Contents

[Pre Requiste Setup : 2](#_Toc74650250)

[Create VM using powershell 3](#_Toc74650251)

[Create VM Using ARM Template 4](#_Toc74650252)

[Azure Virtual Network (VNet) 10](#_Toc74650253)

[NETWORK SECURITY GROUPS 12](#_Toc74650254)

[Additional Information 16](#_Toc74650255)

[Network Security Groups 17](#_Toc74650256)

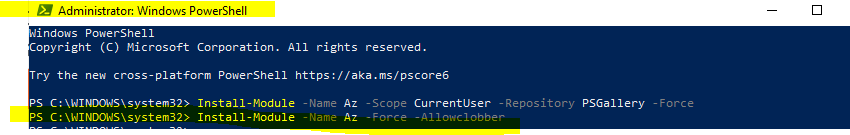
[Setup azure CLI 17](#_Toc74650257)

[Refrences : 18](#_Toc74650258)

Azure Resource Manager Templates provide simple and unified way to create and manage your Azure environments for both developers and administrators. With simple JSON files you can deploy your environment in consistent fashion and at great speed.

# Pre Requiste Setup :

1. Setting up Powershell :

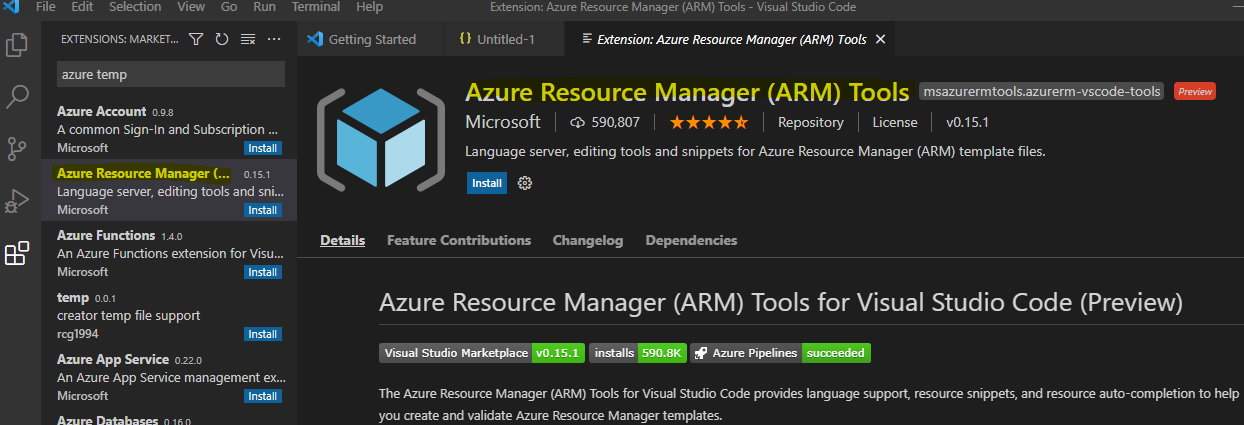


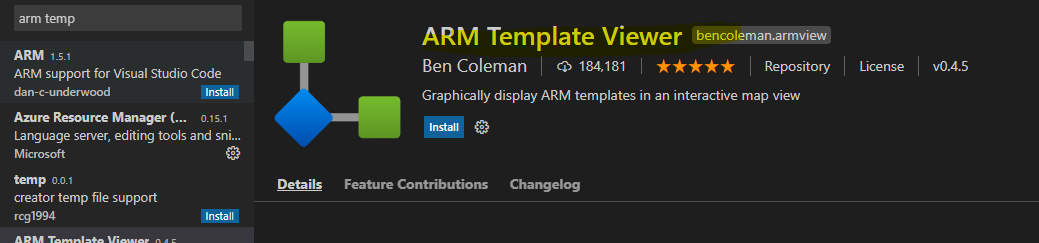
Run

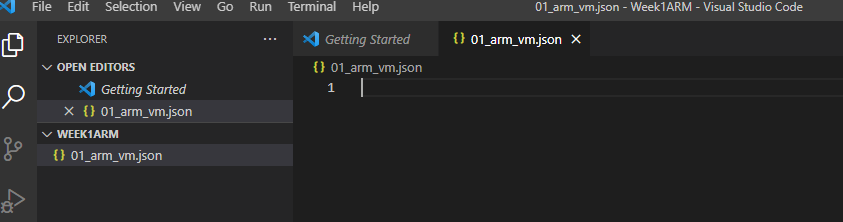


<https://docs.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-6.0.0>

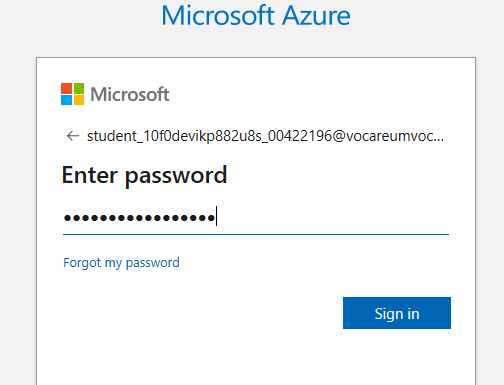
1. Open Visiual Studio Code and Install ARM tools and ARM Template Viewer

