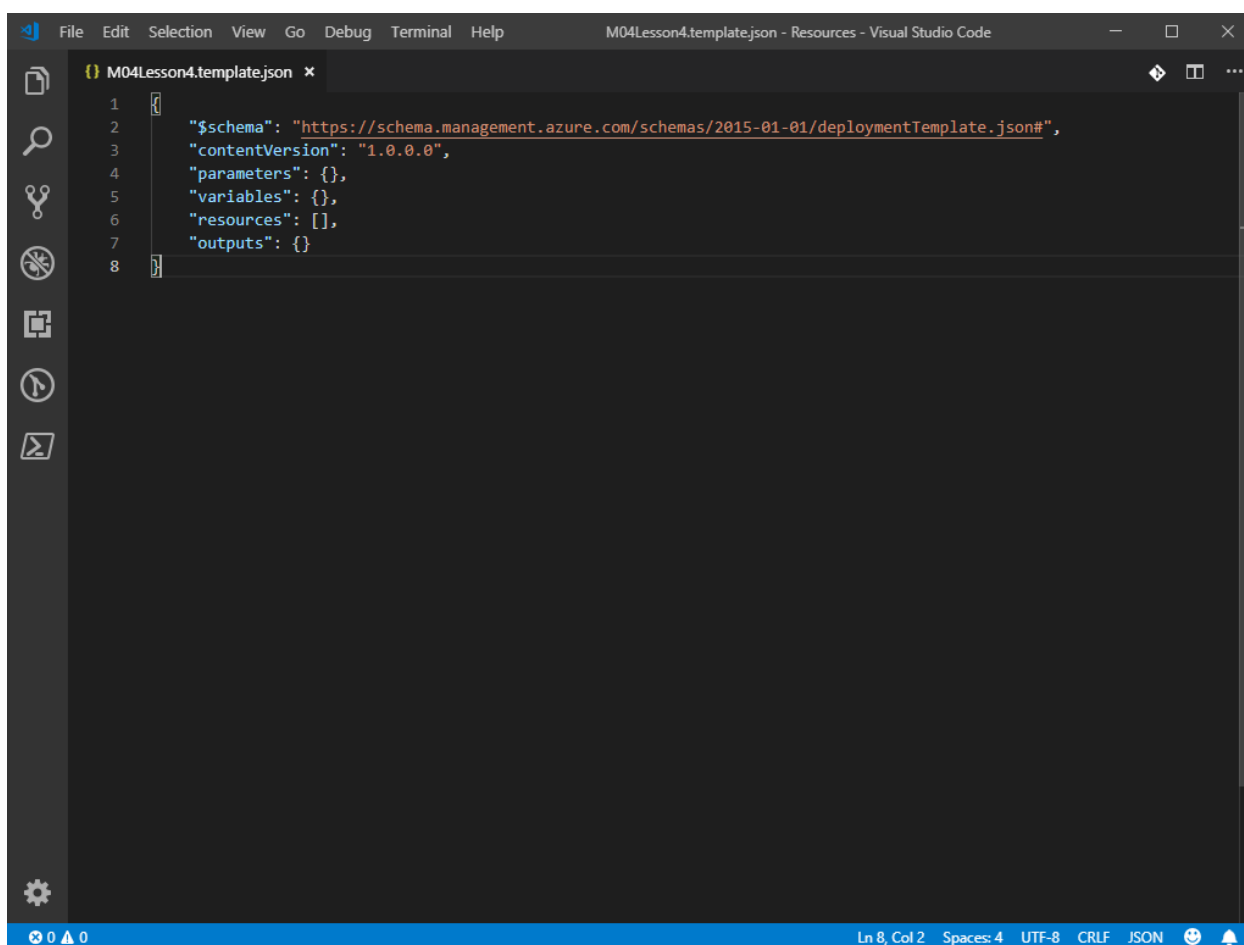


Module 4 - Lesson 4

⚠ Please note that the lab environment is restricted by policy. You will be able to create only the Azure resources required by the lab.

