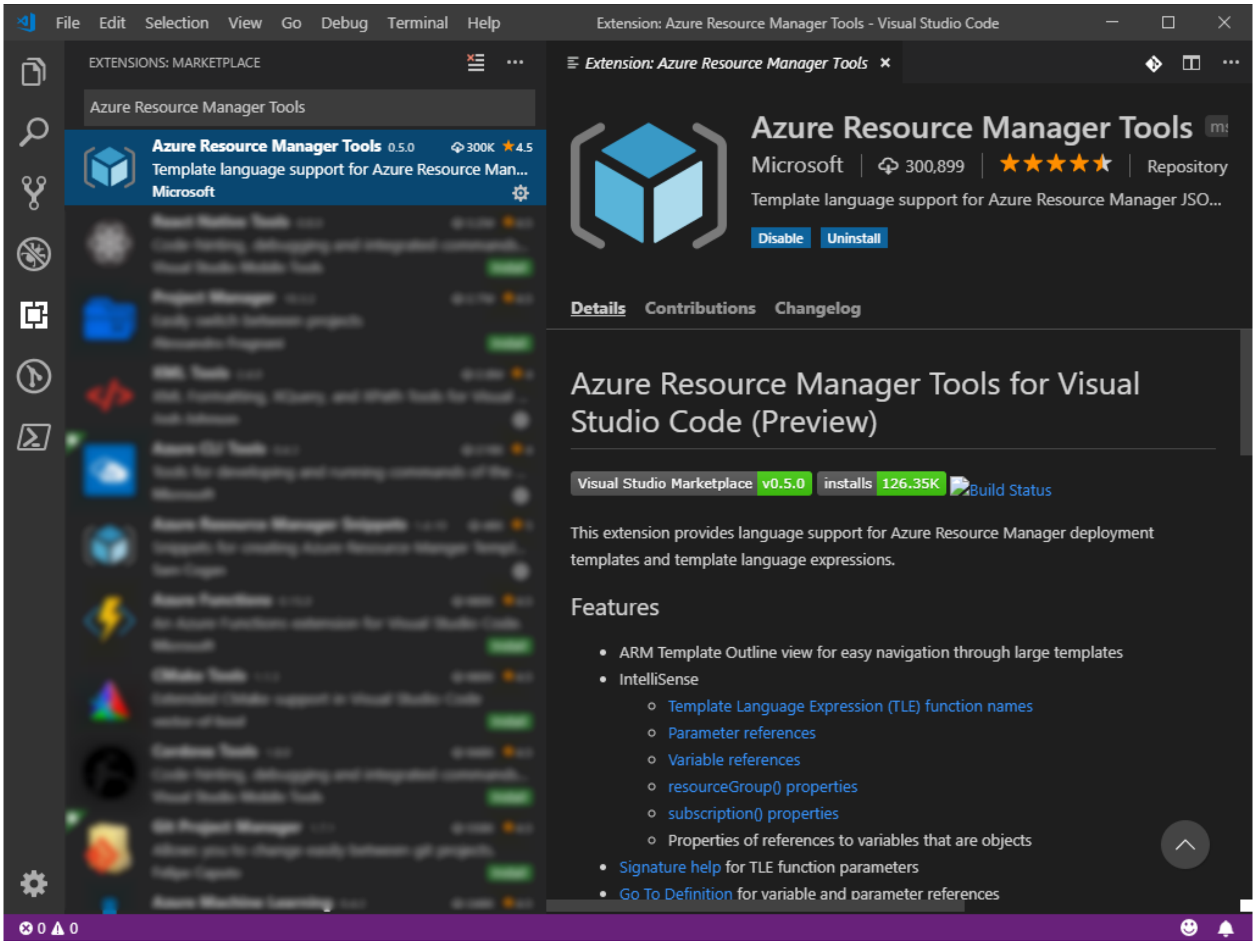
Module 4 - Lesson 2

Install Extensions

1. Launch Visual Studio Code
2. Open the Extensions view
   a. Windows/Linux: `Ctrl+Shift+X`
   b. macOS: `Shift+⌘+X`
3. If the Azure Resource Manager Tools extension is not installed, install it.



File
Edit
Selection
View
Go
Debug
Terminal
Help
Extension: Azure Resource Manager Tools - Visual Studio Code
EXTENSIONS: MARKETPLACE
Azure Resource Manager Tools
Azure Resource Manager Tools 0.5.0 300K ★4.5
Template language support for Azure Resource Man...
Microsoft
Extension: Azure Resource Manager Tools ×
Azure Resource Manager Tools
Microsoft | 300,899 | ★★★★★ | Repository
Template language support for Azure Resource Manager JSO...
Disable
Uninstall
Details
Contributions
Changelog
Visual Studio Marketplace v0.5.0
installs 126.35K
Build Status
⊗ 0 ⚠ 0

Azure Resource Manager Tools for Visual Studio Code (Preview)

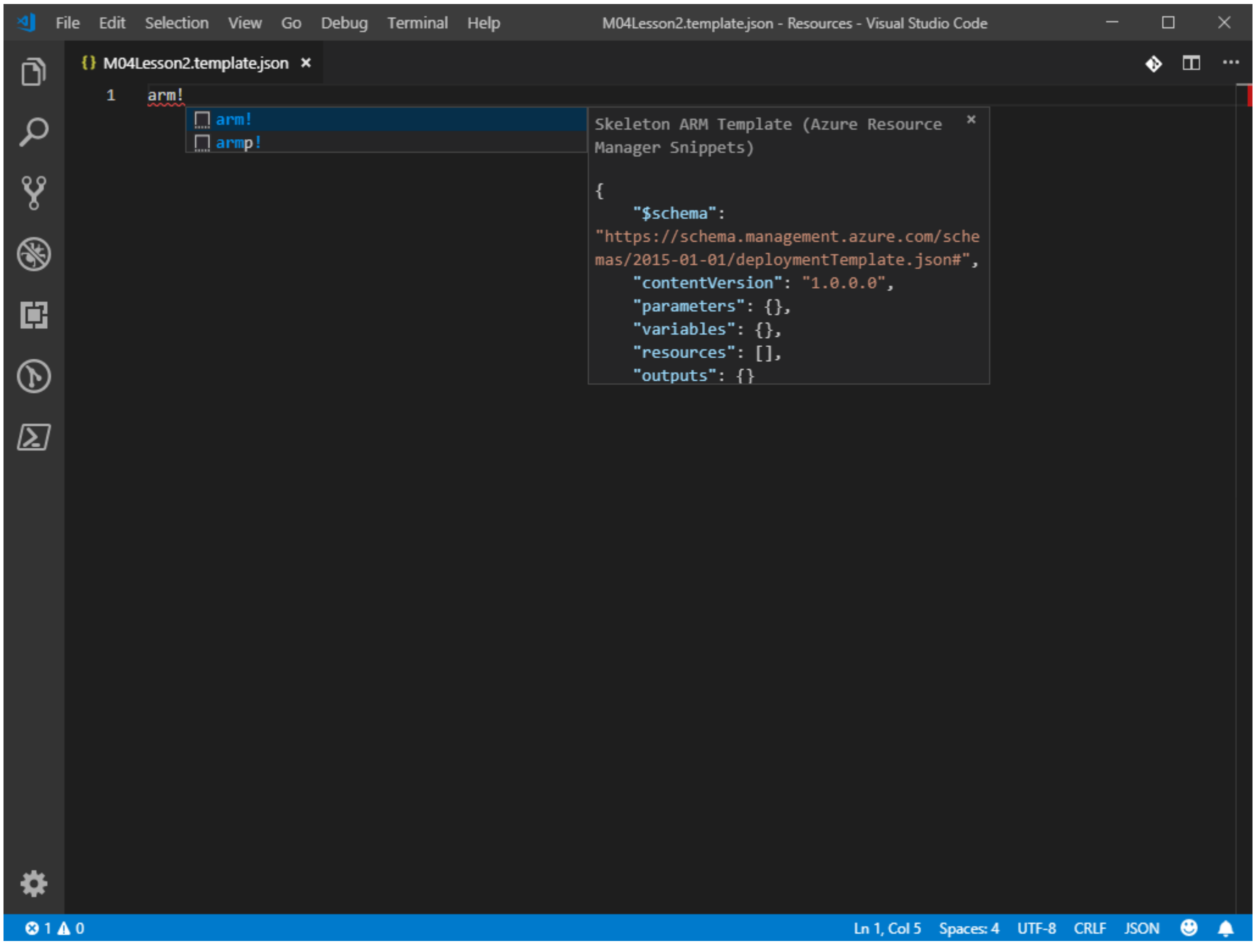
This extension provides language support for Azure Resource Manager deployment templates and template language expressions.

