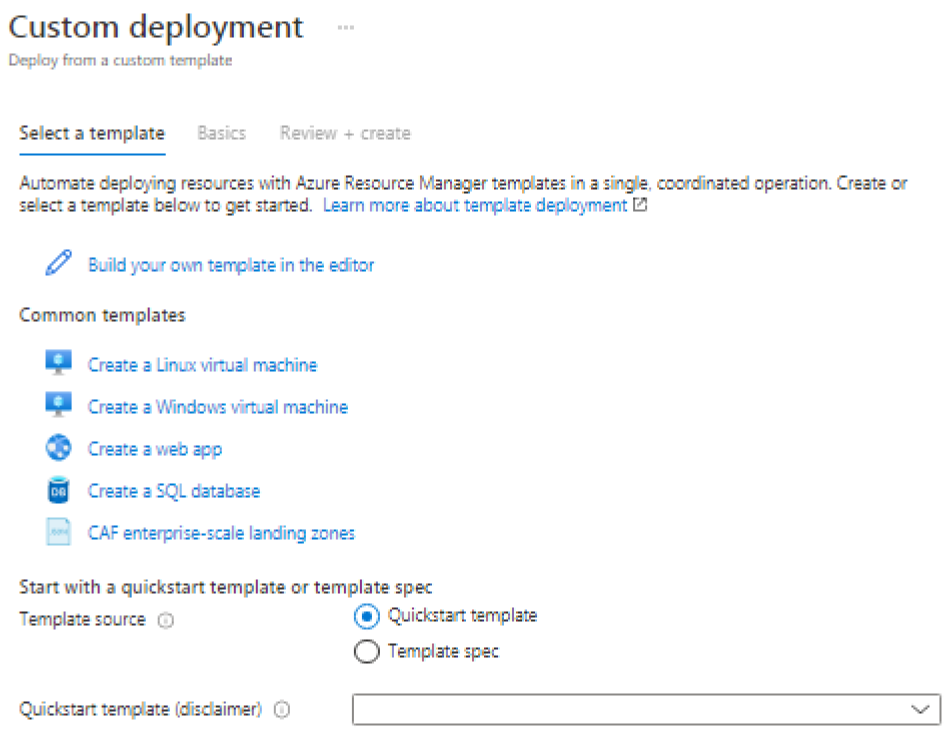
Create Azure portal – Custom template

First Create a Resource group for this Lab in my case **FktempRG**

On the azure portal, search for templates and then **click** Deploy custom template



You can deploy a quick start template using an existing template (Community created).


Custom deployment
Deploy from a custom template

Select a template | Basics | Review + create

Automate deploying resources with Azure Resource Manager templates in a single, coordinated operation. Create or select a template below to get started. Learn more about template deployment

Build your own template in the editor

Common templates

Create a Linux virtual machine
Create a Windows virtual machine
Create a web app
Create a SQL database
CAF enterprise-scale landing zones

Start with a quickstart template or template spec

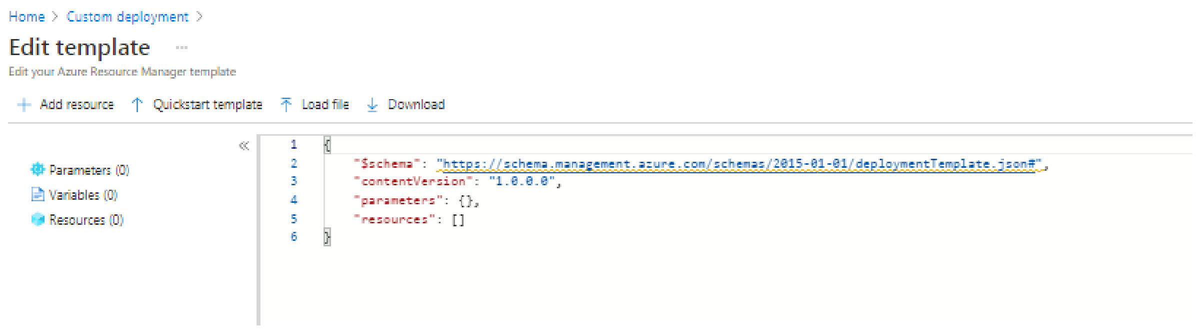
Template source: [✓ Quickstart template] Template spec

Quickstart template (disclaimer)

On the template section Click **Build your own template in the editor**

We will start with JSON Templates.

