Azure cli commands

Install azure cli

Azure cli version

PS C:\Windows\system32> az version

PS C:\Windows\system32> az login

PS C:\Windows\system32> az login Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136 Retrieving tenants and subscriptions for the selection... Authentication failed against tenant d822fb16-73db-404f-a2c6-82903ce57bc8 'Default Directory': (pii). Status: Response\_Status.Status\_InteractionRequired, Error code: 3399614476, Tag: 557973645 If you need to access subscriptions in the following tenants, please use `az login --tenant TENANT\_ID`

d822fb16-73db-404f-a2c6-82903ce57bc8 'Default Directory' No subscriptions found for

deepanshuchaudhry852@gmail.com. PS C:\Windows\system32> az login --tenant d822fb16-73db-404f-a2c6-82903ce57bc8 Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136 Retrieving subscriptions for the selection... [Tenant and subscription selection]

$ login complete

PS C:\Windows\system32> az group create --name learning-azure-cli-eastus --location eastus

{

"id": "/subscriptions/6449dbeb-a056-4321-a82e-ce6acd3048ab/resourceGroups/learning-azure-cli-eastus",

"location": "eastus",

"managedBy": null,

"name": "learning-azure-cli-eastus",

"properties": {

"provisioningState": "Succeeded"

},

"tags": null,

"type": "Microsoft.Resources/resourceGroups"

}

# creating a resource group

PS C:\Windows\system32> az vm create --resource-group learning-azure-cli-eastus --name myVM --image Ubuntu2204 --admin-username azureuser --admin-password MyPassword123 --size Standard\_B1ms

{

"fqdns": "",

"id": "/subscriptions/6449dbeb-a056-4321-a82e-ce6acd3048ab/resourceGroups/learning-azure-cli-eastus/providers/Microsoft.Compute/virtualMachines/myVM",

"location": "eastus",

"macAddress": "7C-ED-8D-22-11-F6",

"powerState": "VM running",

"privateIpAddress": "10.0.0.4",

"publicIpAddress": "172.190.187.162",

"resourceGroup": "learning-azure-cli-eastus",

"zones": ""

}

PS C:\Windows\system32> az vm create --resource-group learning-azure-cli-eastus --name myVM --image Ubuntu2204 --admin-username azureuser --admin-password MyPassword123 --size Standard\_B1ms

{

"fqdns": "",

"id": "/subscriptions/6449dbeb-a056-4321-a82e-ce6acd3048ab/resourceGroups/learning-azure-cli-eastus/providers/Microsoft.Compute/virtualMachines/myVM",

"location": "eastus",

"macAddress": "7C-ED-8D-22-11-F6",

"powerState": "VM running",

"privateIpAddress": "10.0.0.4",

"publicIpAddress": "172.190.187.162",

"resourceGroup": "learning-azure-cli-eastus",

"zones": ""

}

az vm create ` >> --resource-group learning-azure-cli ` >> --name myVM2 ` >> --image Ubuntu2204 ` >> --admin-username azureuser ` >> --generate-ssh-keys ` >> --size Standard\_B2s ` >> --location westus ` >> --public-ip-sku Standard

Message: Resource group 'learning-azure-cli' could not be found.

PS C:\Windows\system32> az vm create --resource-group learning-azure-cli-eastus --name myVM2 --image Ubuntu2204 --admin-username azureuser --generate-ssh-keys --size Standard\_B2s --location westus --public-ip-sku Standard

{

"fqdns": "",

"id": "/subscriptions/6449dbeb-a056-4321-a82e-ce6acd3048ab/resourceGroups/learning-azure-cli-eastus/providers/Microsoft.Compute/virtualMachines/myVM2",

"location": "westus",

"macAddress": "00-0D-3A-35-CF-F2",

"powerState": "VM running",

"privateIpAddress": "10.0.0.4",

"publicIpAddress": "172.184.200.60",

"resourceGroup": "learning-azure-cli-eastus",

"zones": ""