Features

- ARM Template Outline view for easy navigation through large templates
- IntelliSense
  - Template Language Expression (TLE) function names
  - Parameter references
  - Variable references
  - resourceGroup() properties
  - subscription() properties
  - Properties of references to variables that are objects
- Signature help for TLE function parameters
- Go To Definition for variable and parameter references

Create a new ARM template file

1. Open `C:\Lab_Files\M04` in Visual Studio Code and create a subfolder named `S02`
2. Create a new file in `C:\Lab_Files\M04\S02` named `M04Lesson2.template.json` and open the file.
3. Type `arm!` and press `Enter` to insert the ARM template skeleton code snippet

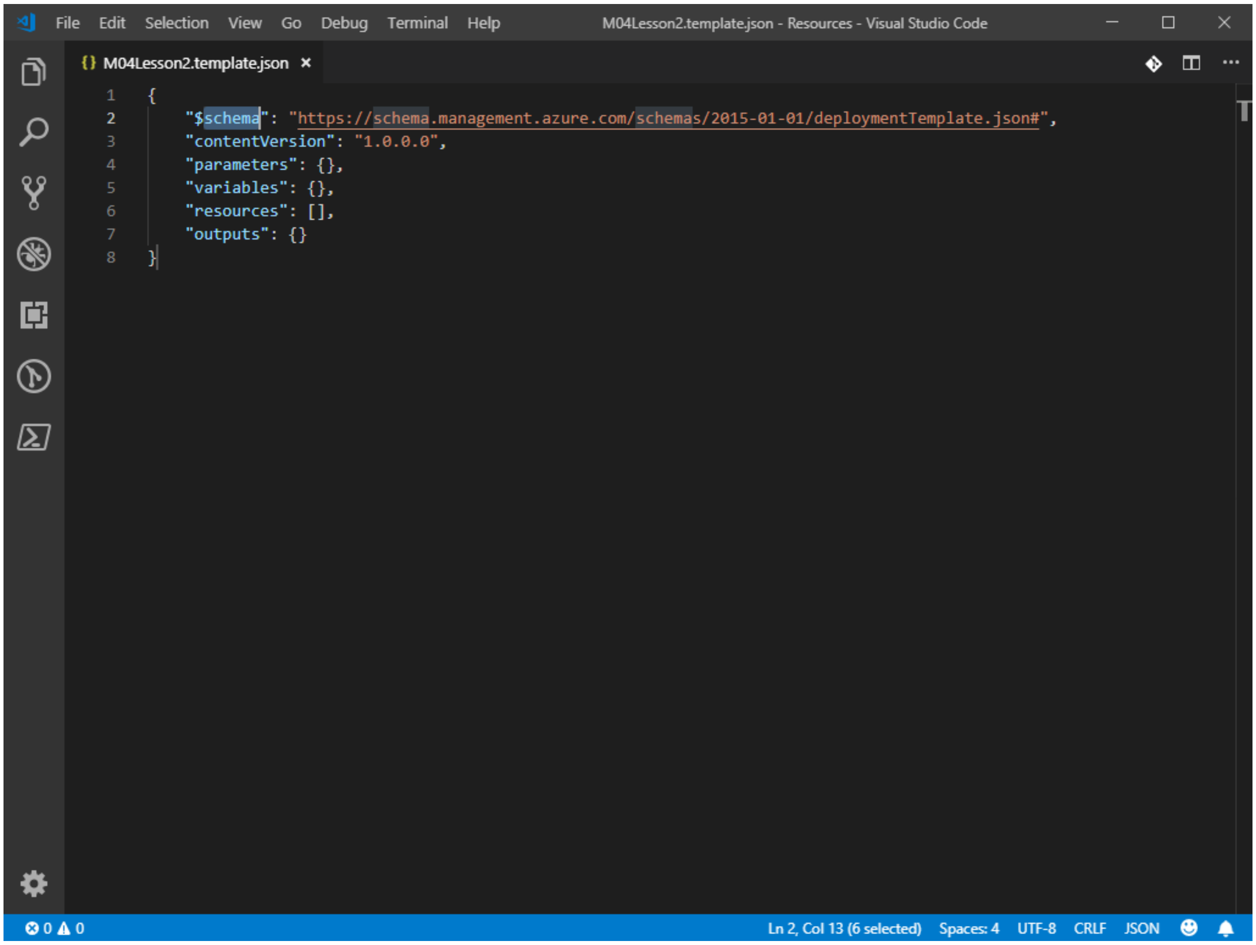


```
arm!
```

```
arm!
armp!
```

Skeleton ARM Template (Azure Resource Manager Snippets)

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



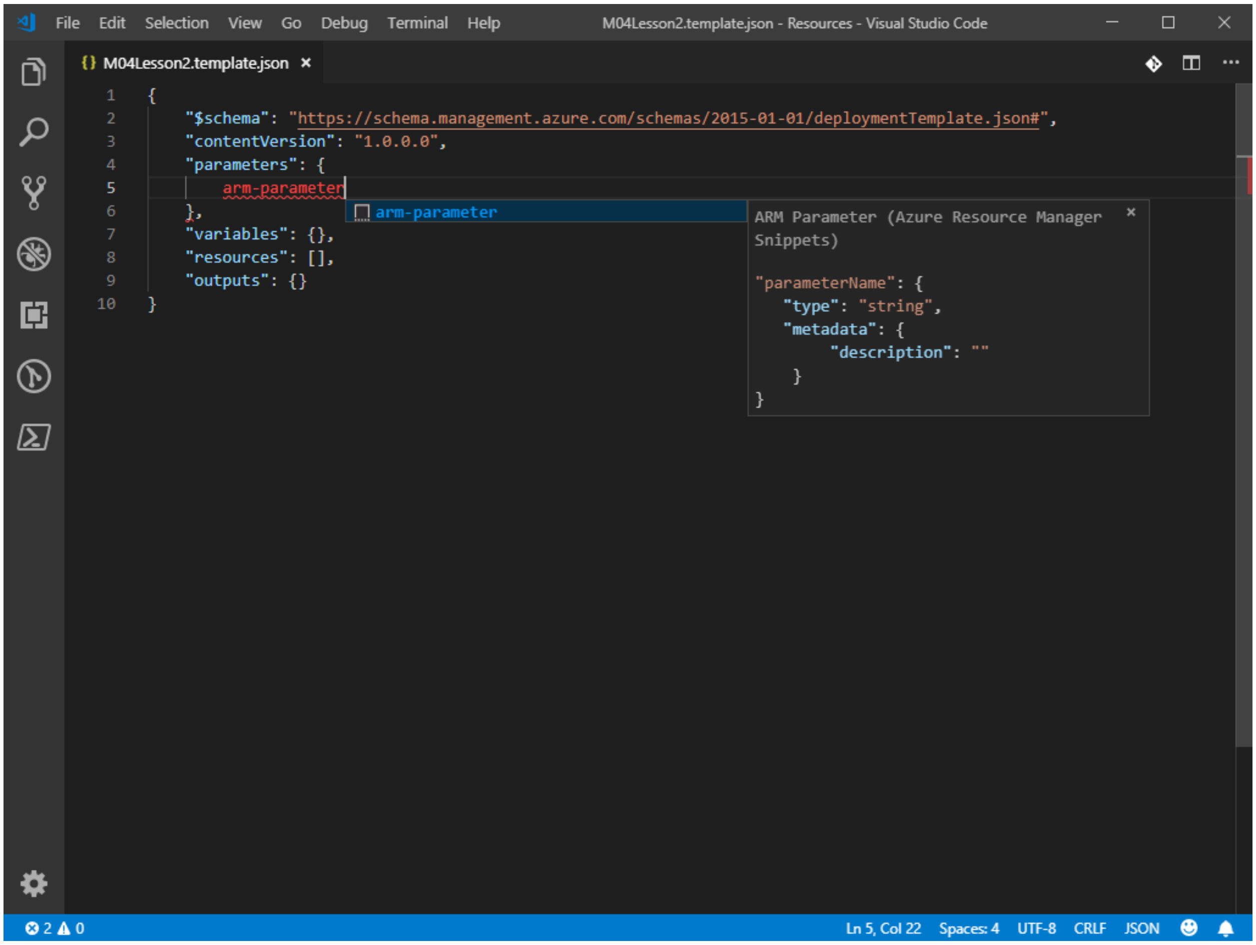
```json
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "variables": {},
    "resources": [],
    "outputs": {}
}
```