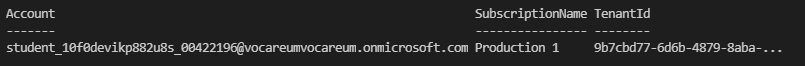




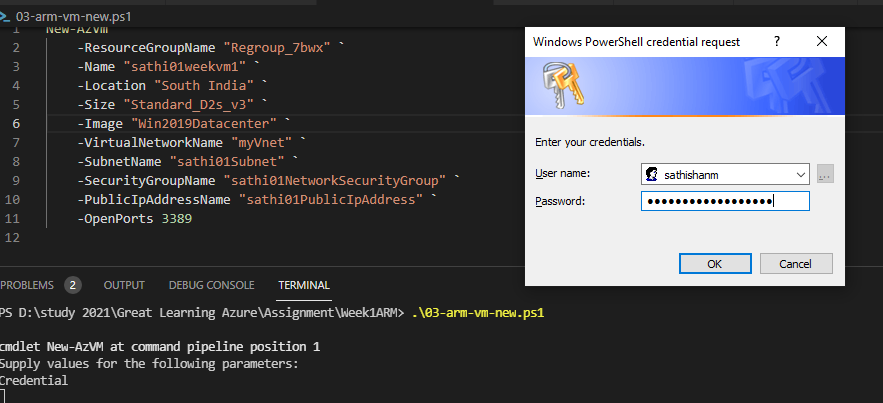
1. Create json file json file 
2. Test azure connection : connect-azaccount

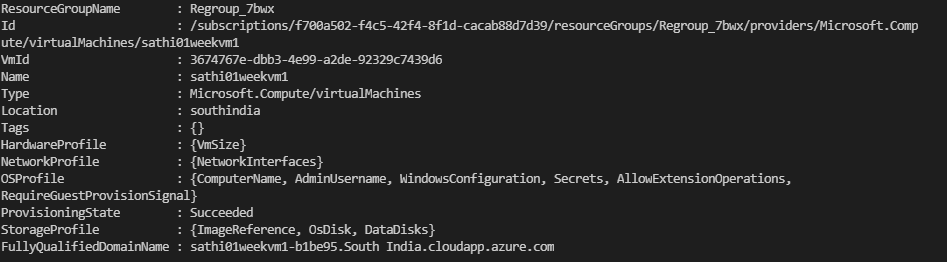


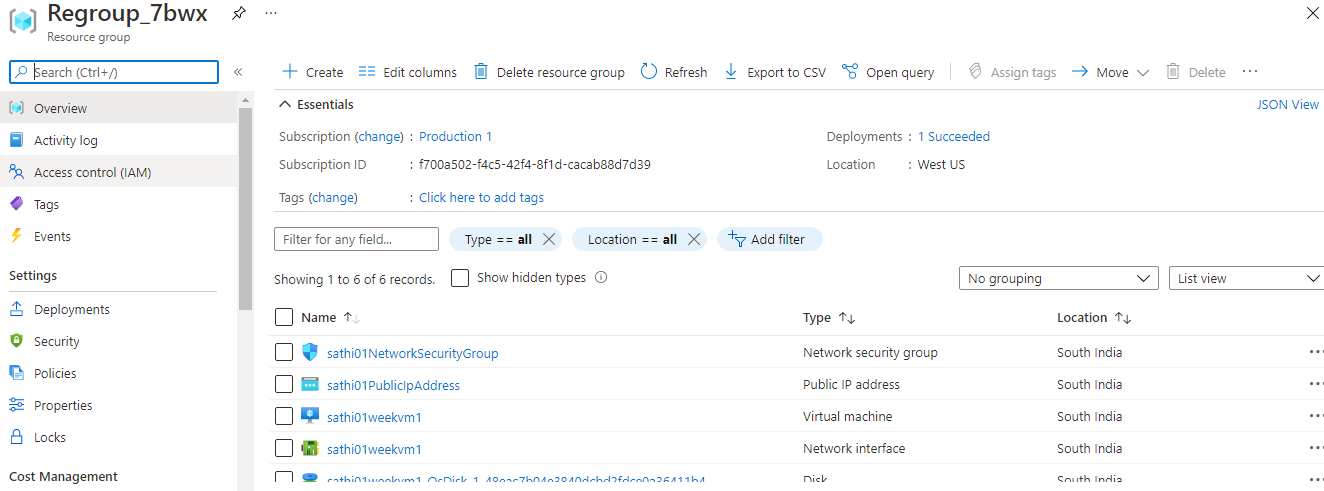




# Create VM using powershell







# Create VM Using ARM Template



{

  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

  "contentVersion": "1.0.0.0",

  "parameters": {

    "adminUsername": {

      "type": "string",

      "defaultValue": "sathishanm",

      "metadata": {

        "description": "Username for the Virtual Machine."

      }

    },

    "adminPassword": {

      "type": "securestring",

      "minLength": 12,

      "metadata": {

        "description": "Password for the Virtual Machine."