Create a new ARM template file

1. Open `C:\Lab_Files\M04` in Visual Studio Code and create a subfolder named `S04`
2. Create a new file in `C:\Lab_Files\M04\S04` named `M04Lesson4.template.json` and open the file.
3. Add the ARM template skeleton using the `arm!` code snippet



The screenshot shows the Visual Studio Code editor with the file `M04Lesson4.template.json` open. The editor displays the following JSON skeleton:

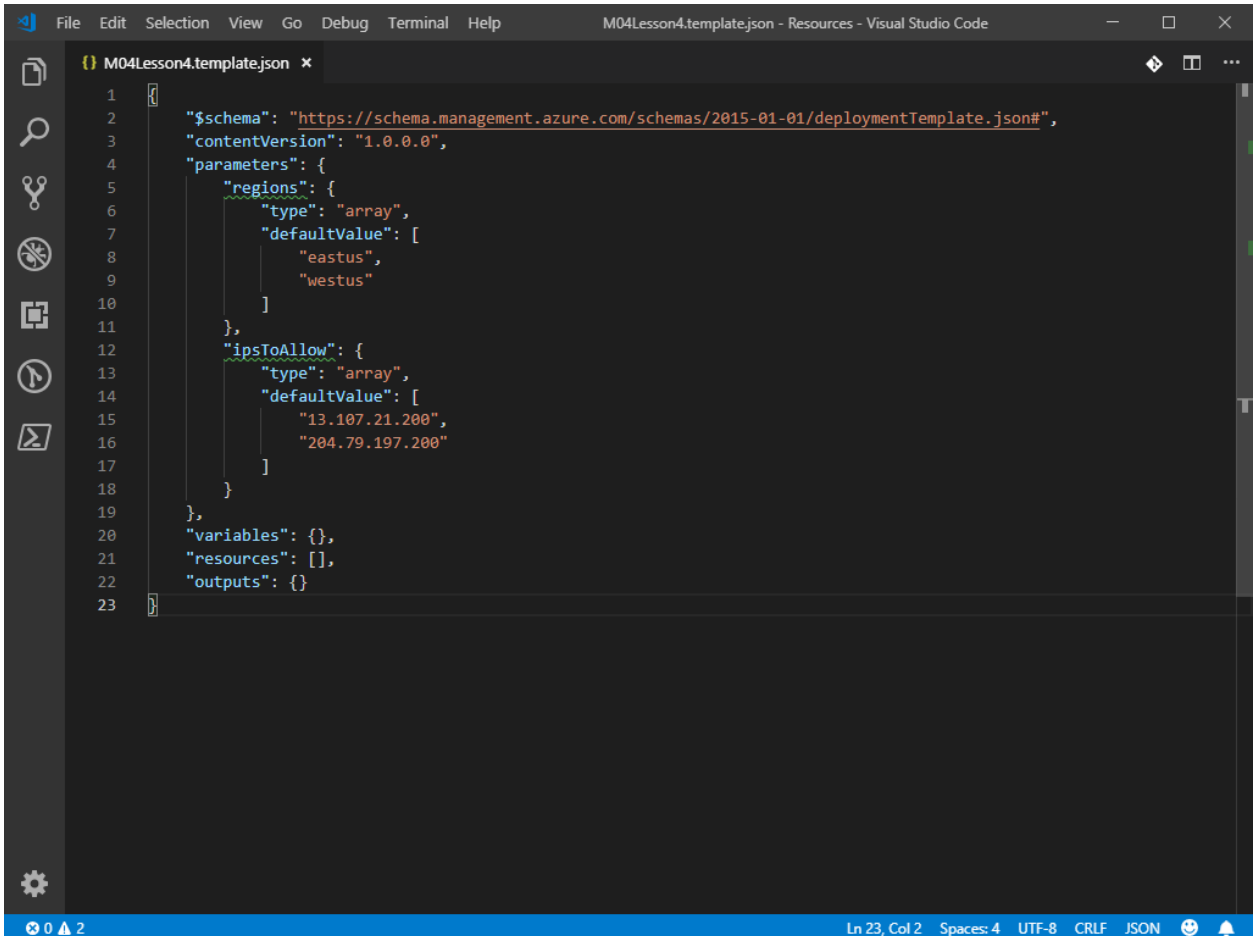
```
1 {  
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  
3   "contentVersion": "1.0.0.0",  
4   "parameters": {},  
5   "variables": {},  
6   "resources": [],  
7   "outputs": {}  
8 }
```