4. **NOTE:** If IntelliSense does not popup or if hitting enter does not insert the block of code, you can trigger IntelliSense by pressing `Ctrl+Space` on Windows/Linux or `^+Space` on macOS

Add parameters to the ARM template

1. Move your cursor in between the `{}` brackets on the line containing `"parameters": {},` and press `Enter` to create a new line

2. Type `arm-param` and press `Enter` to insert a new parameter snippet. (**NOTE:** Depending on the version of the snippet extension, the snippet may be referenced by another name such as `arm-parameter`)

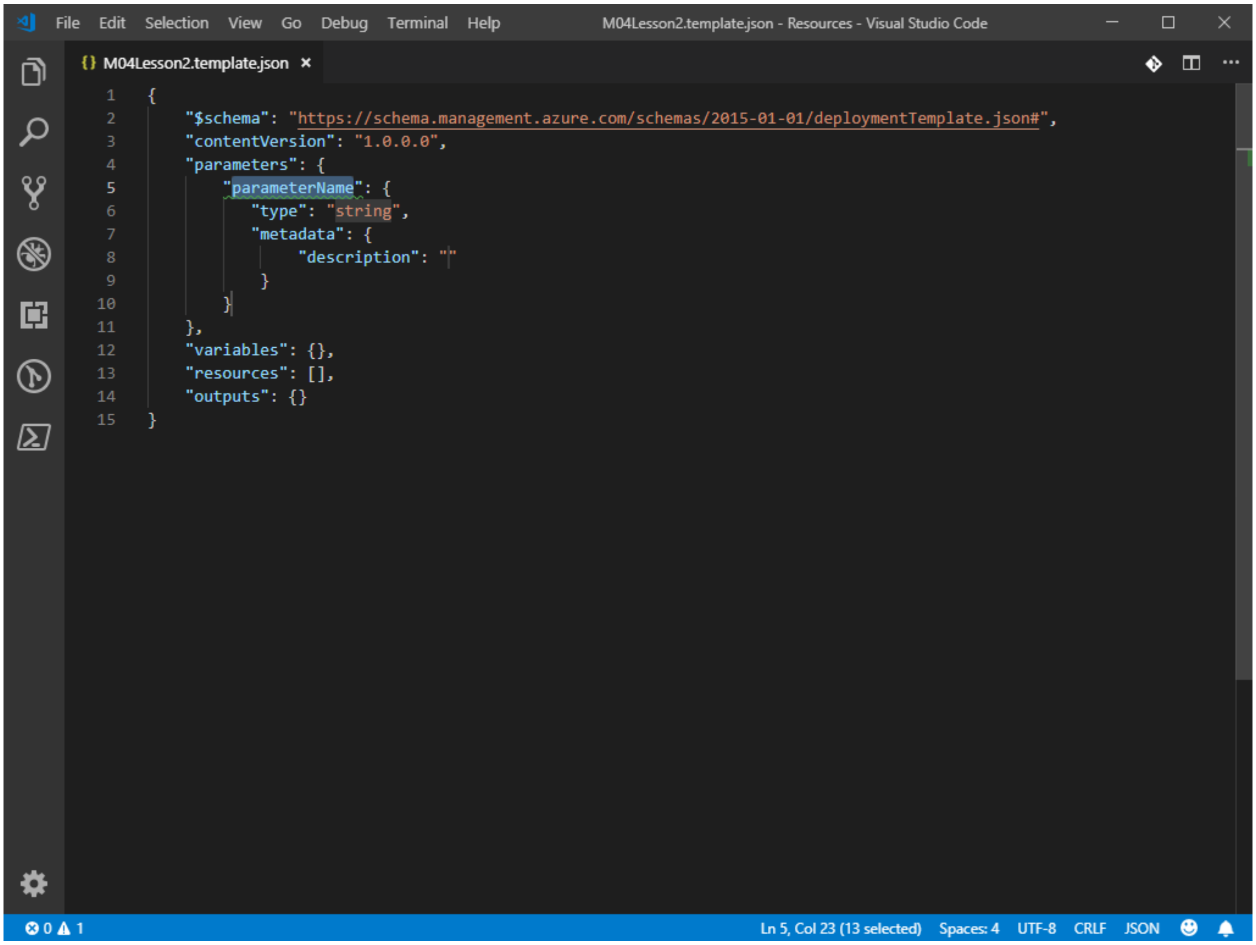


```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        arm-parameter
    },
    "variables": {},
    "resources": [],
    "outputs": {}
}
```

arm-parameter

ARM Parameter (Azure Resource Manager Snippets)

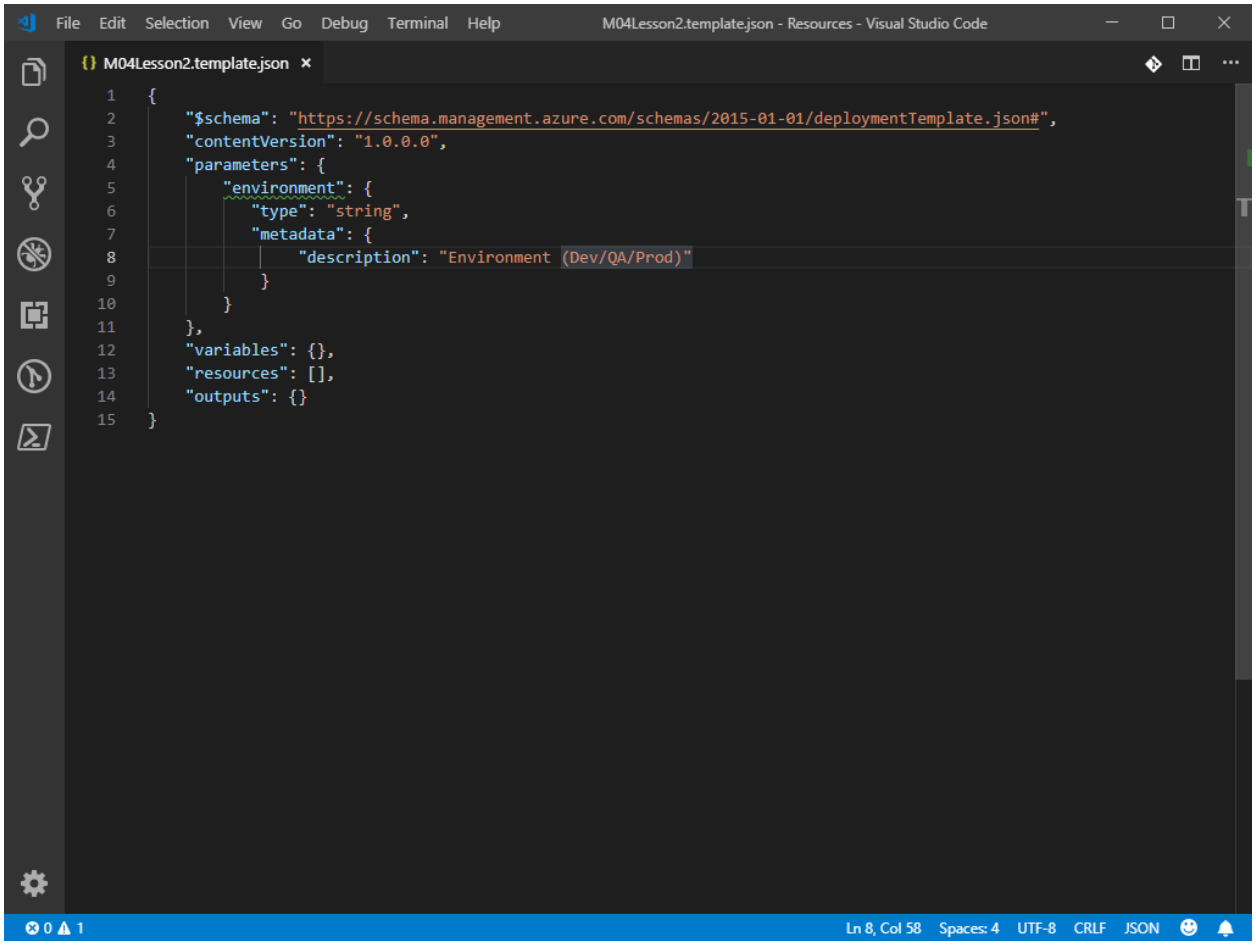
```
"parameterName": {
   "type": "string",
   "metadata": {
        "description": ""
    }
}
```