}

PS C:\Windows\system32> az vm create ` >> --resource-group learning-azure-cli-eastus ` >> --name myVM3 ` >> --image Ubuntu2204 ` >> --admin-username azureuser ` >> --generate-ssh-keys ` >> --size Standard\_B2s ` >> --location westus ` >> --public-ip-sku Standard

az vm create `

--resource-group learning-azure-cli `

--name myVM23 `

--image Ubuntu2204 `

--admin-username azureuser `

--generate-ssh-keys `

--size Standard\_B2s `

--location westus `

--public-ip-sku Standard

My vm has been created inside the resource gp learning-azure-cli  
#

To create diff resources such as vm subnets storage accoutns - we will use azure resourece manager Create resrouce with micrososft azure

https://github.com/iam-veeramalla/Azure-zero-to-hero/blob/main/Day-11/01-storage-account/01- storage-account.json

az storage account create `

--name mystorageaccount2321 `

--resource-group learning-azure-cli `

--location westus `

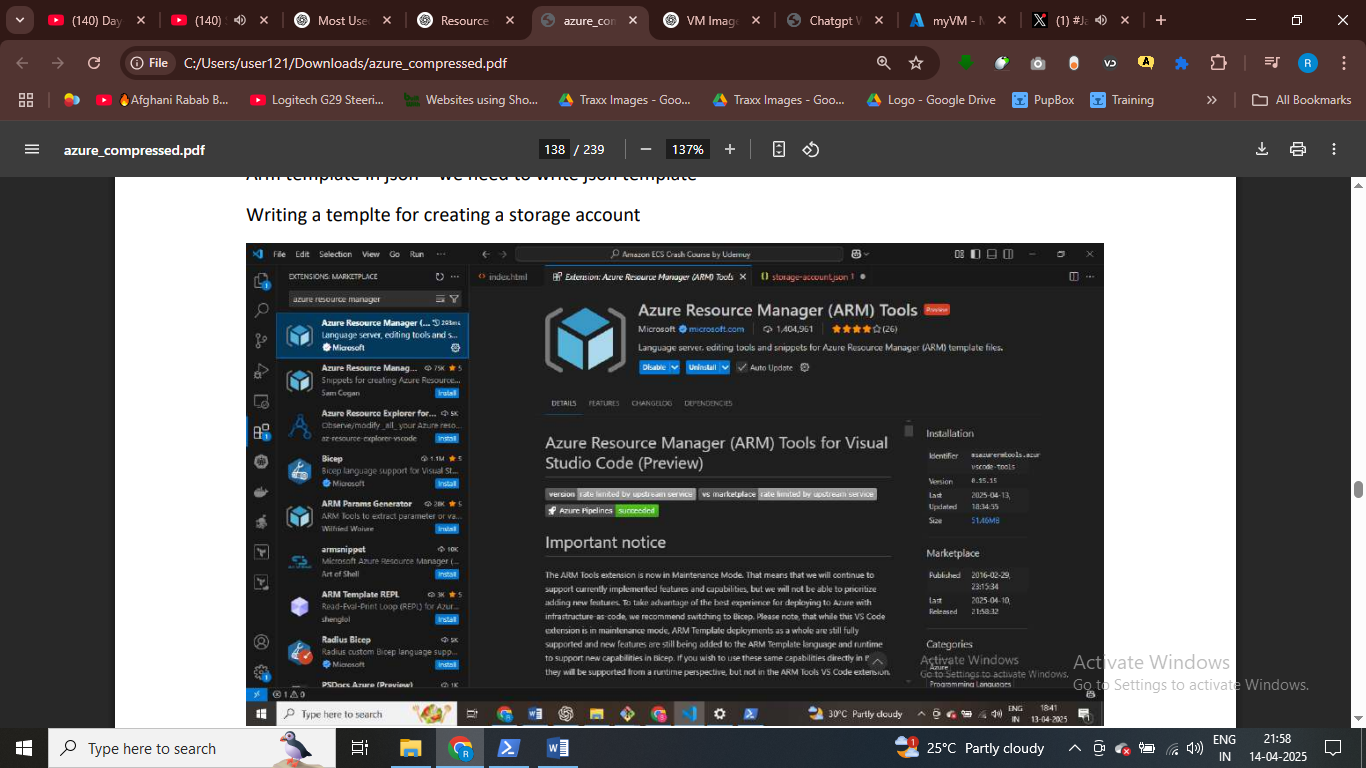
--sku Standard\_LRS `

--kind StorageV2 `

--access-tier Hot

Arm template in json – we need to write json template Writing a templte for creating a storage account

138 in pdf



user121@DESKTOP-JFJ21I5 MINGW64 /d/Amazon ECS Crash Course by Udemuy

$ az group create --name vscode1 --location 'east US'

{

"id": "/subscriptions/6449dbeb-a056-4321-a82e-ce6acd3048ab/resourceGroups/vscode1",

"location": "eastus",

"managedBy": null,

"name": "vscode1",

"properties": {

"provisioningState": "Succeeded"

},

"tags": null,

"type": "Microsoft.Resources/resourceGroups"

}

{

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

  "contentVersion": "1.0.0.0",

  "parameters": {},

  "functions": [],

  "variables": {},

  "resources": [

    {

      "name": "deepanshu23456",

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

      "apiVersion": "2023-05-01",

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

      "kind": "StorageV2",

      "sku": {

        "name": "Premium\_LRS",

        "tier": "Premium"

      }

    }

  ],

  "outputs": {}

}

user121@DESKTOP-JFJ21I5 MINGW64 /d/Amazon ECS Crash Course by Udemuy

$ az deployment group create --resource-group vscode1 --template-file storage.json

{