The status bar at the bottom indicates the file is at Line 8, Column 2, using UTF-8 encoding with CRLF line endings.

Add parameters to the ARM template

1. Add the following parameters to the `"parameters": {}` section
 - a. `regions`
 - i. `"type": "array"`

- ii. `"defaultValue": ["eastus", "westus"]`
- iii. **NOTE:** These must be valid Azure regions and can be found with `Get-AzLocation`
- b. `ipsToAllow`
 - i. `"type": "array"`
 - ii. `"defaultValue": ["13.107.21.200", "204.79.197.200"]`
 - iii. **NOTE:** These entries must be public IP's. The default values above are resolved from bing.com

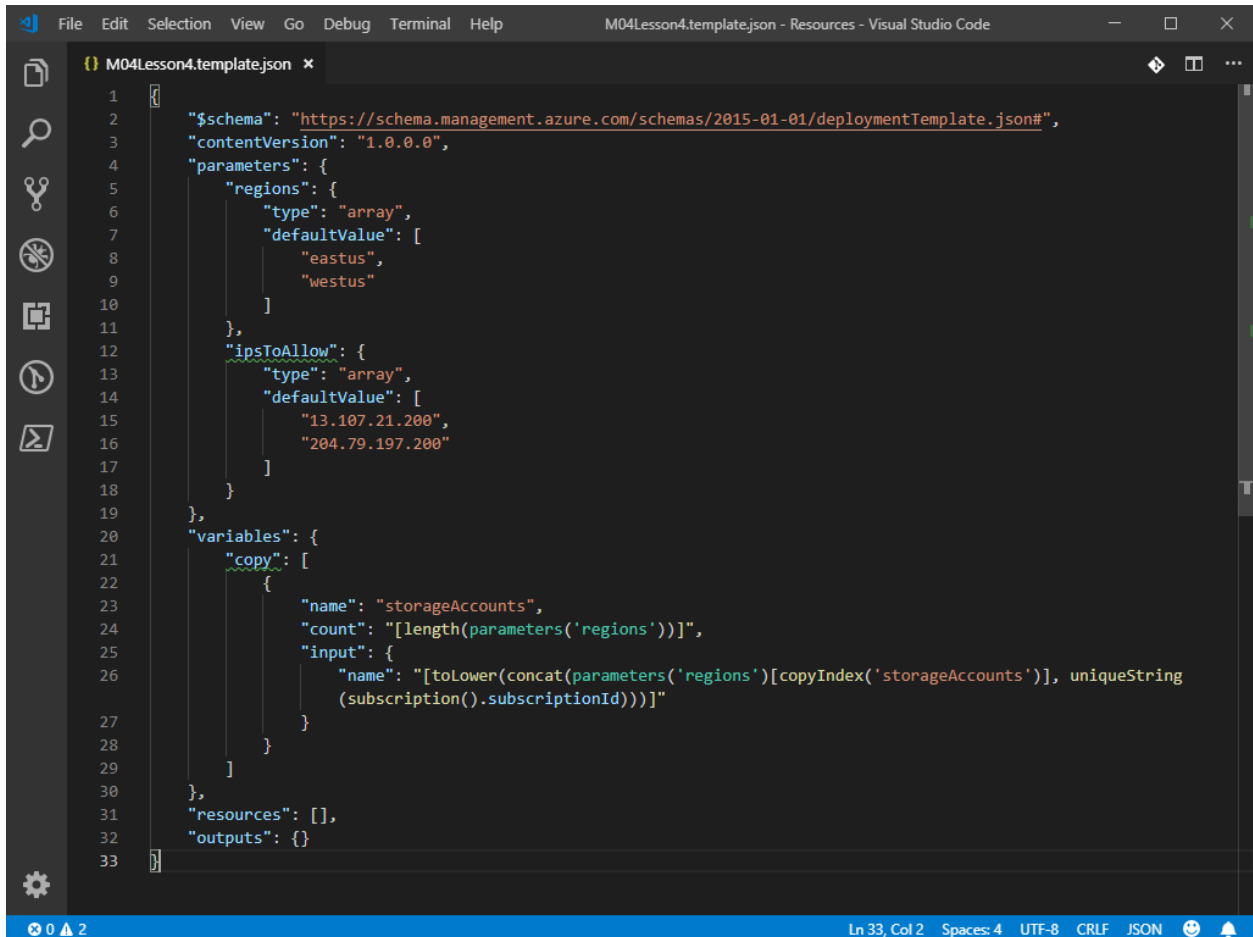


```
1 {
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "regions": {
6       "type": "array",
7       "defaultValue": [
8         "eastus",
9         "westus"
10      ]
11    },
12    "ipsToAllow": {
13      "type": "array",
14      "defaultValue": [
15        "13.107.21.200",
16        "204.79.197.200"
17      ]
18    }
19  },
20  "variables": {},
21  "resources": [],
22  "outputs": {}
23 }
```