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

3. Change `"parameter1"` to `"environment"`

4. Set the `"description"` value to `"Environment (Dev/QA/Prod)"`



```json
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "environment": {
            "type": "string",
            "metadata": {
                "description": "Environment (Dev/QA/Prod)"
            }
        }
    },
    "variables": {},
    "resources": [],
    "outputs": {}
}
```

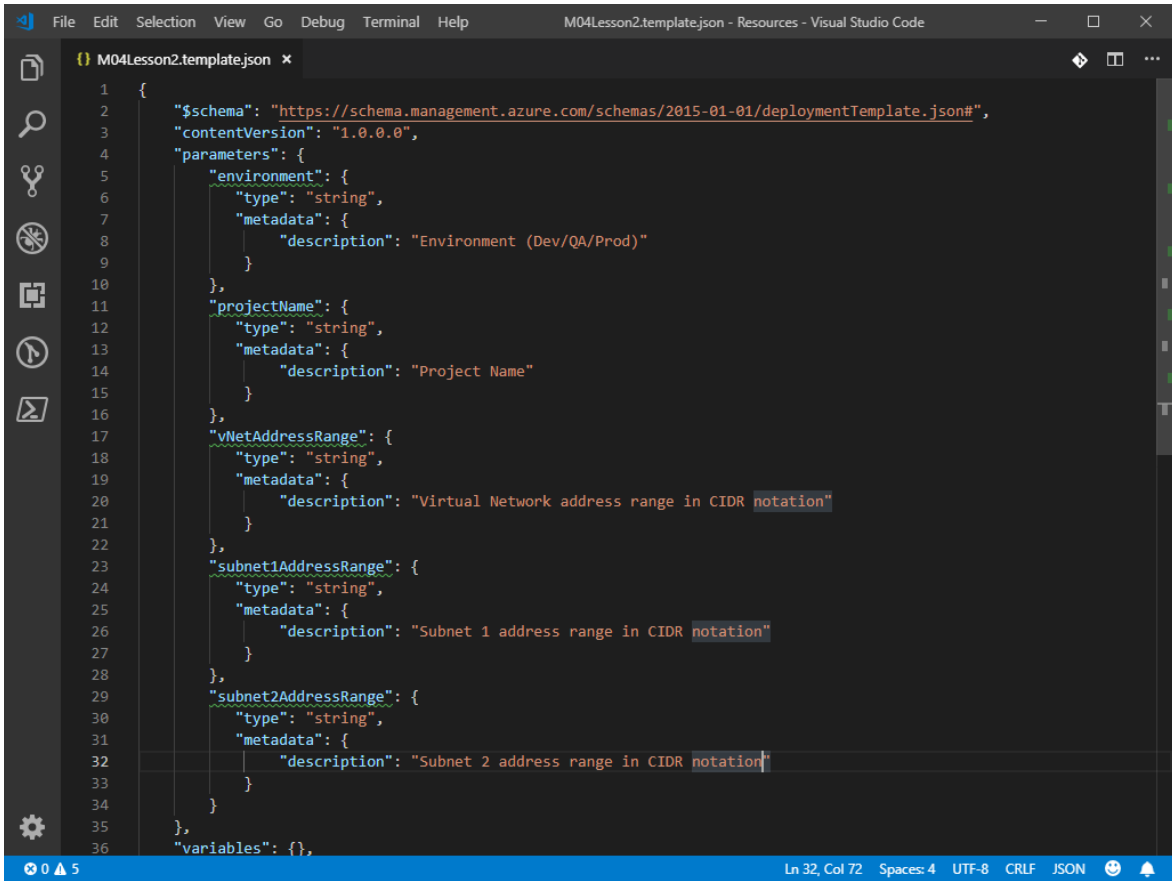
5. Add four additional parameters with a description of your choice. (**NOTE:** you will need to add a `,` after the closing `}` for each parameter block except the final parameter)

    a. `"projectName"`

    b. `"vNetAddressRange"`

    c. `"subnet1AddressRange"`

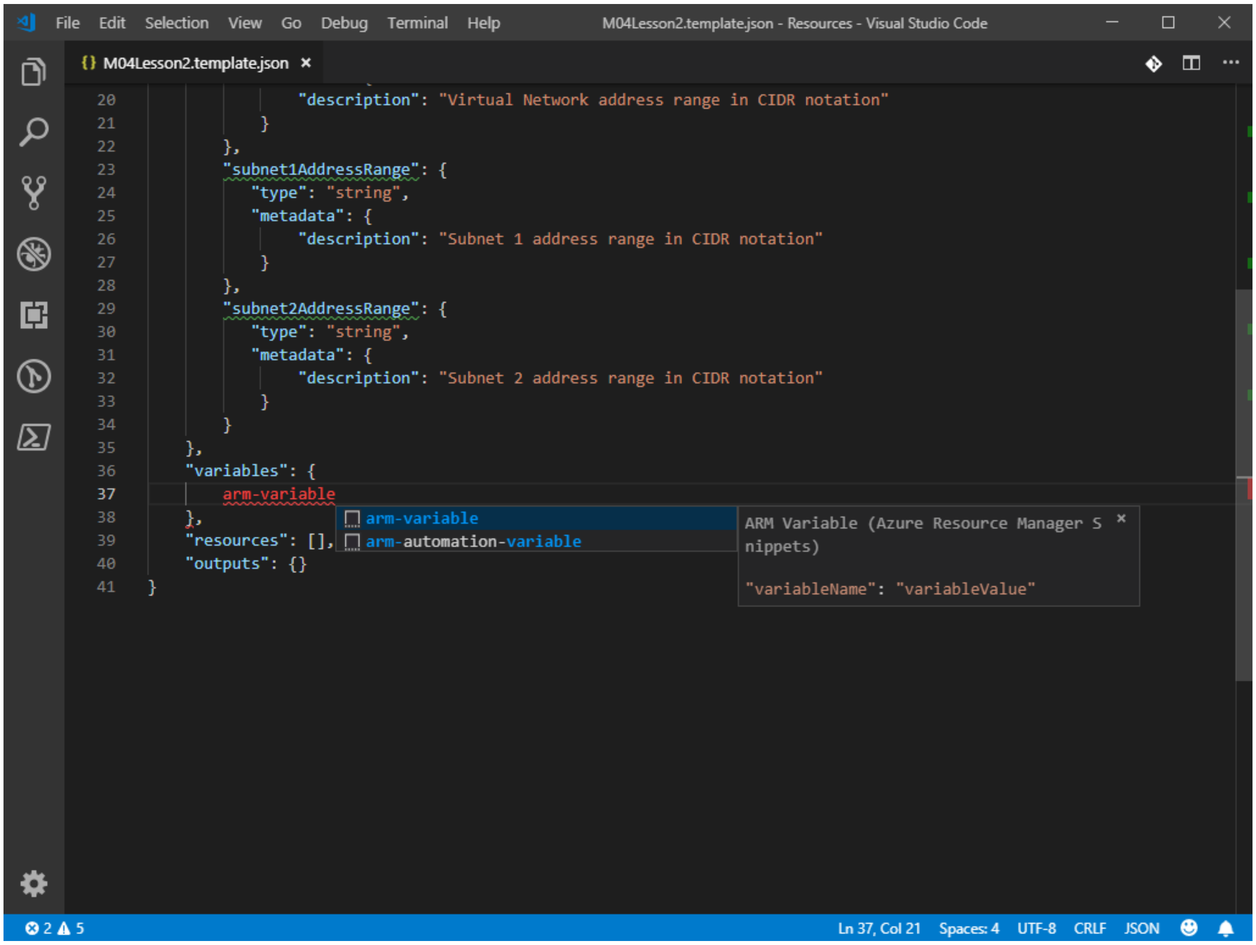
    d. `"subnet2AddressRange"`


File Edit Selection View Go Debug Terminal Help
M04Lesson2.template.json - Resources - Visual Studio Code
{} M04Lesson2.template.json ×
⊗ 0 ⚠ 5
Ln 32, Col 72 Spaces: 4 UTF-8 CRLF JSON