Note: Arm templates does not support automatic versioning (versioning are maintained manually)

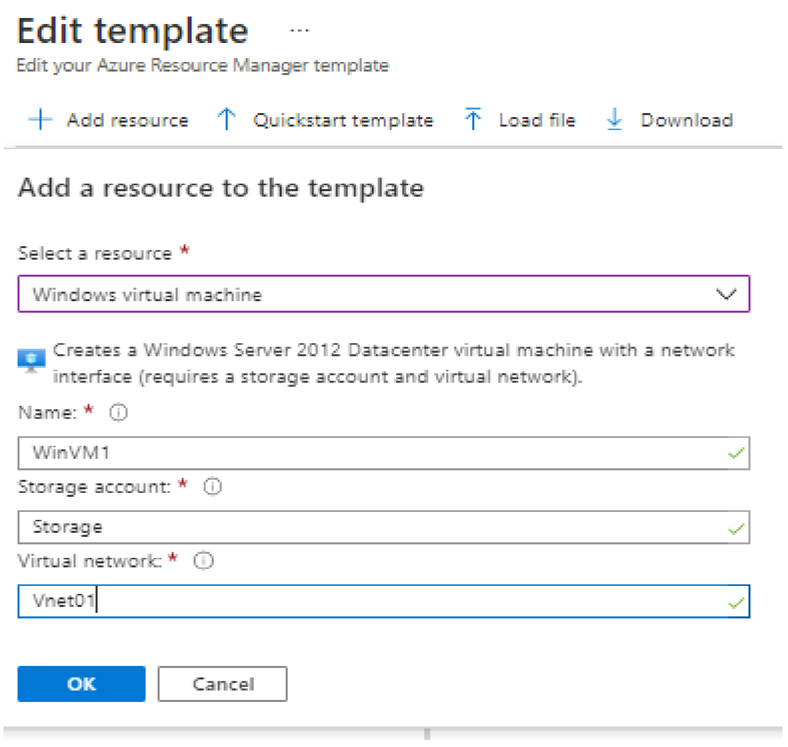


```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "resources": []
}
```

Now lets Add a resource to our Template.

On the template editor, Click on Add a resource

Select **Windows Virtual Machine** as your resource, Enter the following fields.


Edit template
Edit your Azure Resource Manager template
Add resource
Quickstart template
Load file
Download
Add a resource to the template
OK
Cancel