Add a variable iteration copy loop to the ARM template

1. Add a variable iteration copy loop to the `"variables": {}` section
2. Move your cursor in between the `{}` brackets on the line containing `"variables": {}`, and press `Enter` to create a new line
3. Type `"copy": []`. With your cursor between the `[]`, press `Enter` to create a new line
4. Add a new object by typing `{}`. With your cursor between the `{}`, press `Enter` to create a new line
5. Inside the `{}` object, add the following

- a. `"name": "storageAccounts"`
 - i. This will be the name of the variable array created by the copy
 - b. `"count": "[length(parameters('regions'))]"`
6. Below the `"count"` property, add an `"input": {}` object
7. Inside the `"input": {}` object, add the following
- a. `"name": "[toLowerCase(concat(parameters('regions')[copyIndex('storageAccounts')], uniqueString(subscription().subscriptionId)))]"`



```

1  {
2    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3    "contentVersion": "1.0.0.0",
4    "parameters": {
5      "regions": {
6        "type": "array",
7        "defaultValue": [
8          "eastus",
9          "westus"
10       ]
11     },
12     "ipsToAllow": {
13       "type": "array",
14       "defaultValue": [
15         "13.107.21.200",
16         "204.79.197.200"
17       ]
18     }
19   },
20   "variables": {
21     "copy": [
22       {
23         "name": "storageAccounts",
24         "count": "[length(parameters('regions'))]",
25         "input": {
26           "name": "[toLowerCase(concat(parameters('regions')[copyIndex('storageAccounts')], uniqueString(subscription().subscriptionId)))]"
27         }
28       }
29     ]
30   },
31   "resources": [],
32   "outputs": {}
33 }

```

Add a storage resource to the ARM template

1. Add a storage account resource to the `"resources": []` section. This will be done manually, without using a snippet
2. Move your cursor in between the `[]` brackets on the line containing `"resources": []`, and press `Enter` to create a new line
3. Add a new object by typing `{}`. With your cursor between the `{}`, press `Enter` to create a new line
4. Inside the `{}` object, add the following