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "environment": {
            "type": "string",
            "metadata": {
                "description": "Environment (Dev/QA/Prod)"
            }
        },
        "projectName": {
            "type": "string",
            "metadata": {
                "description": "Project Name"
            }
        },
        "vNetAddressRange": {
            "type": "string",
            "metadata": {
                "description": "Virtual Network address range in CIDR notation"
            }
        },
        "subnet1AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 1 address range in CIDR notation"
            }
        },
        "subnet2AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 2 address range in CIDR notation"
            }
        }
    },
    "variables": {},
```

Add variables to the ARM template

1. Move your cursor in between the `{}` brackets on the line containing `"variables": {},` and press `Enter` to create a new line
2. Type `arm-variable` and press `Enter` to insert a new variable snippet



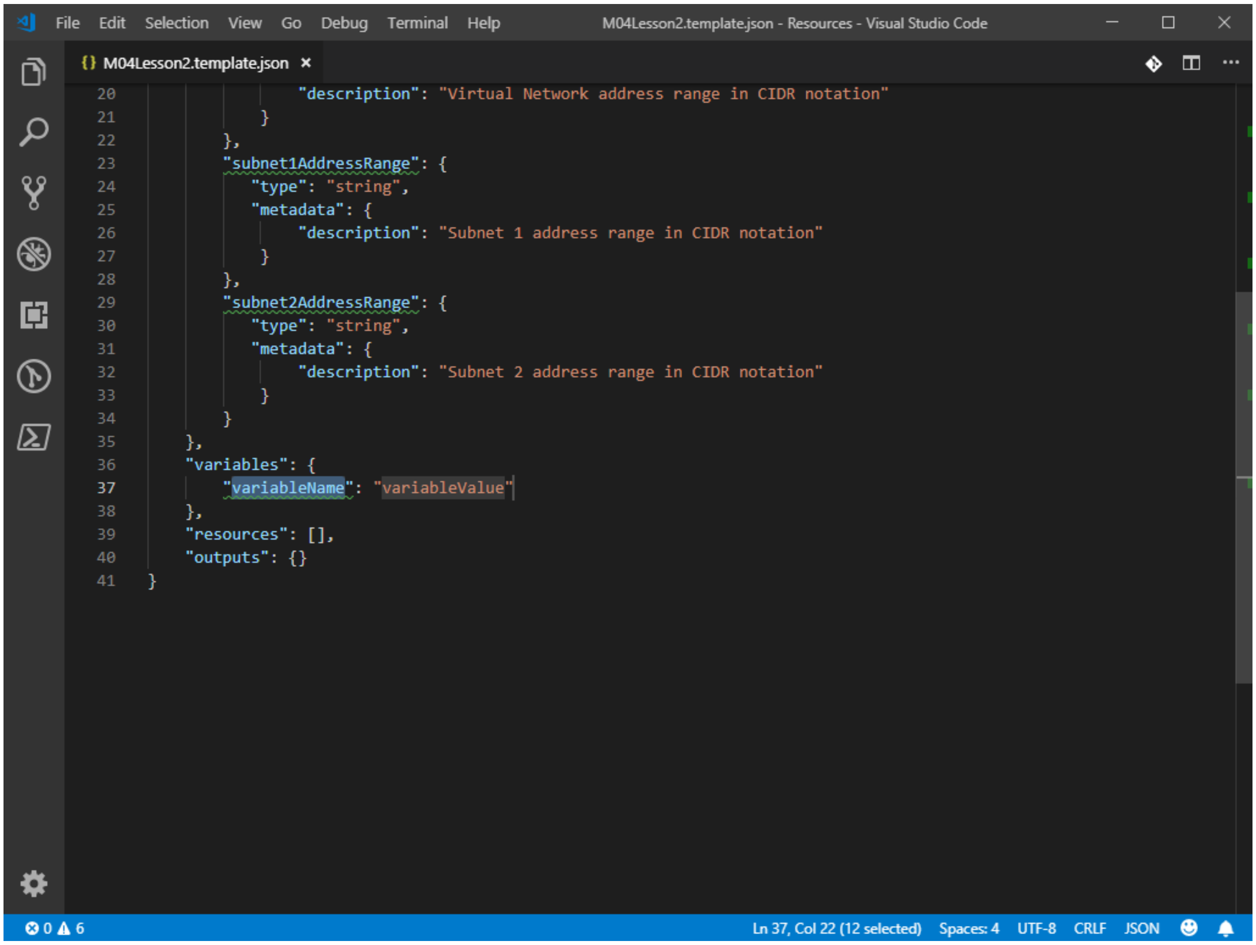
```
                "description": "Virtual Network address range in CIDR notation"
            }
        },
        "subnet1AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 1 address range in CIDR notation"
            }
        },
        "subnet2AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 2 address range in CIDR notation"
            }
        }
    },
    "variables": {
        arm-variable
    },
    "resources": [],
    "outputs": {}
}
```

□ arm-variable

□ arm-automation-variable

ARM Variable (Azure Resource Manager Snippets)

"variableName": "variableValue"


File Edit Selection View Go Debug Terminal Help
M04Lesson2.template.json - Resources - Visual Studio Code
{} M04Lesson2.template.json ×
⊗ 0 ⚠ 6
Ln 37, Col 22 (12 selected)
Spaces: 4
UTF-8
CRLF
JSON