Select a resource *: Windows virtual machine

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

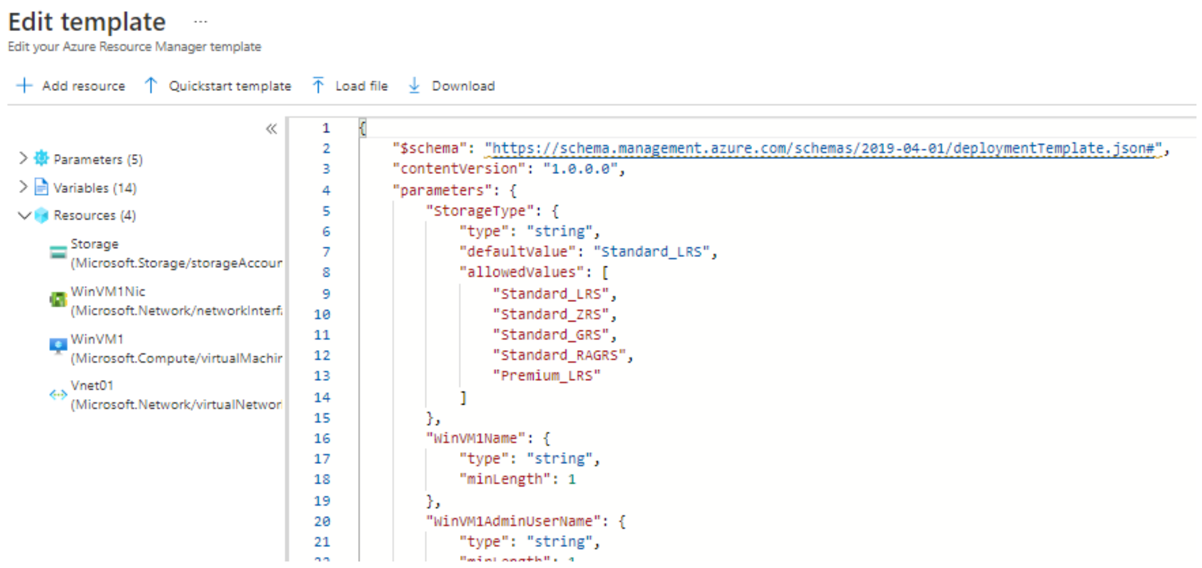
Name: *: WinVM1

Storage account: *: Storage

Virtual network: *: Vnet01

Click Ok

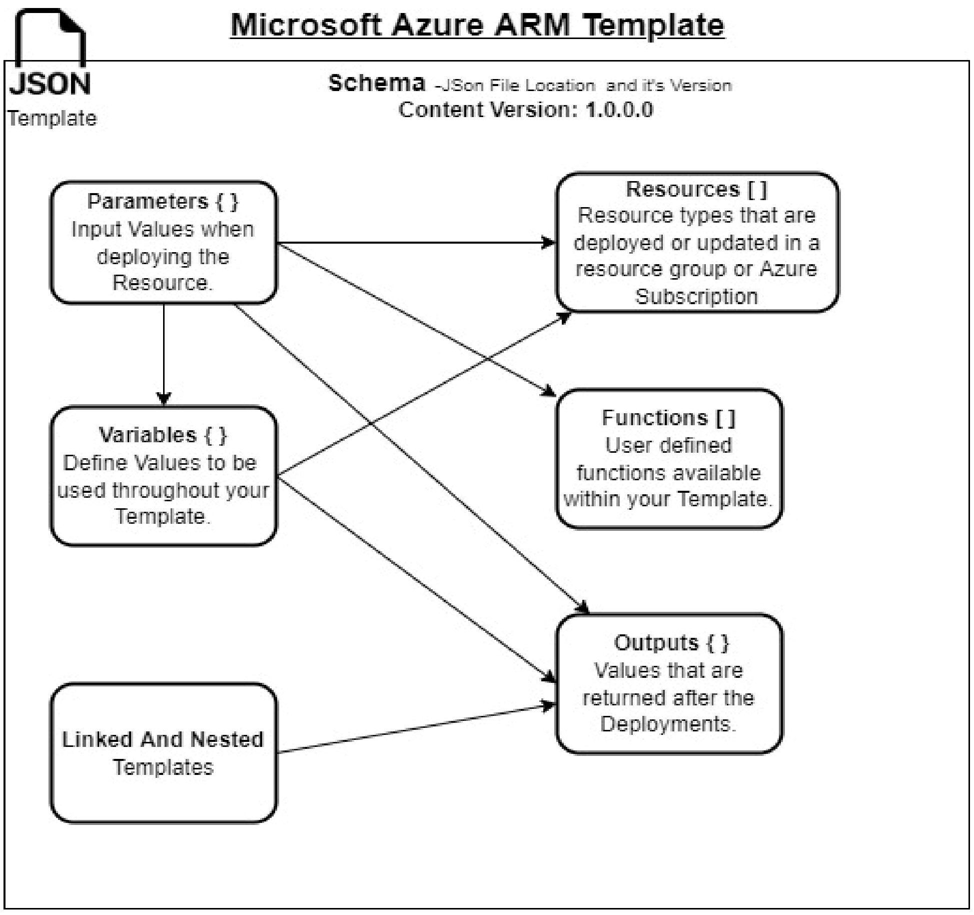
The template is added to your editor providing you the following details.