- a. `"type": "Microsoft.Storage/storageAccounts"`
- b. `"apiVersion": "2018-07-01"`
- c. `"name": "[variables('storageAccounts')[copyIndex()].name]"`
 - i. **NOTE:** VS Code IntelliSense will throw errors on this value because it cannot verify the variable name from a variable iteration copy loop.
- d. `"location": "[parameters('regions')[copyIndex()]]"`
- e. `"kind": "StorageV2"`

5. Notice how the `name` and `location` properties are pulling values from an array using an index

6. Below the `location` property, add a resource iteration copy loop by typing `"copy": {}`. With your cursor between the `{}`, press **Enter** to create a new line

7. Inside the `"copy": {}` object, add the following

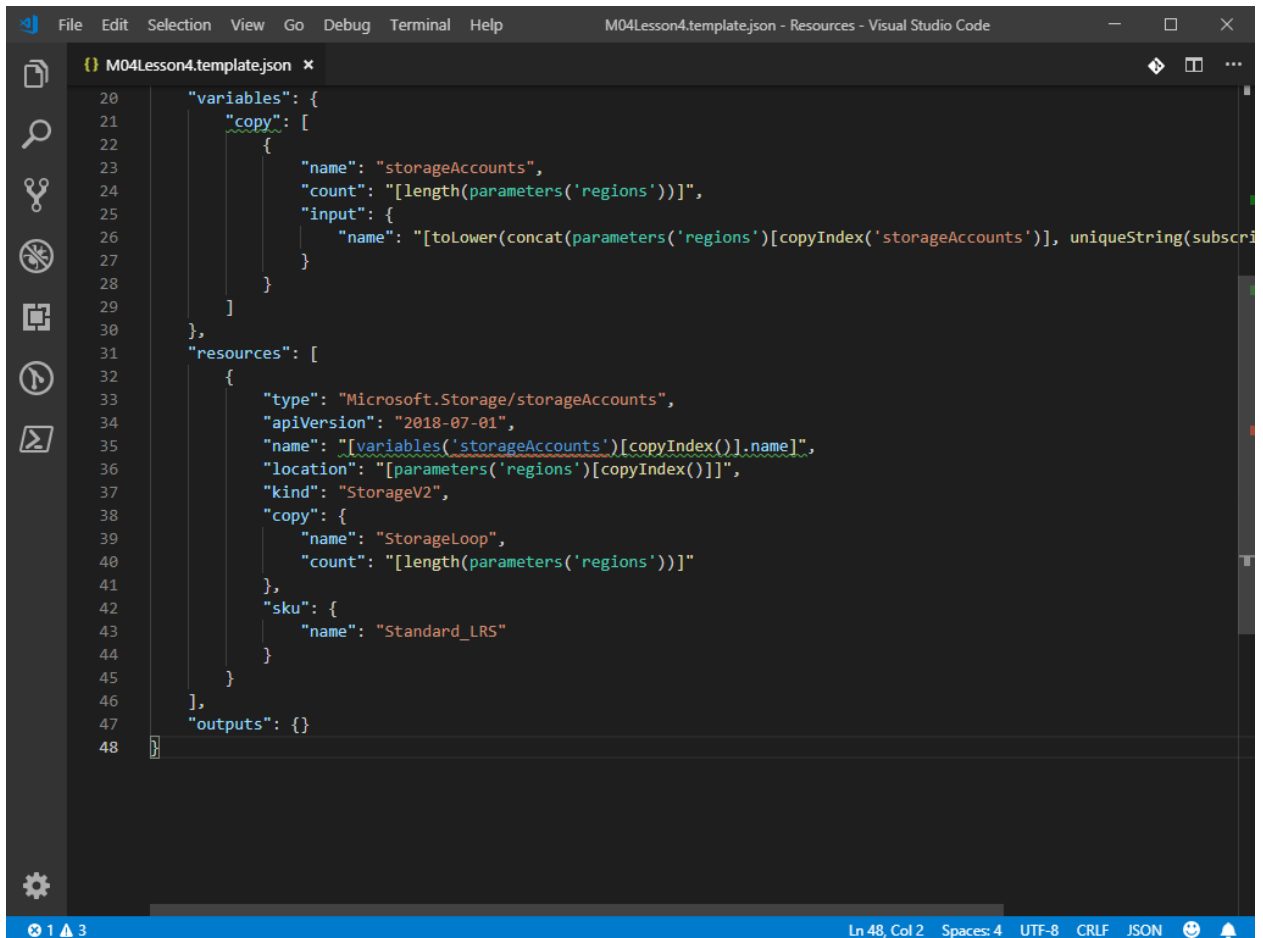
- a. `"name": "StorageLoop"`
 - i. **NOTE:** This name will be used for a dependency for another resource
- b. `"count": "[length(parameters('regions'))]"`