"id": "/subscriptions/6449dbeb-a056-4321-a82e-ce6acd3048ab/resourceGroups/vscode1/providers/Microsoft.Resources/deployments/storage",

"location": null,

"name": "storage",

"properties": {

"correlationId": "35c2aea1-b573-4354-abb6-88617ebc39f4",

"debugSetting": null,

"dependencies": [],

"duration": null,

"error": null,

"mode": "Incremental",

"onErrorDeployment": null,

"outputResources": [

{

"id": "/subscriptions/6449dbeb-a056-4321-a82e-ce6acd3048ab/resourceGroups/vscode1/providers/Microsoft.Storage/storageAccounts/deepanshu23456",

"resourceGroup": "vscode1"

}

Variable resusiblity of code – multiple field in this template

# arm template for the app gateway we need to create it Functions

– we have customize any logic – generate the variable at runtime – generate dynamic storage acc name - use function for it Every time user exec this – diff storage name we want – we can use fucn

**Correct Resource Group**: Verify that the vscode1 resource group exists in your Azure subscription. You can use the command:

bash

CopyEdit

az group show --name vscode1

If it doesn't exist, create the resource group first:

bash

CopyEdit

az group create --name vscode1 --location <your-location>

{

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

    "contentVersion": "1.0.0.0",

    "parameters": {},

    "functions": [],

    "variables": {},

    "resources": [

        {

            "name": "[toLower('ubuntuVM1storageaccount')]",

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

            "apiVersion": "2023-01-01",

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

            "tags": {

                "displayName": "ubuntuVM1 Storage Account"

            },

            "sku": {

                "name": "Standard\_LRS"

            },

            "kind": "Storage"

        },

        {

            "name": "ubuntuVM1-PublicIP",

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

            "apiVersion": "2023-04-01",

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

            "tags": {

                "displayName": "PublicIPAddress"

            },

            "properties": {

                "publicIPAllocationMethod": "Dynamic",

                "dnsSettings": {

                    "domainNameLabel": "[toLower('ubuntuVM1unique')]"