Edit template ···
Edit your Azure Resource Manager template
Add resource
Quickstart template
Load file
Download
Parameters (5)
Variables (14)
Resources (4)
Storage
(Microsoft.Storage/storageAccour
WinVM1Nic
(Microsoft.Network/networkInterf
WinVM1
(Microsoft.Compute/virtualMachir
Vnet01
(Microsoft.Network/virtualNetwor

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "StorageType": {
            "type": "string",
            "defaultValue": "Standard_LRS",
            "allowedValues": [
                "Standard_LRS",
                "Standard_ZRS",
                "Standard_GRS",
                "Standard_RAGRS",
                "Premium_LRS"
            ]
        },
        "WinVM1Name": {
            "type": "string",
            "minLength": 1
        },
        "WinVM1AdminUserName": {
            "type": "string",
```

Now Let’s Understand an Azure ARM Template.



Microsoft Azure ARM Template
JSON
Template
Schema -JSon File Location and it's Version
Content Version: 1.0.0.0
Parameters { } Input Values when deploying the Resource.
Resources [ ] Resource types that are deployed or updated in a resource group or Azure Subscription
Variables { } Define Values to be used throughout your Template.
Functions [ ] User defined functions available within your Template.
Outputs { } Values that are returned after the Deployments.
Linked And Nested Templates

I am providing important **reference links** to understand ARM templates in detail.

JSON Structure:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/syntax

Parameters:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/parameters

Variables:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/variables

User-Defined Functions:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/user-defined-functions

Resources:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/syntax#resources

Outputs:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/outputs?tabs=azure-powershell

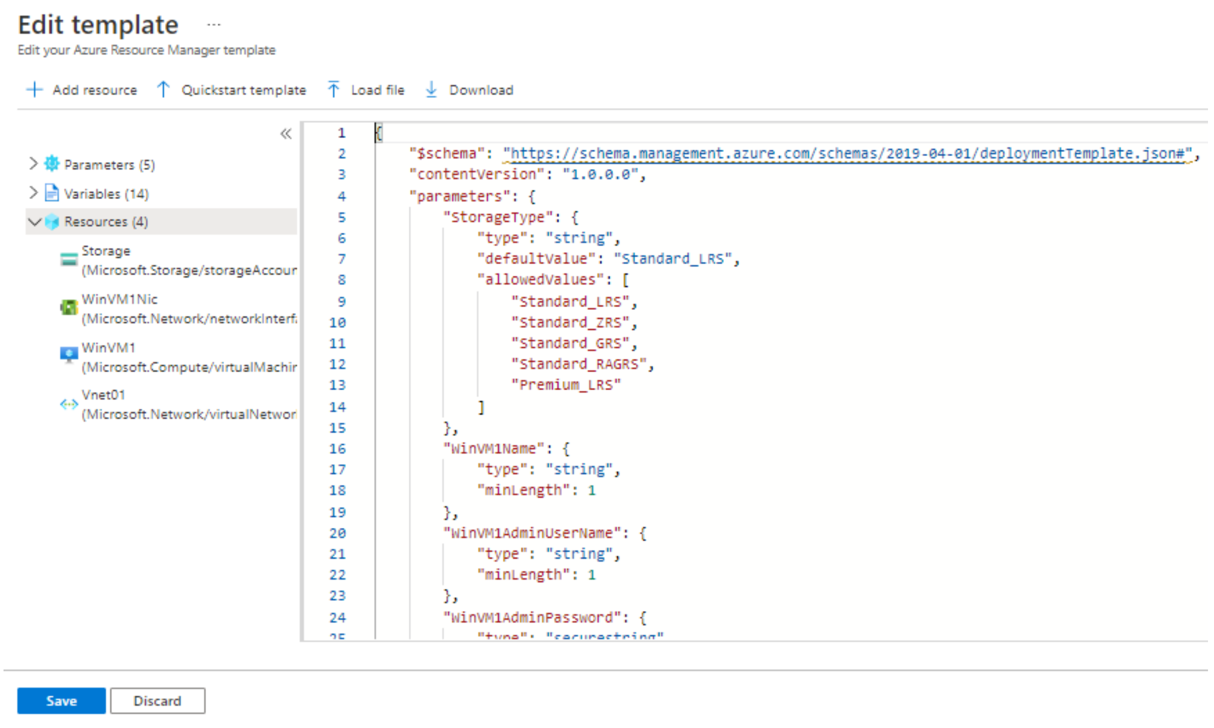
Deploy resources with ARM templates and Azure portal

Deploying Azure resources by using the Azure portal usually involves two steps:

- Create a resource group.
- Deploy resources to the resource group.

Also, you can create a customized ARM template to deploy Azure resources.

Considering our scenario mentioned above and the custom template we created matching our layout.


Edit template
Edit your Azure Resource Manager template
Add resource
Quickstart template
Load file
Download
Parameters (5)
Variables (14)
Resources (4)
Storage
(Microsoft.Storage/storageAccour
WinVM1Nic
(Microsoft.Network/networkInterf
WinVM1
(Microsoft.Compute/virtualMachir
Vnet01
(Microsoft.Network/virtualNetwor
Save
Discard

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "StorageType": {
            "type": "string",
            "defaultValue": "Standard_LRS",
            "allowedValues": [
                "Standard_LRS",
                "Standard_ZRS",
                "Standard_GRS",
                "Standard_RAGRS",
                "Premium_LRS"
            ]
        },
        "WinVM1Name": {
            "type": "string",
            "minLength": 1
        },
        "WinVM1AdminUserName": {
            "type": "string",
            "minLength": 1
        },
        "WinVM1AdminPassword": {
            "type": "securestring"
```