```
                "description": "Virtual Network address range in CIDR notation"
            }
        },
        "subnet1AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 1 address range in CIDR notation"
            }
        },
        "subnet2AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 2 address range in CIDR notation"
            }
        }
    },
    "variables": {
        "variableName": "variableValue"
    },
    "resources": [],
    "outputs": {}
}
```

3. Change `"variable1"` to `"vNetName"`

4. Change `"value"` to `"[concat(parameters('projectName'), '-', parameters('environment'), '-VNet')]"`

    a. **NOTE:** Use IntelliSense to write this more efficiently

    b. Completely remove `"variableValue"` and start with a new `"`. Notice how Visual Studio Code automatically converts this to `""`

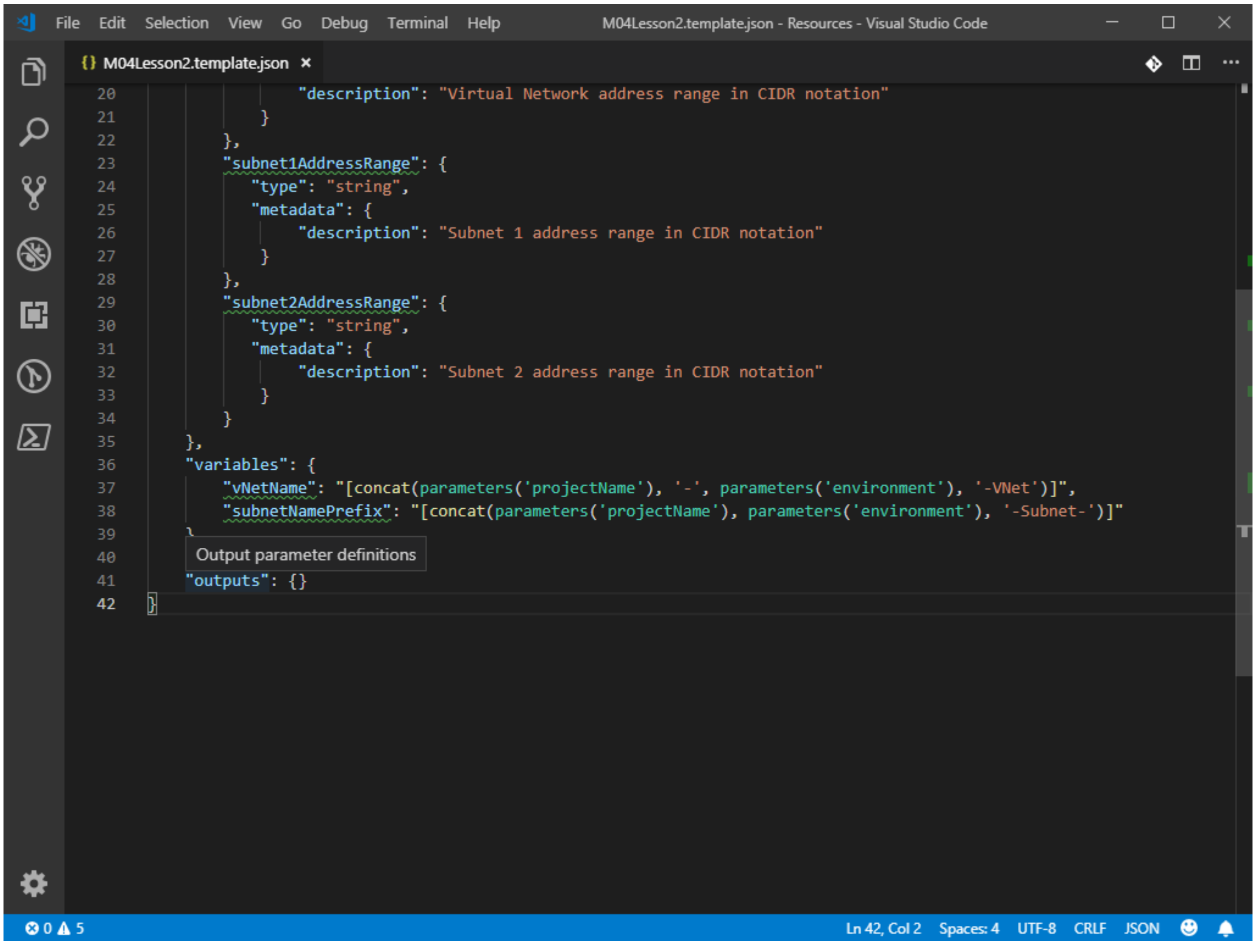
    c. Add a `[` and IntelliSense should present you with a list of functions. If it does not, you can trigger IntelliSense by pressing `Ctrl+Space` on Windows/Linux or `^+Space` on macOS

    d. Pick the `concat` function

    e. Now add the `parameters` function and notice how IntelliSense populates a list of available parameters

    f. Continue using IntelliSense to complete the variable value

5. Add another variable named `"subnetNamePrefix"` with a value of `"[concat(parameters('projectName'), parameters('environment'), '-Subnet-')]"`


File Edit Selection View Go Debug Terminal Help
M04Lesson2.template.json - Resources - Visual Studio Code
M04Lesson2.template.json
Output parameter definitions
0 ⚠ 5
Ln 42, Col 2 Spaces: 4 UTF-8 CRLF JSON

```
                "description": "Virtual Network address range in CIDR notation"
            }
        },
        "subnet1AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 1 address range in CIDR notation"
            }
        },
        "subnet2AddressRange": {
            "type": "string",
            "metadata": {
                "description": "Subnet 2 address range in CIDR notation"
            }
        }
    },
    "variables": {
        "vNetName": "[concat(parameters('projectName'), '-', parameters('environment'), '-VNet')]",
        "subnetNamePrefix": "[concat(parameters('projectName'), parameters('environment'), '-Subnet-')]"
    }
    "outputs": {}
}
```

Add a Virtual Network resource to the ARM template

1. Move your cursor in between the `[]` brackets on the line containing `"resources": [],` and press `Enter` to create a new line

2. Type `arm-vnet` and press `Enter` to insert a new variable snippet. (**NOTE:** Depending on the version of the snippet extension, the snippet may be referenced by another name such as `arm-vn`)

3. Change the values of `"name"` and `"displayName"` from `"VirtualNetwork1"` to `"[variables('vNetName')]"`

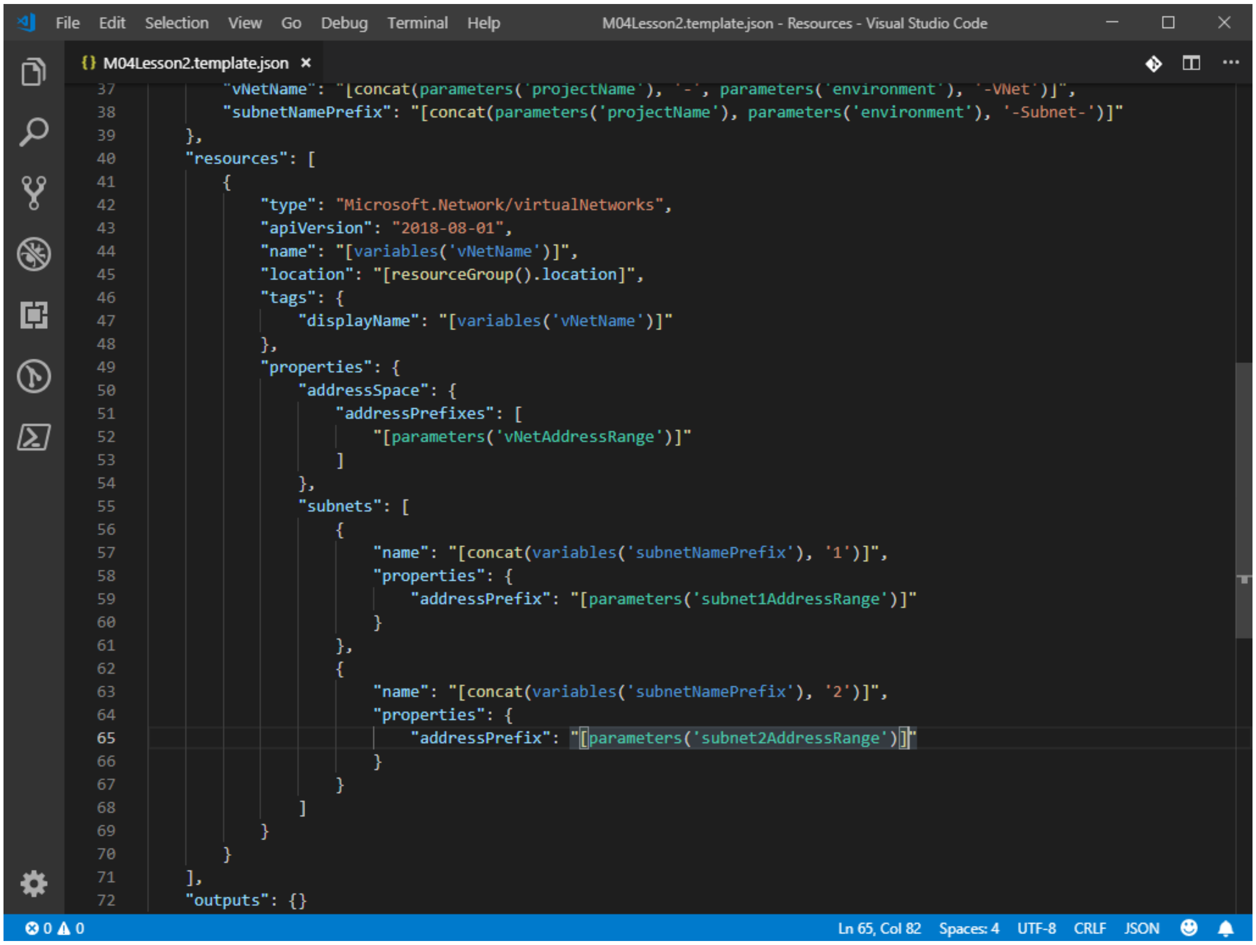
4. Change the value of `"addressPrefixes"` under the `"addressSpace"` property from `"10.0.0.0/16"` to `"[parameters('vNetAddressRange')]"`

5. Change the value of `"name"` for the first subnet object from `"Subnet-1"` to `"[concat(variables('subnetNamePrefix'), '1')]"`

6. Change the value of `"addressPrefix"` for the first subnet object from `"10.0.0.0/24"` to `"[parameters('subnet1AddressRange')]"`

7. Update the second subnet object accordingly

8. Once complete, save the file

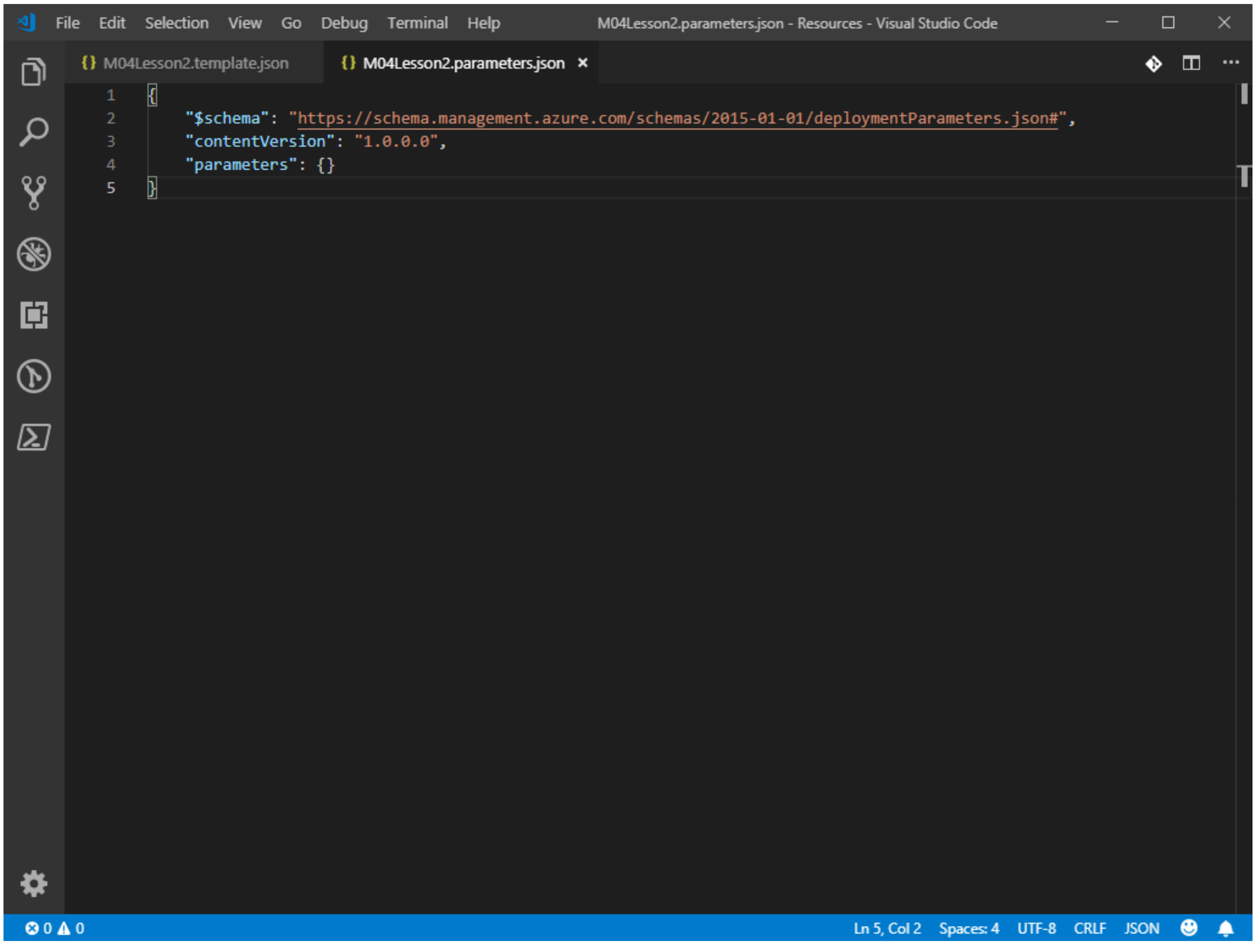