                }

            }

        },

        {

            "name": "ubuntuVM1-nsg",

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

            "apiVersion": "2023-04-01",

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

            "properties": {

                "securityRules": [

                    {

                        "name": "nsgRule1",

                        "properties": {

                            "description": "description",

                            "protocol": "Tcp",

                            "sourcePortRange": "\*",

                            "destinationPortRange": "22",

                            "sourceAddressPrefix": "\*",

                            "destinationAddressPrefix": "\*",

                            "access": "Allow",

                            "priority": 100,

                            "direction": "Inbound"

                        }

                    }

                ]

            }

        },

        {

            "name": "ubuntuVM1-VirtualNetwork",

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

            "apiVersion": "2023-04-01",

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

            "dependsOn": [

                "[resourceId('Microsoft.Network/networkSecurityGroups', 'ubuntuVM1-nsg')]"

            ],

            "tags": {

                "displayName": "ubuntuVM1-VirtualNetwork"

            },

            "properties": {

                "addressSpace": {

                    "addressPrefixes": [

                        "10.0.0.0/16"

                    ]

                },

                "subnets": [

                    {

                        "name": "ubuntuVM1-VirtualNetwork-Subnet",

                        "properties": {

                            "addressPrefix": "10.0.0.0/24",

                            "networkSecurityGroup": {

                                "id": "[resourceId('Microsoft.Network/networkSecurityGroups', 'ubuntuVM1-nsg')]"

                            }

                        }

                    }

                ]

            }

        },

        {

            "name": "ubuntuVM1-NetworkInterface",

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

            "apiVersion": "2023-04-01",

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

            "dependsOn": [

                "[resourceId('Microsoft.Network/publicIPAddresses', 'ubuntuVM1-PublicIP')]",

                "[resourceId('Microsoft.Network/virtualNetworks', 'ubuntuVM1-VirtualNetwork')]"

            ],

            "tags": {

                "displayName": "ubuntuVM1-NetworkInterface"

            },

            "properties": {

                "ipConfigurations": [

                    {

                        "name": "ipConfig1",

                        "properties": {

                            "privateIPAllocationMethod": "Dynamic",

                            "publicIPAddress": {

                                "id": "[resourceId('Microsoft.Network/publicIPAddresses', 'ubuntuVM1-PublicIP')]"

                            },

                            "subnet": {

                                "id": "[resourceId('Microsoft.Network/virtualNetworks/subnets', 'ubuntuVM1-VirtualNetwork', 'ubuntuVM1-VirtualNetwork-Subnet')]"

                            }

                        }

                    }

                ]

            }

        },

        {

            "name": "ubuntuVM1",

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

            "apiVersion": "2023-03-01",

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

            "dependsOn": [

                "[resourceId('Microsoft.Network/networkInterfaces', 'ubuntuVM1-NetworkInterface')]"

            ],

            "tags": {

                "displayName": "ubuntuVM1"

            },

            "properties": {

                "hardwareProfile": {

                    "vmSize": "Standard\_A2\_v2"

                },

                "osProfile": {

                    "computerName": "ubuntuVM1",

                    "adminUsername": "azureuser",

                    "adminPassword": "azureuser@123"

                },

                "storageProfile": {

                    "imageReference": {

                        "publisher": "Canonical",

                        "offer": "UbuntuServer",

                        "sku": "16.04-LTS",

                        "version": "latest"

                    },

                    "osDisk": {

                        "name": "ubuntuVM1-OSDisk",

                        "caching": "ReadWrite",

                        "createOption": "FromImage"

                    }

                },

                "networkProfile": {

                    "networkInterfaces": [

                        {

                            "id": "[resourceId('Microsoft.Network/networkInterfaces', 'ubuntuVM1-NetworkInterface')]"

                        }

                    ]