Now to match our solution Layout , I did some changes in the Template.

Added **2019-Datacenter** Fields in the OS Version which were missing earlier.

```
        "WinVm1WindowsOSVersion": {
            "type": "string",
            "defaultValue": "2019-Datacenter",
            "allowedValues": [
                "2008-R2-SP1",
                "2012-Datacenter",
                "2012-R2-Datacenter",
                "2019-Datacenter",
                "Windows-Server-Technical-Preview"
            ]
```

Defined hard-coded variable for the Windows Vm size.

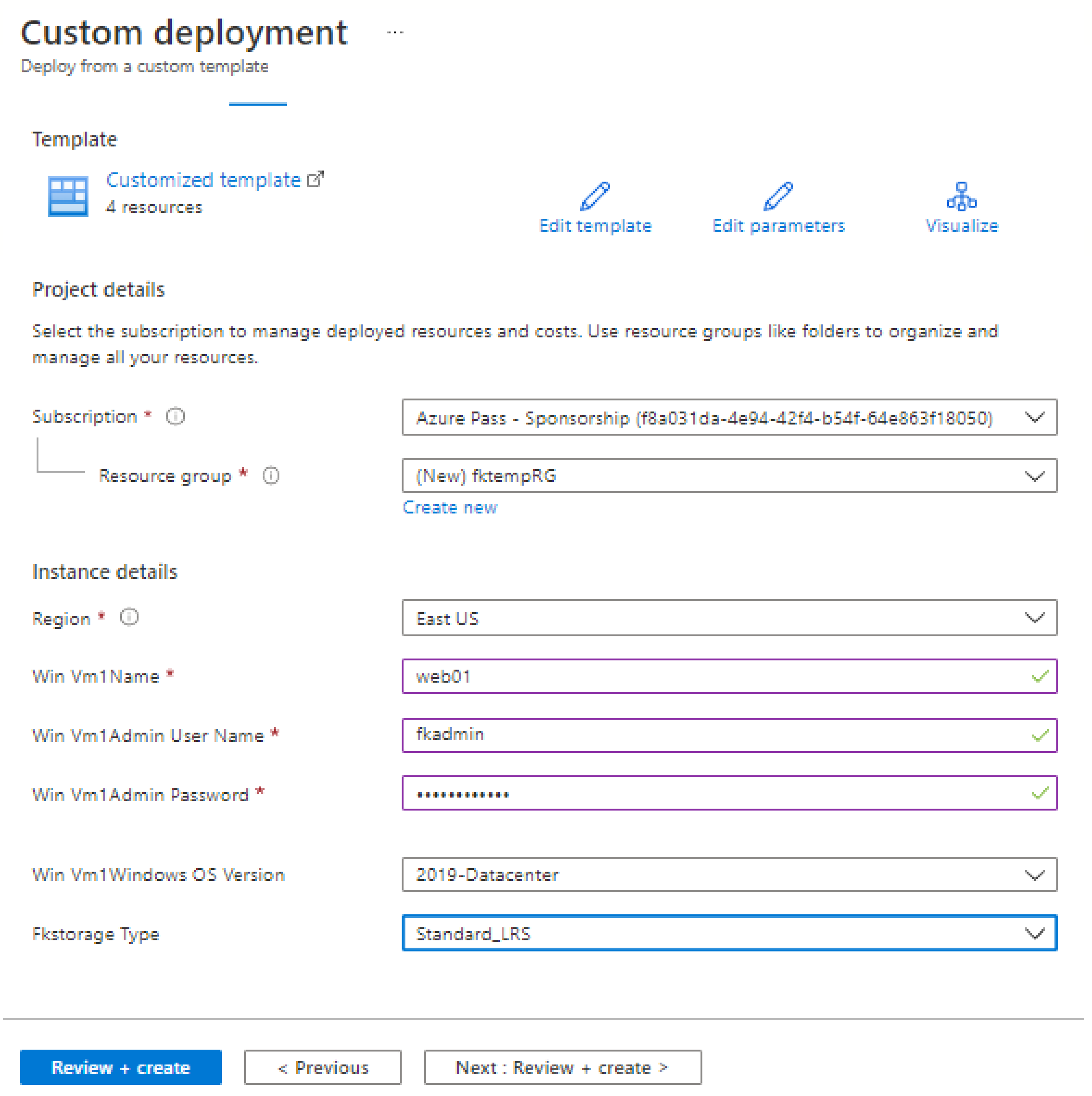
```
"WinVm1VmSize": "Standard_DS2_v2",
```