```
        "vNetName": "[concat(parameters('projectName'), '-', parameters('environment'), '-VNet')]",
        "subnetNamePrefix": "[concat(parameters('projectName'), parameters('environment'), '-Subnet-')]"
    },
    "resources": [
        {
            "type": "Microsoft.Network/virtualNetworks",
            "apiVersion": "2018-08-01",
            "name": "[variables('vNetName')]",
            "location": "[resourceGroup().location]",
            "tags": {
                "displayName": "[variables('vNetName')]"
            },
            "properties": {
                "addressSpace": {
                    "addressPrefixes": [
                        "[parameters('vNetAddressRange')]"
                    ]
                },
                "subnets": [
                    {
                        "name": "[concat(variables('subnetNamePrefix'), '1')]",
                        "properties": {
                            "addressPrefix": "[parameters('subnet1AddressRange')]"
                        }
                    },
                    {
                        "name": "[concat(variables('subnetNamePrefix'), '2')]",
                        "properties": {
                            "addressPrefix": "[parameters('subnet2AddressRange')]"
                        }
                    }
                ]
            }
        }
    ],
    "outputs": {}
```

Create a new ARM parameters file

1. Create a new file named `M04Lesson2.parameters.json` and open the file.

2. Type `armp!` and press `Enter` to insert the ARM parameters skeleton code snippet

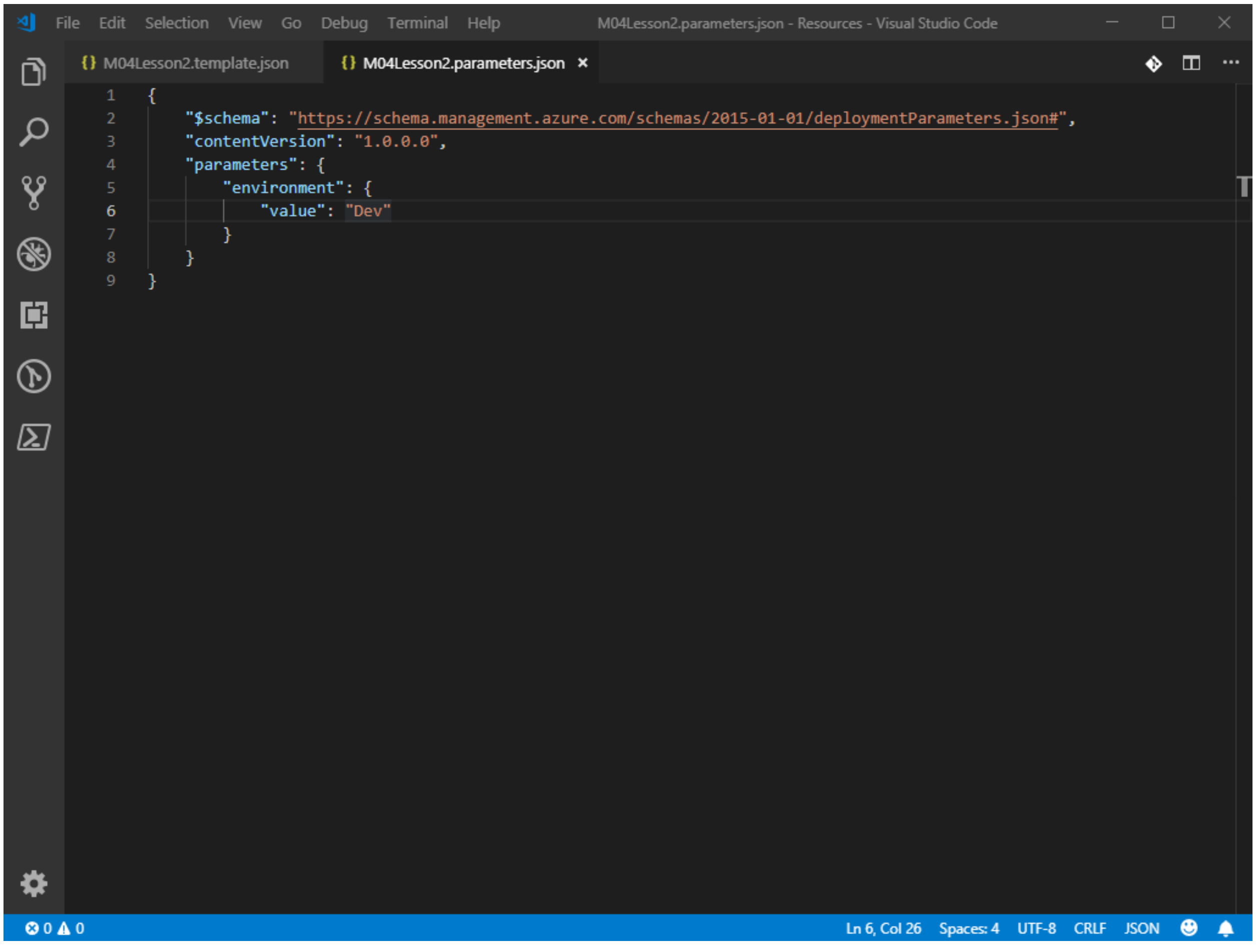


File Edit Selection View Go Debug Terminal Help
M04Lesson2.parameters.json - Resources - Visual Studio Code
M04Lesson2.template.json
M04Lesson2.parameters.json
Ln 5, Col 2 Spaces: 4 UTF-8 CRLF JSON

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

Add parameters to the ARM parameters file

1. Move your cursor in between the `{}` brackets on the line containing `"parameters": {},` and press `Enter` to create a new line
2. Type `new-parameter-value` and press `Enter` to insert a new parameter value snippet. (**NOTE:** Depending on the version of the snippet extension, the snippet may be referenced by another name such as `arm-paramvalue` or `arm-param-value`)
3. Change `"parameter1"` to `"environment"`
4. Set `"value"` to `"Dev"`

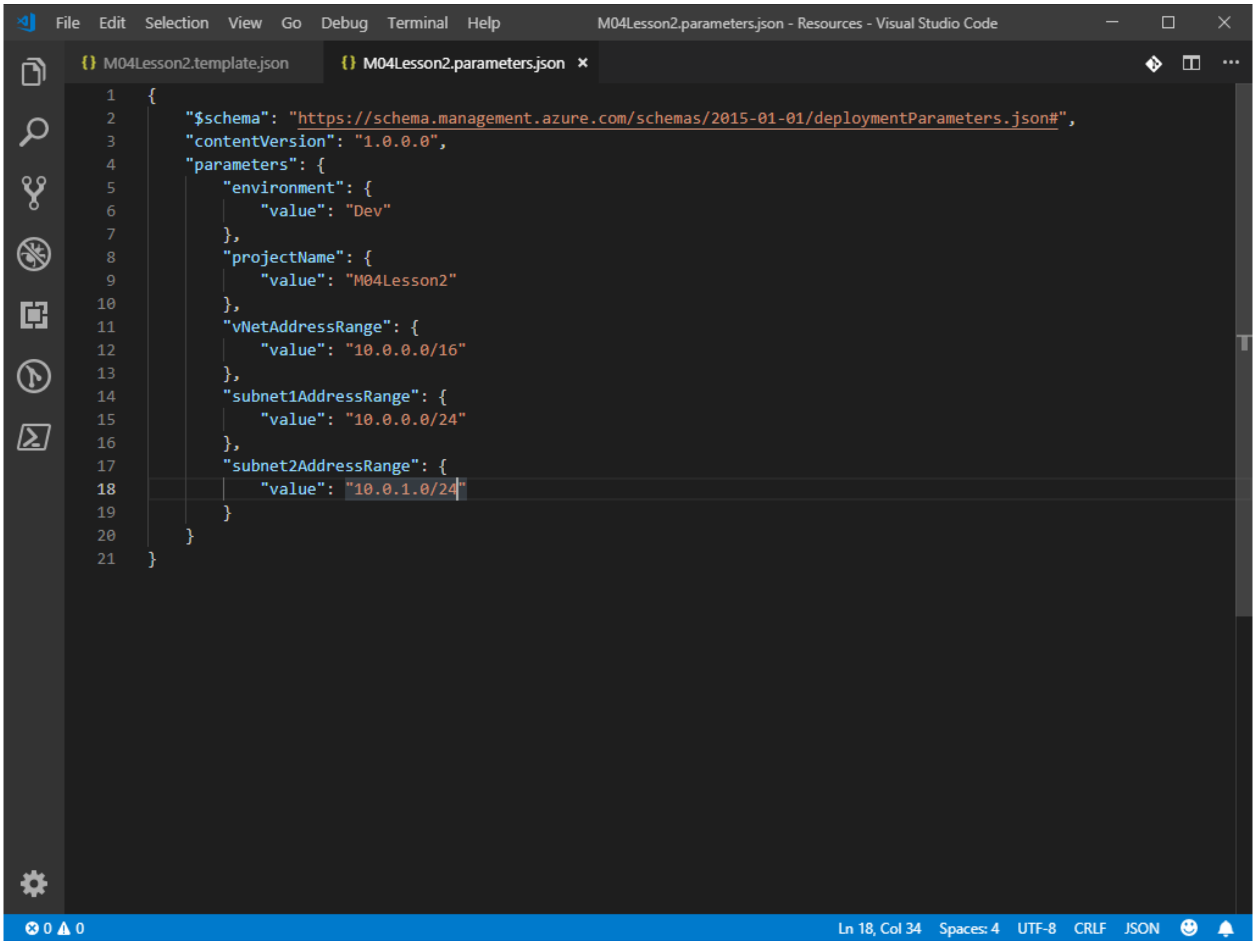

File Edit Selection View Go Debug Terminal Help
M04Lesson2.parameters.json - Resources - Visual Studio Code
M04Lesson2.template.json
M04Lesson2.parameters.json
0 0
Ln 6, Col 26 Spaces: 4 UTF-8 CRLF JSON