                },

                "diagnosticsProfile": {

                    "bootDiagnostics": {

                        "enabled": true,

                        "storageUri": "[reference(resourceId('Microsoft.Storage/storageAccounts', toLower('ubuntuVM1storageaccount')), '2023-01-01').primaryEndpoints.blob]"

                    }

                }

            }

        }

    ],

    "outputs": {}

}

user121@DESKTOP-JFJ21I5 MINGW64 /d/Amazon ECS Crash Course by Udemuy

$ az deployment group create --resource-group vscode1 --template-file vm.json

I now cli and terraform – for azure – arm templates – terraform you can also use

Bicep is repaalceing arm template

Arm template container – parameter , content , variables , outputs

Bicep is a domain-specific language (DSL) for deploying Azure resources declaratively. It simplifies the authoring of Azure Resource Manager (ARM) templates by providing a more concise and readable syntax while maintaining full compatibility with ARM templates.

 **Bicep CLI**: The Bicep CLI allows you to compile and deploy Bicep files.

* Install it using the following command:

bash

CopyEdit

az bicep install

 **Visual Studio Code** (VS Code) and **Bicep Extension**: For editing and syntax highlighting in Bicep.

* Install the extension from the [VS Code marketplace](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-bicep).

<https://github.com/Azure/bicep/releases>

win-64.exe

how to create first bicep file with resoruce gp learn-azure-cli

ChatGPT said:

To create your first Bicep file that deploys an Azure resource group, follow these steps:

Step 1: Create a New Bicep File

Open a text editor (like Visual Studio Code, Notepad++, or even Notepad).

Create a new file and save it with the .bicep extension (e.g., myResourceGroup.bicep).

Step 2: Write Your First Bicep Code

A basic Bicep file to create a resource group looks like this:

bicep

Copy

Edit

resource myResourceGroup 'Microsoft.Resources/resourceGroups@2021-04-01' = {

  name: 'learn-azure-cli'  // Name of the resource group

  location: 'East US'      // Location where the resource group will be created

}

Explanation:

resource: This defines a resource in Azure. In this case, it's a resource group.

myResourceGroup: This is the logical name of the resource.

'Microsoft.Resources/resourceGroups@2021-04-01': This is the Azure resource provider and API version used for the resource group.

name: The name of the resource group (learn-azure-cli).

location: The region where the resource group will be deployed (in this example, East US).

Step 3: Save Your Bicep File

Save the file as myResourceGroup.bicep.

Step 4: Build and Deploy the Bicep File

Open Azure CLI (PowerShell, Command Prompt, or the integrated terminal in Visual Studio Code).

Login to Azure (if not already logged in):

bash

Copy

Edit

az login

Build the Bicep file into an ARM template (this step is optional, but it's good to verify):

bash

Copy

Edit

bicep build myResourceGroup.bicep

This will generate a myResourceGroup.json file, which is the equivalent ARM template.

Deploy the Bicep file to Azure:

bash

Copy

Edit

az deployment sub create --location eastus --template-file myResourceGroup.bicep

az deployment sub create: This command deploys the resources at the subscription level.

--location eastus: The region where the deployment will occur.

--template-file myResourceGroup.bicep: Path to your Bicep file.

Step 5: Verify the Deployment

You can check if the resource group was successfully created by navigating to the Azure portal or by using the Azure CLI:

bash

Copy

Edit

az group show --name learn-azure-cli

This will display the details of the resource group learn-azure-cli if it was created successfully.

This is how you can create your first Bicep file to deploy a resource group. Let me know if you need further assistance!

You said:

create storage account in this resource gp using bicep file

ChatGPT said:

To create a Storage Account in the resource group learn-azure-cli that you created earlier, you can extend your Bicep file to include the Microsoft.Storage/storageAccounts resource. Below is an example of how you can update your Bicep file to create both the resource group and the storage account.