8. Below the `"copy": {}` object, add a `"sku": {}` object. With your cursor between the `{}`, press **Enter** to create a new line

9. Inside the `"sku": {}` object, add the following

- a. `"name": "Standard_LRS"`

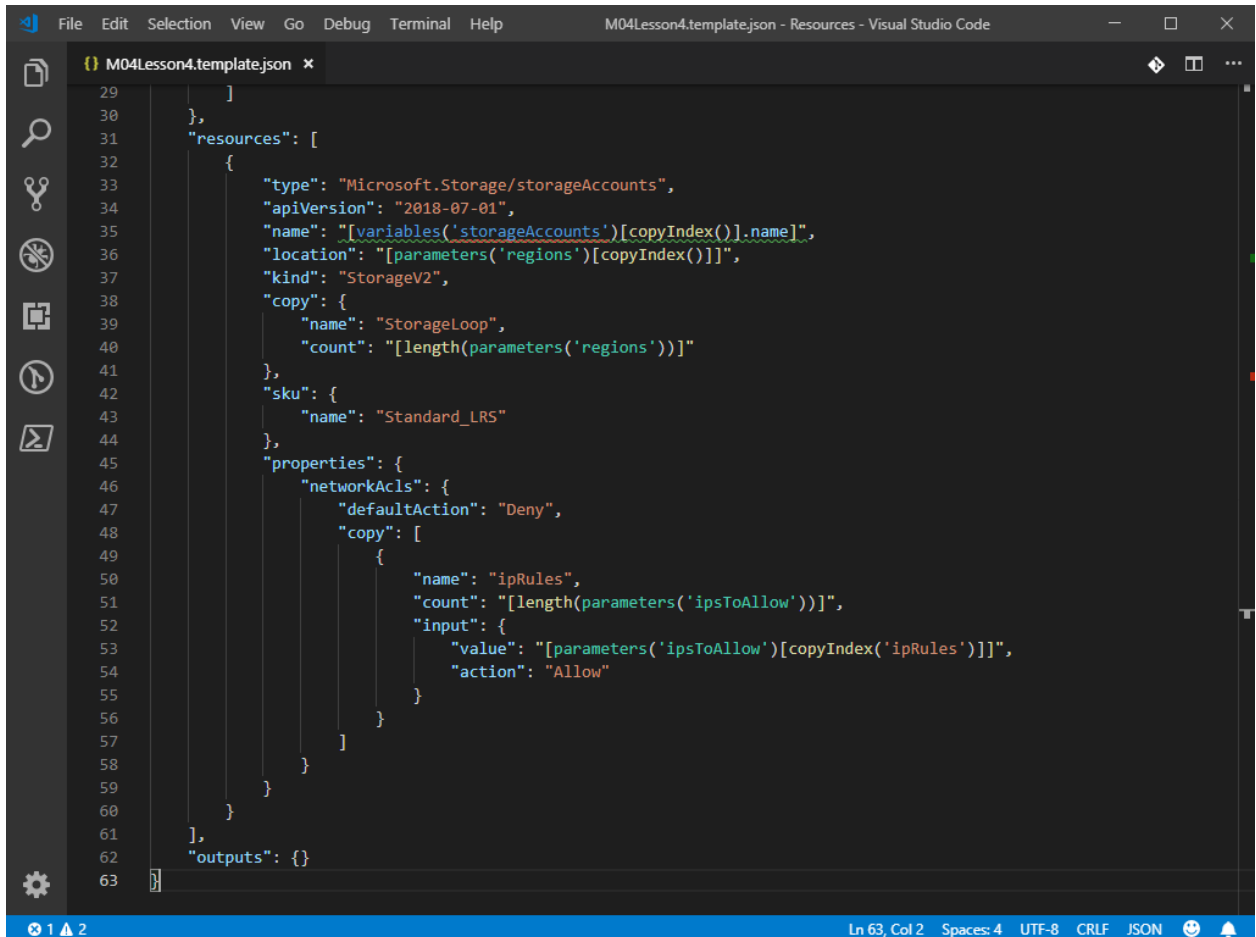
10. At this point, your resource should look as follows



```
20  "variables": {
21    "copy": [
22      {
23        "name": "storageAccounts",
24        "count": "[length(parameters('regions'))]",
25        "input": {
26          "name": "[toLower(concat(parameters('regions')[copyIndex('storageAccounts')], uniqueString(subscri
27        }
28      ]
29    },
30    "resources": [
31      {
32        "type": "Microsoft.Storage/storageAccounts",
33        "apiVersion": "2018-07-01",
34        "name": "[variables('storageAccounts')[copyIndex()].name]",
35        "location": "[parameters('regions')[copyIndex()]]",
36        "kind": "StorageV2",
37        "copy": {
38          "name": "StorageLoop",
39          "count": "[length(parameters('regions'))]"
40        },
41        "sku": {
42          "name": "Standard_LRS"
43        }
44      }
45    ],
46    "outputs": {}
47  }
48 }
```

