

Update a resource in an Azure Resource Manager template

10/31/2018 • 3 minutes to read • Contributors  all

In this article

[Example template](#)

[Try the template](#)

[Next steps](#)

There are some scenarios in which you need to update a resource during a deployment. You might encounter this scenario when you cannot specify all the properties for a resource until other, dependent resources are created. For example, if you create a backend pool for a load balancer, you might update the network interfaces (NICs) on your virtual machines (VMs) to include them in the backend pool. And while Resource Manager supports updating resources during deployment, you must design your template correctly to avoid errors and to ensure the deployment is handled as an update.

First, you must reference the resource once in the template to create it and then reference the resource by the same name to update it later. However, if two resources have the same name in a template, Resource Manager throws an exception. To avoid this error, specify the updated resource in a second template that's either linked or included as a subtemplate using the `Microsoft.Resources/deployments` resource type.

Second, you must either specify the name of the existing property to change or a new name for a property to add in the nested template. You must also specify the original properties and their original values. If you fail to provide the original properties and values, Resource Manager assumes you want to create a new resource and deletes the original resource.

Example template

Let's look at an example template that demonstrates this. Our template deploys a virtual network named `firstVNet` that has one subnet named `firstSubnet`. It then deploys a virtual network interface (NIC) named `nic1` and associates it with our subnet. Then, a deployment resource named `updateVNet` includes a nested template that updates our `firstVNet` resource by adding a second subnet named `secondSubnet`.

JSON

 Copy

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {},
  "resources": [
    {
      "apiVersion": "2016-03-30",
      "name": "firstVNet",
      "location": "[resourceGroup().location]",
      "type": "Microsoft.Network/virtualNetworks",
      "properties": {
        "addressSpace": { "addressPrefixes": [
          "10.0.0.0/22"
        ] },
        "subnets": [
          {
            "name": "firstSubnet",
            "properties": {
```

```

        "addressPrefix": "10.0.0.0/24"
    }
}
    ]
}
},
{
    "apiVersion": "2015-06-15",
    "type": "Microsoft.Network/networkInterfaces",
    "name": "nic1",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "firstVNet"
    ],
    "properties": {
        "ipConfigurations": [
            {
                "name": "ipconfig1",
                "properties": {
                    "privateIPAllocationMethod": "Dynamic",
                    "subnet": {
                        "id": "[concat(resourceId('Microsoft.Network/virtualNetworks', 'firstVNet'), '/subnets/firstSubnet')]"
                    }
                }
            }
        ]
    }
},
{
    "apiVersion": "2015-01-01",
    "type": "Microsoft.Resources/deployments",
    "name": "updateVNet",
    "dependsOn": [
        "nic1"
    ],
    "properties": {
        "mode": "Incremental",
        "parameters": {},
        "template": {
            "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
            "contentVersion": "1.0.0.0",
            "parameters": {},
            "variables": {},
            "resources": [
                {
                    "apiVersion": "2016-03-30",
                    "name": "firstVNet",
                    "location": "[resourceGroup().location]",
                    "type": "Microsoft.Network/virtualNetworks",
                    "properties": {
                        "addressSpace": "[reference('firstVNet').addressSpace]",
                        "subnets": [
                            {
                                "name": "[reference('firstVNet').subnets[0].name]",
                                "properties": {
                                    "addressPrefix": "[reference('firstVNet').subnets[0].properties.addressPrefix]"
                                }
                            },
                            {
                                "name": "secondSubnet",
                                "properties": {
                                    "addressPrefix": "10.0.1.0/24"
                                }
                            }
                        ]
                    }
                }
            ]
        }
    }
}
}

```

```

    },
    "outputs": {}
  }
},
"outputs": {}
}

```

Let's take a look at the resource object for our `firstVNet` resource first. Notice that we respecify the settings for our `firstVNet` in a nested template—this is because Resource Manager doesn't allow the same deployment name within the same template and nested templates are considered to be a different template. By respecifying our values for our `firstSubnet` resource, we are telling Resource Manager to update the existing resource instead of deleting it and redeploying it. Finally, our new settings for `secondSubnet` are picked up during this update.

Try the template

An example template is available on [GitHub](#). To deploy the template, run the following [Azure CLI](#) commands:

```

bash

az group create --location <location> --name <resource-group-name>
az group deployment create -g <resource-group-name> \
  --template-uri https://raw.githubusercontent.com/mspnp/template-examples/master/example1-
  update/deploy.json

```

Once deployment has finished, open the resource group you specified in the portal. You see a virtual network named `firstVNet` and a NIC named `nic1`. Click `firstVNet`, then click `subnets`. You see the `firstSubnet` that was originally created, and you see the `secondSubnet` that was added in the `updateVNet` resource.

NAME	ADDRESS RANGE	AVAILABLE ADDRESSES	SECURITY GROUP
<code>firstSubnet</code>	10.0.0.0/24	250	-
<code>secondSubnet</code>	10.0.1.0/24	251	-

Then, go back to the resource group and click `nic1` then click `IP configurations`. In the `IP configurations` section, the subnet is set to `firstSubnet (10.0.0.0/24)`.

IP configurations

* Subnet

firstSubnet (10.0.0.0/24)

Search IP configurations

NAME	IP VERSION	TYPE	PRIVATE IP ADDRESS	PUBLIC IP ADDRESS
<code>ipconfig1</code>	IPv4	Primary	10.0.0.4 (Dynamic)	-

The original `firstVNet` has been updated instead of re-created. If `firstVNet` had been re-created, `nic1` would not be associated with `firstVNet`.

Next steps

- Learn how deploy a resource based on a condition, such as whether a parameter value is present. See [Conditionally deploy a resource in an Azure Resource Manager template](#).