      }

    },

    "dnsLabelPrefix": {

      "type": "string",

      "defaultValue": "[toLower(concat(parameters('vmName'),'-', uniqueString(resourceGroup().id, parameters('vmName'))))]",

      "metadata": {

        "description": "Unique DNS Name for the Public IP used to access the Virtual Machine."

      }

    },

    "publicIpName": {

      "type": "string",

      "defaultValue": "myPublicIP",

      "metadata": {

        "description": "Name for the Public IP used to access the Virtual Machine."

      }

    },

    "publicIPAllocationMethod": {

      "type": "string",

      "defaultValue": "Dynamic",

      "allowedValues": [

        "Dynamic",

        "Static"

      ],

      "metadata": {

        "description": "Allocation method for the Public IP used to access the Virtual Machine."

      }

    },

    "publicIpSku": {

      "type": "string",

      "defaultValue": "Basic",

      "allowedValues": [

        "Basic",

        "Standard"

      ],

      "metadata": {

        "description": "SKU for the Public IP used to access the Virtual Machine."

      }

    },

    "OSVersion": {

      "type": "string",

      "defaultValue": "2019-Datacenter",

      "allowedValues": [

        "2008-R2-SP1",

        "2012-Datacenter",

        "2012-R2-Datacenter",

        "2016-Nano-Server",

        "2016-Datacenter-with-Containers",

        "2016-Datacenter",

        "2019-Datacenter",

        "2019-Datacenter-Core",

        "2019-Datacenter-Core-smalldisk",

        "2019-Datacenter-Core-with-Containers",

        "2019-Datacenter-Core-with-Containers-smalldisk",

        "2019-Datacenter-smalldisk",

        "2019-Datacenter-with-Containers",

        "2019-Datacenter-with-Containers-smalldisk"

      ],

      "metadata": {

        "description": "The Windows version for the VM. This will pick a fully patched image of this given Windows version."

      }

    },

    "vmSize": {

      "type": "string",

      "defaultValue": "Standard\_DS1\_v2",

      "metadata": {

        "description": "Size of the virtual machine."

      }

    },

    "location": {

      "type": "string",

      "defaultValue": "[resourceGroup().location]",

      "metadata": {

        "description": "Location for all resources."

      }

    },

    "vmName": {

      "type": "string",

      "defaultValue": "saweekdemovm",

      "metadata": {

        "description": "Name of the virtual machine."

      }

    }

  },

  "variables": {

    "storageAccountName": "[concat('bootdiags', uniquestring(resourceGroup().id))]",

    "nicName": "savnnickname",

    "addressPrefix": "10.0.0.0/16",

    "subnetName": "Subnet",

    "subnetPrefix": "10.0.0.0/24",

    "virtualNetworkName": "savirnetdemo",

    "subnetRef": "[resourceId('Microsoft.Network/virtualNetworks/subnets', variables('virtualNetworkName'), variables('subnetName'))]",

    "networkSecurityGroupName": "default-NSG"

  },

  "resources": [

    {

      "type": "Microsoft.Storage/storageAccounts",

      "apiVersion": "2019-06-01",

      "name": "[variables('storageAccountName')]",

      "location": "[parameters('location')]",

      "sku": {

        "name": "Standard\_LRS"

      },

      "kind": "Storage",

      "properties": {}

    },

    {

      "type": "Microsoft.Network/publicIPAddresses",

      "apiVersion": "2020-06-01",

      "name": "[parameters('publicIPName')]",

      "location": "[parameters('location')]",

      "sku": {

        "name": "[parameters('publicIpSku')]"

      },

      "properties": {

        "publicIPAllocationMethod": "[parameters('publicIPAllocationMethod')]",

        "dnsSettings": {

          "domainNameLabel": "[parameters('dnsLabelPrefix')]"

        }

      }

    },

    {

      "type": "Microsoft.Network/networkSecurityGroups",

      "apiVersion": "2020-06-01",

      "name": "[variables('networkSecurityGroupName')]",

      "location": "[parameters('location')]",

      "properties": {

        "securityRules": [

          {

            "name": "default-allow-3389",

            "properties": {

              "priority": 1000,

              "access": "Allow",

              "direction": "Inbound",

              "destinationPortRange": "3389",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*"

            }

          }

        ]

      }

    },

    {

      "type": "Microsoft.Network/virtualNetworks",

      "apiVersion": "2020-06-01",

      "name": "[variables('virtualNetworkName')]",

      "location": "[parameters('location')]",

      "dependsOn": [

        "[resourceId('Microsoft.Network/networkSecurityGroups', variables('networkSecurityGroupName'))]"

      ],

      "properties": {

        "addressSpace": {

          "addressPrefixes": [

            "[variables('addressPrefix')]"

          ]

        },

        "subnets": [

          {

            "name": "[variables('subnetName')]",

            "properties": {

              "addressPrefix": "[variables('subnetPrefix')]",

              "networkSecurityGroup": {

                "id": "[resourceId('Microsoft.Network/networkSecurityGroups', variables('networkSecurityGroupName'))]"