11. Below the `"sku": {}` object, add a `"properties": {}` object. With your cursor between the `{}`, press **Enter** to create a new line
12. Inside the `"properties": {}` object, add a `"networkAcls": {}` object. With your cursor between the `{}`, press **Enter** to create a new line
13. Inside the `"networkAcls": {}` object, add the following
 - a. `"defaultAction": "Deny"`
14. Below the `defaultAction` property, add a property iteration copy loop by typing `"copy": []`. With your cursor between the `[]`, press **Enter** to create a new line
15. Add a new object by typing `{}`. With your cursor between the `{}`, press **Enter** to create a new line
16. Inside the `{}` object, add the following
 - a. `"name": "ipRules"`
 - b. `"count": "[length(parameters('ipsToAllow'))]"`
17. Below the `"count"` property, add an `"input": {}` object
18. Inside the `"input": {}` object, add the following
 - a. `"value": "[parameters('ipsToAllow')[copyIndex('ipRules')]]"`

b. `"action": "Allow"`



```
29   ],
30   },
31   "resources": [
32     {
33       "type": "Microsoft.Storage/storageAccounts",
34       "apiVersion": "2018-07-01",
35       "name": "[variables('storageAccounts')[copyIndex()].name]",
36       "location": "[parameters('regions')[copyIndex()]]",
37       "kind": "StorageV2",
38       "copy": {
39         "name": "StorageLoop",
40         "count": "[length(parameters('regions'))]"
41       },
42       "sku": {
43         "name": "Standard_LRS"
44       },
45       "properties": {
46         "networkAcls": {
47           "defaultAction": "Deny",
48           "copy": [
49             {
50               "name": "ipRules",
51               "count": "[length(parameters('ipsToAllow'))]",
52               "input": {
53                 "value": "[parameters('ipsToAllow')[copyIndex('ipRules')]]",
54                 "action": "Allow"
55               }
56             }
57           ]
58         }
59       }
60     },
61   ],
62   "outputs": {}
63 }
```