Defined hard-coded variables for Vnet and subnet prefix.

```
"Vnet01Prefix": "10.1.0.0/16",
        "Vnet01Subnet1Name": "Subnet-1",
        "Vnet01Subnet1Prefix": "10.1.1.0/24",
        "Vnet01Subnet2Name": "Subnet-2",
        "Vnet01Subnet2Prefix": "10.1.2.0/24"
```

After the above changes , On the Template Editor **Click Save.**

Portal prompts you to input the defined parameter values for your template.



Custom deployment ...

Deploy from a custom template

Template

Customized template

4 resources

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Azure Pass - Sponsorship (f8a031da-4e94-42f4-b54f-64e863f18050)
Resource group *	(New) fktempRG

Create new

Instance details

Region *	East US
Win Vm1Name *	web01
Win Vm1Admin User Name *	fkadmin
Win Vm1Admin Password *	•••••••••••••
Win Vm1Windows OS Version	2019-Datacenter
Fkstorage Type	Standard_LRS

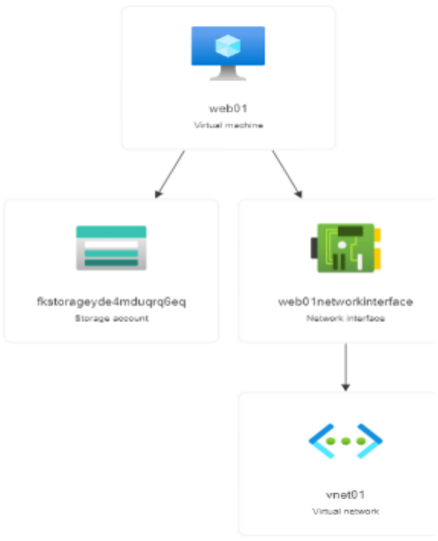
Then **click** Review + create

The template is saved and deployed on Azure cloud.