              }

            }

          }

        ]

      }

    },

    {

      "type": "Microsoft.Network/networkInterfaces",

      "apiVersion": "2020-06-01",

      "name": "[variables('nicName')]",

      "location": "[parameters('location')]",

      "dependsOn": [

        "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicIPName'))]",

        "[resourceId('Microsoft.Network/virtualNetworks', variables('virtualNetworkName'))]"

      ],

      "properties": {

        "ipConfigurations": [

          {

            "name": "ipconfig1",

            "properties": {

              "privateIPAllocationMethod": "Dynamic",

              "publicIPAddress": {

                "id": "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicIPName'))]"

              },

              "subnet": {

                "id": "[variables('subnetRef')]"

              }

            }

          }

        ]

      }

    },

    {

      "type": "Microsoft.Compute/virtualMachines",

      "apiVersion": "2020-06-01",

      "name": "[parameters('vmName')]",

      "location": "[parameters('location')]",

      "dependsOn": [

        "[resourceId('Microsoft.Storage/storageAccounts', variables('storageAccountName'))]",

        "[resourceId('Microsoft.Network/networkInterfaces', variables('nicName'))]"

      ],

      "properties": {

        "hardwareProfile": {

          "vmSize": "[parameters('vmSize')]"

        },

        "osProfile": {

          "computerName": "[parameters('vmName')]",

          "adminUsername": "[parameters('adminUsername')]",

          "adminPassword": "[parameters('adminPassword')]"

        },

        "storageProfile": {

          "imageReference": {

            "publisher": "MicrosoftWindowsServer",

            "offer": "WindowsServer",

            "sku": "[parameters('OSVersion')]",

            "version": "latest"

          },

          "osDisk": {

            "createOption": "FromImage",

            "managedDisk": {

              "storageAccountType": "StandardSSD\_LRS"

            }

          },

          "dataDisks": [

            {

              "diskSizeGB": 1023,

              "lun": 0,

              "createOption": "Empty"

            }

          ]

        },

        "networkProfile": {

          "networkInterfaces": [

            {

              "id": "[resourceId('Microsoft.Network/networkInterfaces', variables('nicName'))]"

            }

          ]

        },

        "diagnosticsProfile": {

          "bootDiagnostics": {

            "enabled": true,

            "storageUri": "[reference(resourceId('Microsoft.Storage/storageAccounts', variables('storageAccountName'))).primaryEndpoints.blob]"

          }

        }

      }

    }

  ],

  "outputs": {

    "hostname": {

      "type": "string",

      "value": "[reference(parameters('publicIPName')).dnsSettings.fqdn]"

    }

  }

}



$resourceGroupName = "Regroup\_1tbb"

$location = "westus"

$OSVersion = "2019-Datacenter"

$adminUsername = "sathishanm"

$adminPassword = Read-Host -Prompt "Enter the administrator password" -AsSecureString

$dnsLabelPrefix = "week1demoip"

New-AzResourceGroup -Name $resourceGroupName -Location "$location"

New-AzResourceGroupDeployment `

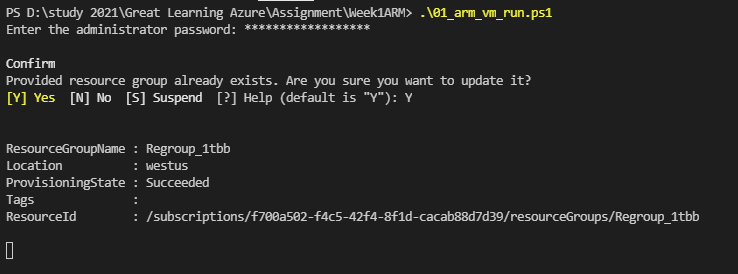
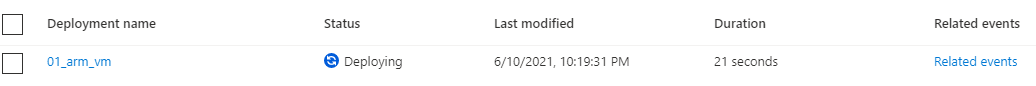
    -ResourceGroupName $resourceGroupName `