Step 1: Update Your Bicep File

You can modify your myResourceGroup.bicep file to include both the resource group and the storage account. Here's the updated Bicep code:

bicep

Copy

Edit

// Define the resource group

resource myResourceGroup 'Microsoft.Resources/resourceGroups@2021-04-01' = {

  name: 'learn-azure-cli'  // Name of the resource group

  location: 'East US'      // Location where the resource group will be created

}

// Define the storage account inside the resource group

resource myStorageAccount 'Microsoft.Storage/storageAccounts@2021-09-01' = {

  name: 'mystorageaccount01'  // Unique name for the storage account (must be globally unique)

  location: myResourceGroup.location  // Use the location of the resource group

  sku: {

    name: 'Standard\_LRS'  // Storage account type (Standard\_LRS for locally redundant storage)

  }

  kind: 'StorageV2'  // Specifies the type of storage account (General-purpose v2)

}

create a vm in this resource gp

ChatGPT said:

To create a Virtual Machine (VM) in the learn-azure-cli resource group, we will need to add the resource definition for the VM to the Bicep file. Below is an updated version of your Bicep file that creates a VM in the same resource group.

Step 1: Updated Bicep File

Here's how you can extend your Bicep file to include the creation of a Virtual Machine (Microsoft.Compute/virtualMachines):

bicep

Copy

Edit

// Define the resource group

resource myResourceGroup 'Microsoft.Resources/resourceGroups@2021-04-01' = {

  name: 'learn-azure-cli'  // Name of the resource group

  location: 'East US'      // Location where the resource group will be created

}

// Define the storage account

resource myStorageAccount 'Microsoft.Storage/storageAccounts@2021-09-01' = {

  name: 'mystorageaccount01'  // Unique name for the storage account (must be globally unique)

  location: myResourceGroup.location  // Use the location of the resource group

  sku: {

    name: 'Standard\_LRS'  // Storage account type (Standard\_LRS for locally redundant storage)

  }

  kind: 'StorageV2'  // Specifies the type of storage account (General-purpose v2)

}

// Define the virtual network

resource myVnet 'Microsoft.Network/virtualNetworks@2021-03-01' = {

  name: 'myVnet'

  location: myResourceGroup.location

  properties: {

    addressSpace: {

      addressPrefixes: [

        '10.0.0.0/16'

      ]

    }

  }

}

// Define the subnet

resource mySubnet 'Microsoft.Network/virtualNetworks/subnets@2021-03-01' = {

  name: 'mySubnet'

  parent: myVnet

  properties: {

    addressPrefix: '10.0.0.0/24'

  }

}

// Define the network interface (NIC)

resource myNic 'Microsoft.Network/networkInterfaces@2021-03-01' = {

  name: 'myNic'

  location: myResourceGroup.location

  properties: {

    ipConfigurations: [

      {

        name: 'ipconfig1'

        properties: {

          subnet: {

            id: mySubnet.id

          }

          privateIPAllocationMethod: 'Dynamic'

        }

      }

    ]

  }

}

// Define the virtual machine

resource myVM 'Microsoft.Compute/virtualMachines@2021-07-01' = {

  name: 'myVM'

  location: myResourceGroup.location

  properties: {

    hardwareProfile: {

      vmSize: 'Standard\_B1s'  // Size of the VM

    }

    osProfile: {

      computerName: 'myVM'

      adminUsername: 'azureuser'  // Username for the VM

      adminPassword: 'YourPassword123!'  // Password for the VM

    }

    storageProfile: {

      imageReference: {

        publisher: 'MicrosoftWindowsServer'

        offer: 'WindowsServer'

        sku: '2019-Datacenter'

        version: 'latest'

      }

      osDisk: {

        name: 'myOSDisk'

        createOption: 'FromImage'

        managedDisk: {

          storageAccountType: 'Standard\_LRS'

        }

      }

    }

    networkProfile: {

      networkInterfaces: [

        {

          id: myNic.id

        }

      ]

    }

  }

}