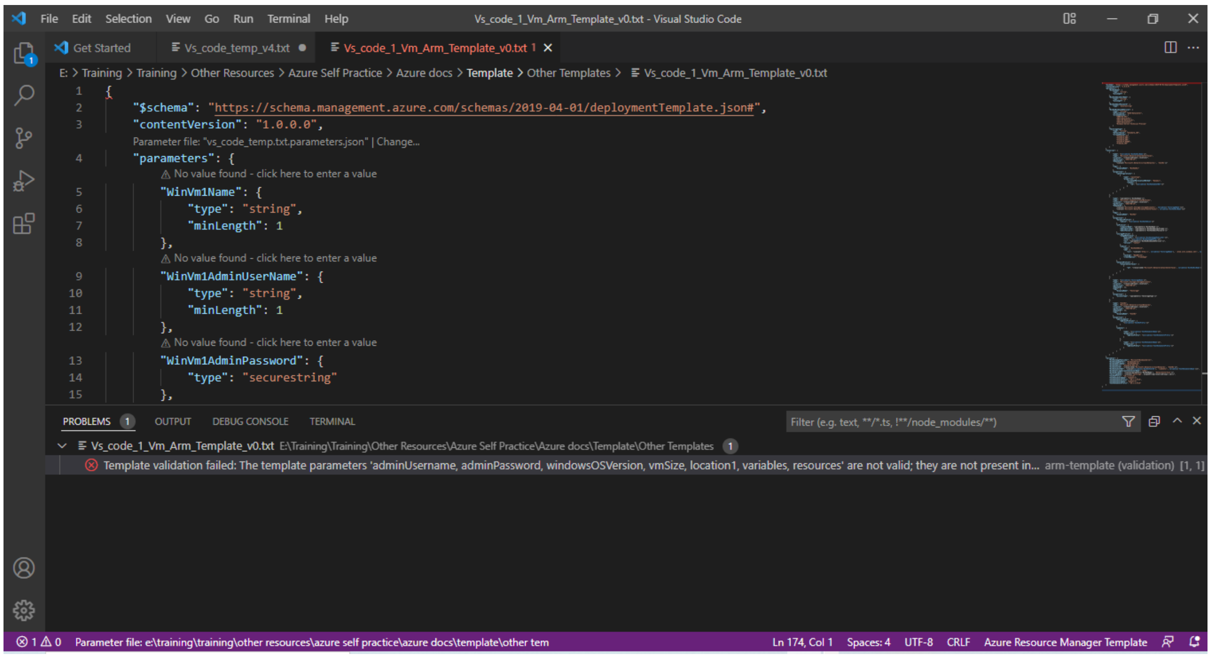
You can download this template or copy the Json script on wordpad and notepad providing a logical name.

You can always create or use the script via **Azure Templates**.

Resource Visualizer Image for the Deployed resources via Arm Template.



Visual Studio Code Screen Shot


File Edit Selection View Go Run Terminal Help
Vs_code_1_Vm_Arm_Template_v0.txt - Visual Studio Code
Get Started
Vs_code_temp_v4.txt
Vs_code_1_Vm_Arm_Template_v0.txt 1
E: > Training > Training > Other Resources > Azure Self Practice > Azure docs > Template > Other Templates > Vs_code_1_Vm_Arm_Template_v0.txt

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    Parameter file: "vs_code_temp.txt.parameters.json" | Change...
    "parameters": {
        ⚠ No value found - click here to enter a value
        "WinVm1Name": {
            "type": "string",
            "minLength": 1
        },
        ⚠ No value found - click here to enter a value
        "WinVm1AdminUserName": {
            "type": "string",
            "minLength": 1
        },
        ⚠ No value found - click here to enter a value
        "WinVm1AdminPassword": {
            "type": "securestring"
        },
```

```
Vs_code_1_Vm_Arm_Template_v0.txt E:\Training\Training\Other Resources\Azure Self Practice\Azure docs\Template\Other Templates 1
ⓧ Template validation failed: The template parameters 'adminUsername, adminPassword, windowsOSVersion, vmSize, location1, variables, resources' are not valid; they are not present in... arm-template (validation) [1, 1]
```

