😊 Creating a Network Interface

In this lab exercise, the goal is to create a network interface in Azure using an ARM (Azure Resource Manager) template. The steps involve:

- **Creating a new ARM template in Visual Studio Code.**
- **Writing code to define the properties of the network interface, including referencing the existing virtual network and public IP address.**
- **Deploying the template through the Azure Portal's template deployment feature.**
- **Verifying that the deployment is successful and confirming that the network interface is correctly attached to the specified resources.**

The end goal is to provide hands-on experience with Azure resource deployment using ARM templates, understanding how to define and deploy infrastructure as code, and ensuring proper resource association within an Azure environment. This exercise helps users become familiar with key concepts of Azure resource management and deployment workflows.

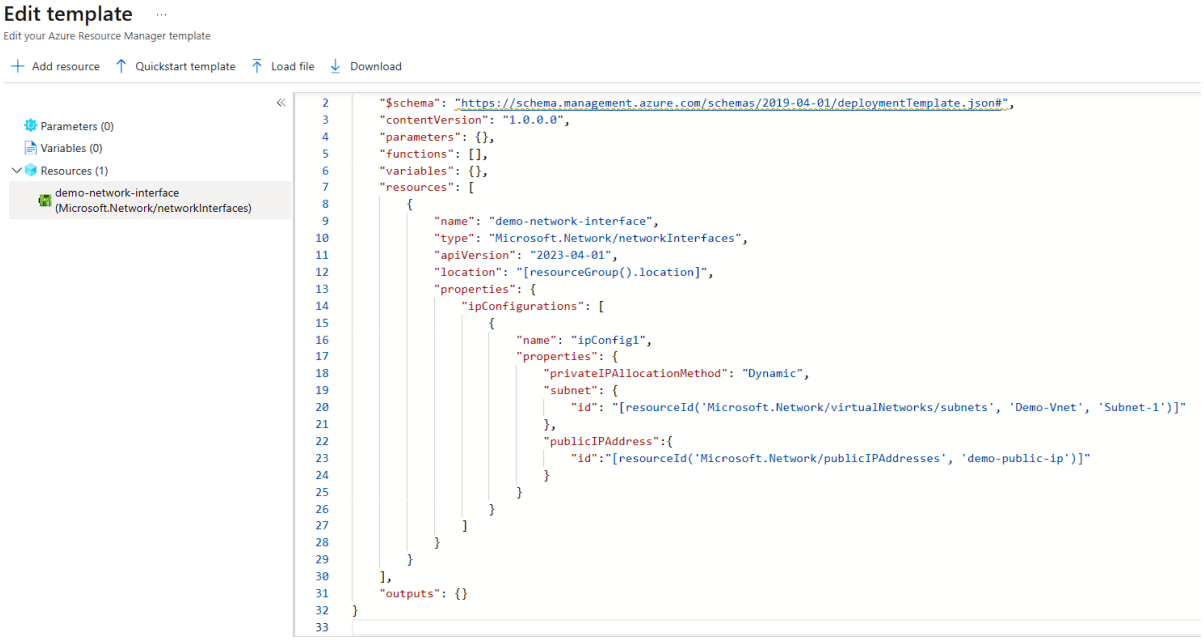
1. In previous labs, we have created a virtual network and a public IP.
2. Now in this lab we are going to create a network interface for that Vnet and Public IP.
3. For first you are going to create a new template in VS Code then start your coding.
4. Below is the code which we are going to use to create a network interface.
5. Here in the subnet part, we have defined the ID or say the name of our virtual network and the name of the subnet.
6. Then we defined the public IP address also and told it what is its name.

Temp5.json

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    Select or create a parameter file...
    "parameters": {},
    "functions": [],
    "variables": {},
    "resources": [
        {
            "name": "demo-network-interface",
            "type": "Microsoft.Network/networkInterfaces",
            "apiVersion": "2023-04-01",
            "location": "[resourceGroup().location]",
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipConfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Dynamic",
                            "subnet": {
                                "id": "[resourceId('Microsoft.Network/virtualNetworks/subnets', 'Demo-Vnet', 'Subnet-1')]"
                            },
                            "publicIPAddress":{
                                "id":"[resourceId('Microsoft.Network/publicIPAddresses', 'demo-public-ip')]"
                            }
                        }
                    }
                ]
            }
        }
    ],
    "outputs": {}
}
```

7. Now go to your Azure Portal and go to template deployment.

8. Here also you can see that it will create our network interface.



9. Now wait until the deployment is complete.
10. Once the deployment is completed. You can see that it is showing you that the network interface has been created successfully.

11. If you click on go to resource you can see that our Vnet, subnet and public IP has been attached successfully.