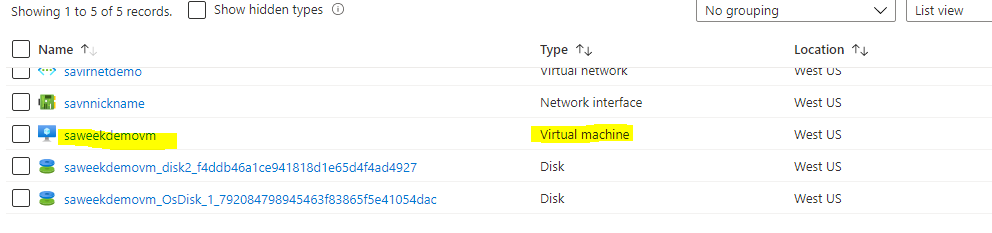
    -TemplateFile "01\_arm\_vm.json" `

    -adminUsername $adminUsername `

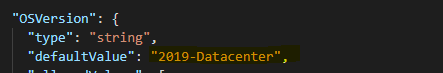
    -adminPassword $adminPassword `

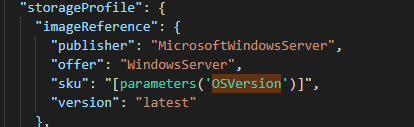
    -dnsLabelPrefix $dnsLabelPrefix

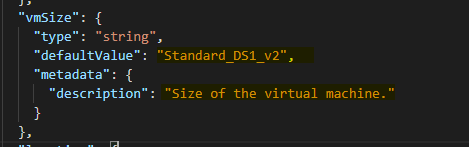
 



1. OS: Windows 2019 datacenter







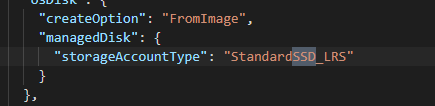
b) VM size: DS1\_v2



1. Open ports: 3389



1. Storage option: Standard SSD



# Azure Virtual Network (VNet)

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks.There is no charge for using Azure VNet, it is free of cost. Standard charges are applicable for resources, such as Virtual Machines (VMs)

ARM for creating Vnet

{

    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

    "contentVersion": "1.0.0.0",

    "parameters": {

        "virtualNetworks\_vnetsathidemo\_name": {

            "defaultValue": "vnetsathidemo",

            "type": "String"

        }

    },

    "variables": {},

    "resources": [

        {

            "type": "Microsoft.Network/virtualNetworks",

            "apiVersion": "2020-11-01",

            "name": "[parameters('virtualNetworks\_vnetsathidemo\_name')]",

            "location": "westus",

            "properties": {

                "addressSpace": {

                    "addressPrefixes": [

                        "10.1.0.0/16"

                    ]

                },

                "subnets": [

                    {

                        "name": "vsubnet",

                        "properties": {

                            "addressPrefix": "10.1.0.0/24",

                            "delegations": [],

                            "privateEndpointNetworkPolicies": "Enabled",

                            "privateLinkServiceNetworkPolicies": "Enabled"

                        }

                    },

                    {

                        "name": "AzureBastionSubnet",

                        "properties": {

                            "addressPrefix": "10.1.1.0/24",

                            "delegations": [],

                            "privateEndpointNetworkPolicies": "Enabled",

                            "privateLinkServiceNetworkPolicies": "Enabled"

                        }

                    }

                ],

                "virtualNetworkPeerings": [],

                "enableDdosProtection": false

            }

        },

        {

            "type": "Microsoft.Network/virtualNetworks/subnets",

            "apiVersion": "2020-11-01",

            "name": "[concat(parameters('virtualNetworks\_vnetsathidemo\_name'), '/AzureBastionSubnet')]",

            "dependsOn": [

                "[resourceId('Microsoft.Network/virtualNetworks', parameters('virtualNetworks\_vnetsathidemo\_name'))]"

            ],

            "properties": {

                "addressPrefix": "10.1.1.0/24",

                "delegations": [],

                "privateEndpointNetworkPolicies": "Enabled",

                "privateLinkServiceNetworkPolicies": "Enabled"

            }

        },

        {

            "type": "Microsoft.Network/virtualNetworks/subnets",

            "apiVersion": "2020-11-01",

            "name": "[concat(parameters('virtualNetworks\_vnetsathidemo\_name'), '/vsubnet')]",

            "dependsOn": [

                "[resourceId('Microsoft.Network/virtualNetworks', parameters('virtualNetworks\_vnetsathidemo\_name'))]"

            ],

            "properties": {

                "addressPrefix": "10.1.0.0/24",

                "delegations": [],

                "privateEndpointNetworkPolicies": "Enabled",

                "privateLinkServiceNetworkPolicies": "Enabled"