```
ⓧ 1 ⚠ 0   Parameter file: e:\training\training\other resources\azure self practice\azure docs\template\other tem   Ln 174, Col 1   Spaces: 4   UTF-8   CRLF   Azure Resource Manager Template
```

I found Syntax troubleshooting difficult using Azure template editor, It is much easier to trouble shoot using Visual Studio Code, as it states the problems in your template for addressing the same.

Under Visual studio Code **select** the language identifier as **Arm-Template** as language identifier when working with azure templates.

Learn More about **Azure templates** using the following Reference Links.

Supported Data Types in Arm Template

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/data-types

Azure Resource Manager Deployment Modes:

When deploying your resources, you specify that the deployment is either an incremental update or a complete update. The difference between these two modes is how Resource Manager handles existing resources in the resource group that aren't in the template.

- **Complete Mode**
- **Incremental Mode**

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deployment-modes

Export Arm Templates via Azure Portal

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/export-template-portal

Test Templates via Arm toolkit

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/test-toolkit

Template Spec

A template spec is a resource type for storing an Azure Resource Manager template (ARM template) in Azure for later deployment.

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-specs?tabs=azure-powershell

Continuous Integration (CI) / Continuous Deployment (CD)

Integrate Arm Templates with Azure Pipelines

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/add-template-to-azure-pipelines

Azure Template Best Practices

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/best-practices

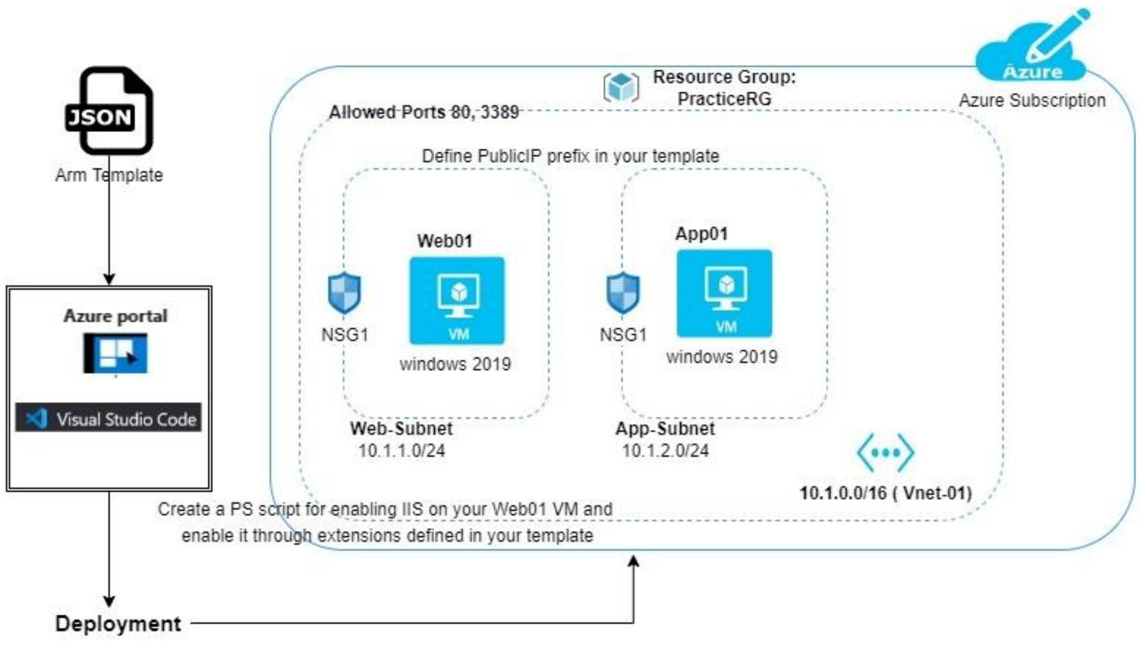
FAQ's about ARM templates

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/frequently-asked-questions

Overall Azure Template Documentation

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/

Practice Scenario: Practice the below scenario using Azure Templates.



This completes are topic on Azure templates, our next topic will be on Azure PaaS solutions and GIT.

Do follow me on LinkedIn

Faiz Kazi

https://www.linkedin.com/in/faiz-kazi-13675b4a/