```json
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "environment": {
            "value": "Dev"
        }
    }
}
```

5. Add the remaining parameters with the following values

   a. `"projectName"` = `"M04Lesson2"`
   b. `"vNetAddressRange"` = `"10.0.0.0/16"`
   c. `"subnet1AddressRange"` = `"10.0.0.0/24"`
   d. `"subnet2AddressRange"` = `"10.0.1.0/24"`

6. Once complete, save the file


File Edit Selection View Go Debug Terminal Help
M04Lesson2.parameters.json - Resources - Visual Studio Code
{} M04Lesson2.template.json
{} M04Lesson2.parameters.json ×
⊗ 0 ⚠ 0
Ln 18, Col 34 Spaces: 4 UTF-8 CRLF JSON

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "environment": {
            "value": "Dev"
        },
        "projectName": {
            "value": "M04Lesson2"
        },
        "vNetAddressRange": {
            "value": "10.0.0.0/16"
        },
        "subnet1AddressRange": {
            "value": "10.0.0.0/24"
        },
        "subnet2AddressRange": {
            "value": "10.0.1.0/24"
        }
    }
}
```

Deploy the ARM template file with the parameters file

1. Open PowerShell in `C:\Lab_Files\M04\S02` (**NOTE:** Visual Studio Code has a built-in PowerShell terminal that can be used in place of a standalone terminal.)
2. Authenticate PowerShell to Azure by running `Connect-AzAccount` as T {USERNAME} using T {PASSWORD} as the password. (**NOTE:** If using Visual Studio Code's built-in terminal, the authentication window may be hidden behind the Visual Studio Code window.)
3. Run the following PowerShell commands to deploy the template

PowerShell

```powershell
Set-AzContext -Subscription '{SUBSCRIPTION_ID}'
New-AzResourceGroupDeployment -Name 'M04Lesson2' -ResourceGroupName '{RESOURCE_GRO
```

Inspect the deployed Virtual Network

1. Open the Azure Portal as T {USERNAME} using T {PASSWORD} as the password.
2. Navigate to the resource group `{RESOURCE_GROUP_NAME}`
3. Open the `M04Lesson2-Dev-VNet` virtual network

4. Notice the virtual network name and tag changed from `"[concat(parameters('projectName'), '-', parameters('environment'), '-VNet')]"` to `M04Lesson2-Dev-VNet`
5. Review the other virtual network properties and see how the template functions changed after deployment