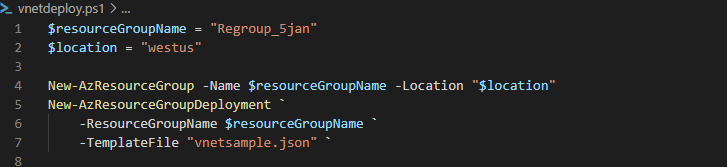
            }

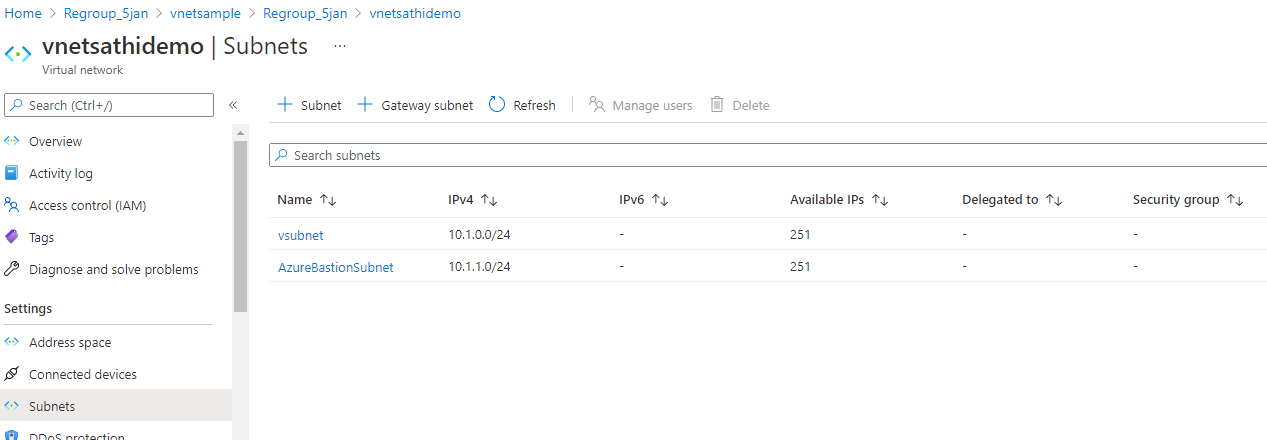
        }

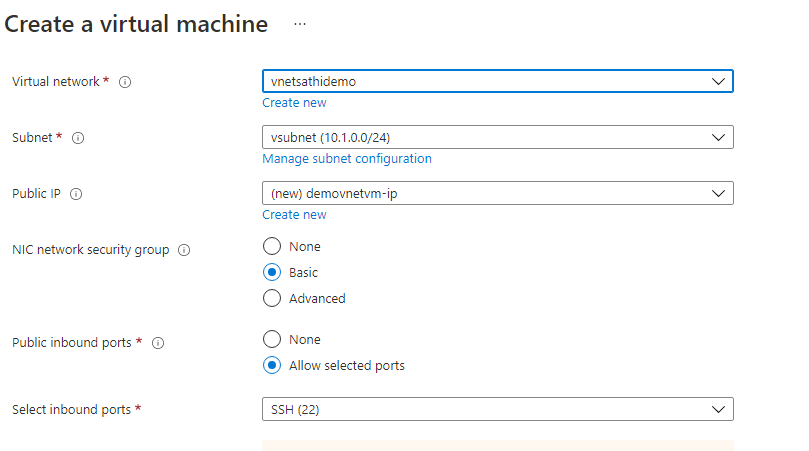
    ]

}

Deploy ARM







# NETWORK SECURITY GROUPS

Sample template for allwing RDP connection on 3389 port



ARM template

{

  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",

  "contentVersion": "1.0.0.0",

  "parameters": {

   "securityGroupName": {

      "type": "string",

      "metadata": {

        "description": "Name of the network security group"

      }

    }

  },

  "variables": {

    "location": "[resourceGroup().location]"

  },

  "resources": [

    {

      "apiVersion": "2015-05-01-preview",

      "type": "Microsoft.Network/networkSecurityGroups",

      "name": "[parameters('securityGroupName')]",

      "location": "[variables('location')]",

      "properties": {

        "securityRules": [

          {

            "name": "allow\_http",

            "properties": {

              "description": "Permit access to HTTP",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "destinationPortRange": "80",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 105,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_https",

            "properties": {

              "description": "Permit access to HTTPS",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "destinationPortRange": "443",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 110,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_h323cs",

            "properties": {

              "description": "Permit access to H.323 CS",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "destinationPortRange": "1720",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 115,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_sip\_tcp",

            "properties": {

              "description": "Permit access to SIP/TCP",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "destinationPortRange": "5060",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 120,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_sip\_tls",

            "properties": {

              "description": "Permit access to SIP/TLS",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "destinationPortRange": "5061",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 125,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_signalling\_tcp",

            "properties": {

              "description": "Permit access to ephemeral TCP call signalling ports",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "destinationPortRange": "33000-39999",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 130,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_media\_tcp",

            "properties": {

              "description": "Permit access to ephemeral TCP media ports",

              "protocol": "Tcp",

              "sourcePortRange": "\*",

              "destinationPortRange": "40000-49999",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 135,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_h323ls",

            "properties": {

              "description": "Permit access to H.323 LS",

              "protocol": "Udp",

              "sourcePortRange": "\*",

              "destinationPortRange": "1719",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 140,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_signalling\_udp",

            "properties": {

              "description": "Permit access to ephemeral UDP call signalling ports",

              "protocol": "Udp",

              "sourcePortRange": "\*",

              "destinationPortRange": "33000-39999",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 150,

              "direction": "Inbound"

            }

          },

          {

            "name": "allow\_media\_udp",

            "properties": {

              "description": "Permit access to ephemeral UDP media ports",

              "protocol": "Udp",

              "sourcePortRange": "\*",

              "destinationPortRange": "40000-49999",

              "sourceAddressPrefix": "\*",

              "destinationAddressPrefix": "\*",

              "access": "Allow",

              "priority": 155,

              "direction": "Inbound"

            }

          }

        ]