Add a blob container child resource to the ARM template

1. You will now add a blob container resource to the template. This is called a child resource, with the parent resource being the storage account.
2. Below the `Microsoft.Storage/storageAccounts` object, add a new object by typing `{ }`. With your cursor between the `{ }`, press `Enter` to create a new line
3. Inside the `{ }` object, add the following
 - a. `"type": "Microsoft.Storage/storageAccounts/blobServices/containers"`
 - b. `"apiVersion": "2018-07-01"`
 - c. `"name": "[concat(variables('storageAccounts')[copyIndex()].name, '/default/container1')]"`
 - i. Notice how the blob container name must reference the parent storage account resource
4. Below the `name` property, add a resource iteration copy loop by typing `"copy": { }`. With your cursor between the `{ }`, press `Enter` to create a new line
5. Inside the `"copy": { }` object, add the following

- a. `"name": "ContainerLoop"`
 - b. `"count": "[length(parameters('regions'))]"`
6. Below the `"copy": {}` object, add a `"dependsOn": []` array. With your cursor between the `[]`, press **Enter** to create a new line
7. Inside the `"dependsOn": []` array, add the following
- a. `"StorageLoop"`
 - i. Notice how this reference the copy loop name from the storage account resource iteration instead of a named resource

```

41  },
42  "sku": {
43    "name": "Standard_LRS"
44  },
45  "properties": {
46    "networkAcls": {
47      "defaultAction": "Deny",
48      "copy": [
49        {
50          "name": "ipRules",
51          "count": "[length(parameters('ipsToAllow'))]",
52          "input": {
53            "value": "[parameters('ipsToAllow')[copyIndex('ipRules')]]",
54            "action": "Allow"
55          }
56        }
57      ]
58    }
59  },
60  },
61  {
62    "type": "Microsoft.Storage/storageAccounts/blobServices/containers",
63    "apiVersion": "2018-07-01",
64    "name": "[concat(variables('storageAccounts')[copyIndex()].name, '/default/container1')]",
65    "copy": {
66      "name": "ContainerLoop",
67      "count": "[length(parameters('regions'))]"
68    },
69    "dependsOn": [
70      "StorageLoop"
71    ]
72  }
73 ],
74 "outputs": {}
75 ]

```

Deploy the ARM template file

1. Open PowerShell in `C:\Lab_Files\M04\S04`
2. Authenticate PowerShell to Azure by running `Connect-AzAccount` as `{USERNAME}` using `{PASSWORD}` as the password.
3. Run the following PowerShell commands to deploy the template

PowerShell

```
Set-AzContext -Subscription '{SUBSCRIPTION_ID}'
```

```
New-AzResourceGroupDeployment -Name 'M04Lesson4' -ResourceGroupName '{RESOURCE_G
```



Because both parameters have default values, you do not need to defined a parameter file. If you would like to override these values, please create a coordinating ARM parameter file. See the lab for Module 4 Lesson 2 for details on creating this file. If using a parameters file, use the command below

PowerShell



```
New-AzResourceGroupDeployment -Name 'M04Lesson4' -ResourceGroupName '{RESOURCE_GROU
```

Inspect the deployed Storage Account

1. Open the Azure Portal as `{USERNAME}` using `{PASSWORD}` as the password.
2. Navigate to the resource group `{RESOURCE_GROUP_NAME}`
3. There should be one storage account per region defined in the `regions` parameter array
4. Open one of the storage accounts
5. Inside the **Firewalls and virtual networks** blade, there should be an entry for each public IP defined in the `ipsToAllow` parameter array
6. Inside the **Blobs** blade, there should be a container named container1
7. Review the other storage accounts deployed. Their properties should match with the exception of being deployed to a different region