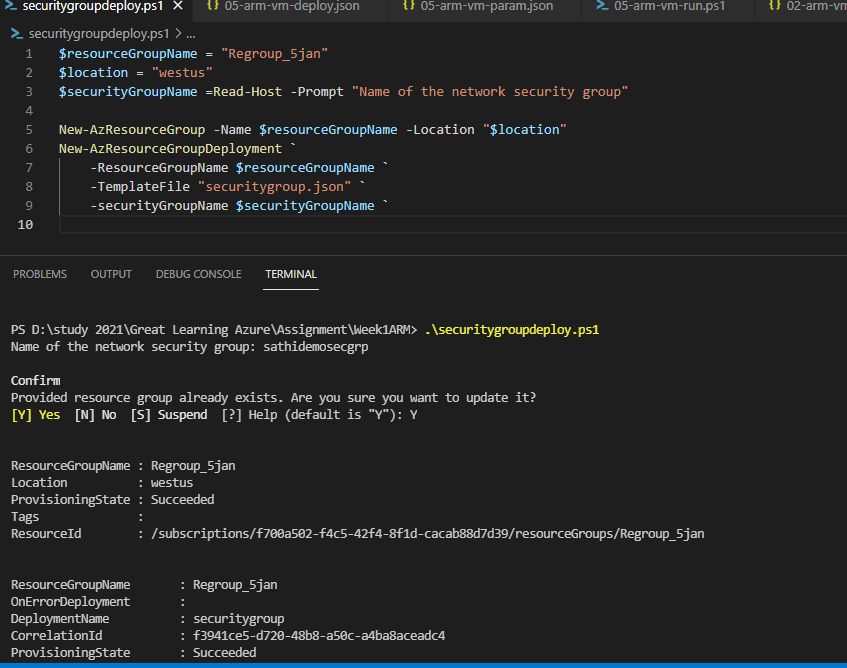
      }

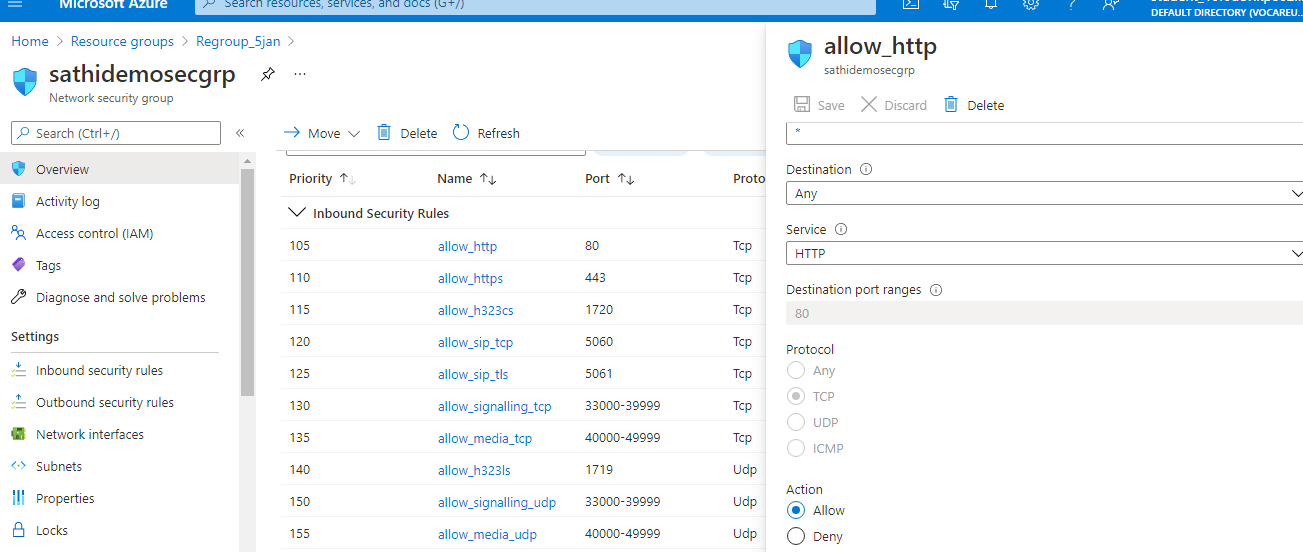
    }

  ]

}

Deploy



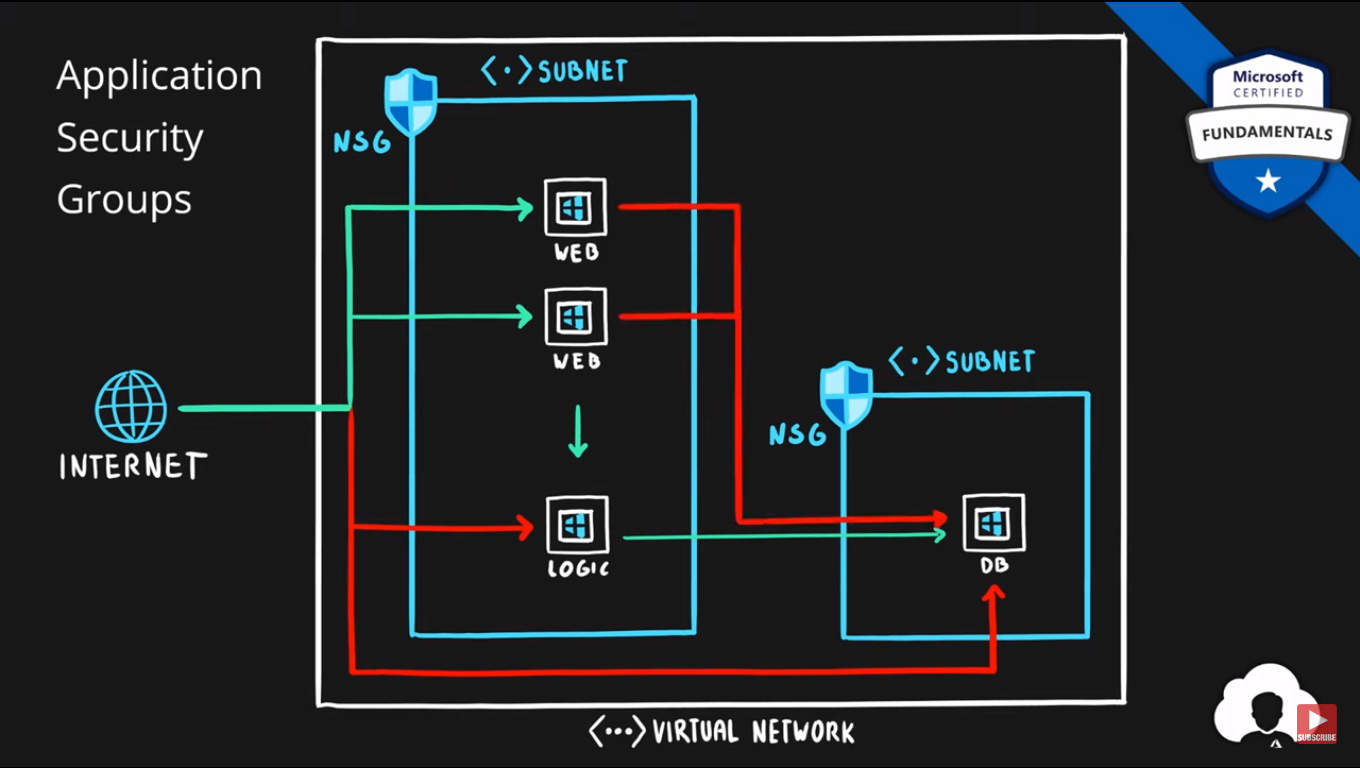


# Additional Information

### Network Security Groups

# 

Green – allow red – do not allow



## Setup azure CLI

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?view=azure-cli-latest&tabs=azure-powershell>

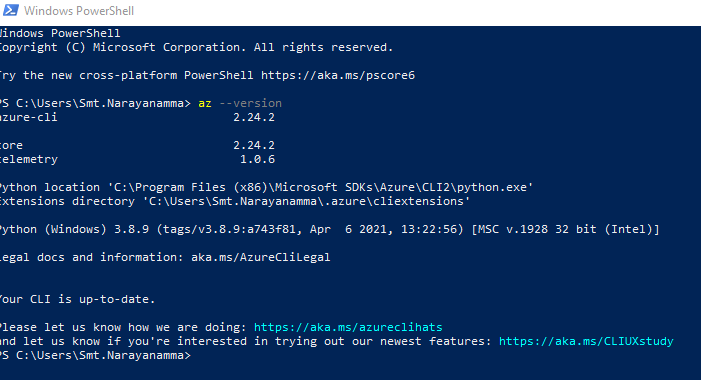
<https://aka.ms/installazurecliwindows>

or

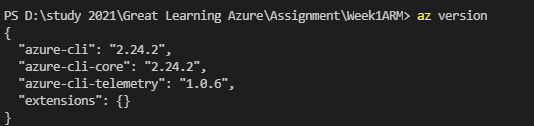
run in powershell

*Invoke-WebRequest -Uri* [*https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi*](https://aka.ms/installazurecliwindows%20-OutFile%20.\AzureCLI.msi)*; Start-Process msiexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi*

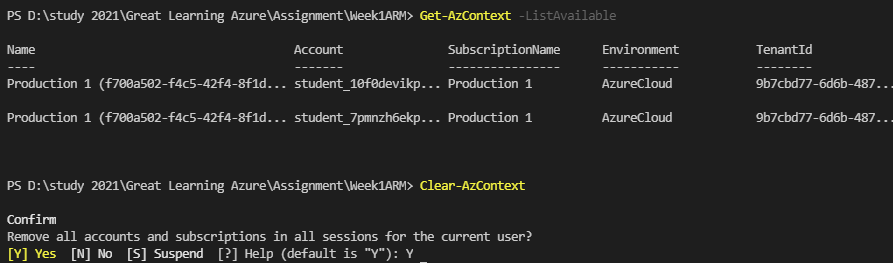
After Installation , Verify version in power shell



And in vs code



List and clear contexts



# Refrences :

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-powershell>

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/ps-template>

<https://docs.microsoft.com/en-us/azure/templates/microsoft.network/virtualnetworks?tabs=json>

<https://docs.microsoft.com/en-us/azure/templates/microsoft.network/networksecuritygroups?tabs=json>