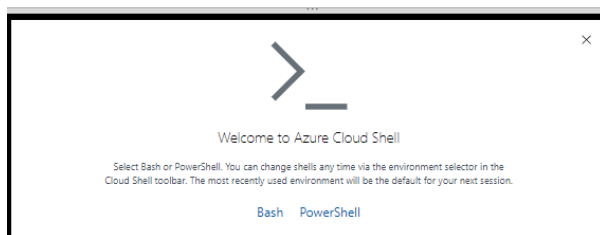


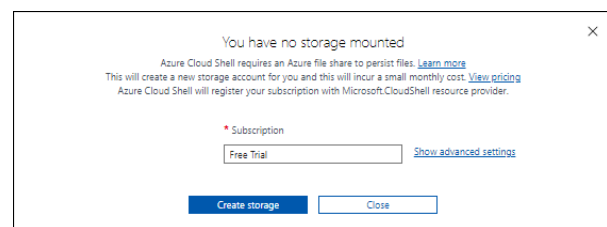


Azure Section 7 ARM Template

Before going to the ARM template we will first set our azure CLI. For that we need to click on the Azure cli option which is first on the top right corner. It looks like command line interface. Here after clicking on Azure cli option choose bash.



Going forward choose the subscription after you choose bash.



We need to mount the storage to store our commands and our work. This will create a resource group which i will show you further.

So here we can see that there was this resource group which we created it automatically. Its because of we created a storage under the free subscription.



Now we will see what arm template is.

ARM Template:

Stands for Azure Resource management template. This is written in Json format. This is used by azure in most of the deployments. When we create any resource in azure then azure create arm template at the end of the creation and passes it on to the deployment engine of azure.

We modify export and again deploy the modified file as well! It can also be created from scratch.

The arm template is declarative. Which means that it tells us what will be the end result look like.

Hands on ARM template:

Download the arm template from the git repository which we used for getting the code.

here is the link to the repository <https://github.com/AbhinavDeodhar/PracticeAppAzure.git>

- Now download it and unzip it.
- The arm template has three main parts Parameters, Variables and resources.

- Resources part contains all the resources that are part of the creation of virtual machine.
- If you go to the resource file code then there we can see the attributes which are associated with the machine such as its name, location, storage account type, disk etc. for example.
- And all the attributes of the machine are gets their values from the parameter file.
- It means if its name of the virtual machine written in the resource json file we will get its value directed to the parameters file.
- If we go to the parameters file then there is all the values which are used for the virtual machine creation.
- What ever values we will enter in the parameters files will effect the creation of virtual machine.

```

"name": "[parameters('virtualNetworkName')]",
"type": "Microsoft.Network/virtualNetworks",
"apiVersion": "2019-09-01",
"location": "[parameters('location')]",
"properties": {
  "addressSpace": {
    "addressPrefixes": "[parameters('addressPrefixes')]"
  },
  "subnets": "[parameters('subnets')]"
}

```

```

() parameters.json > ...
14     "networkSecurityGroupRules": {
15         "value": [
16             {
17                 "name": "RDP",
18                 "properties": {
19                     "priority": 300,

```

So template attributes get their values from parameters. It can be seen here.

The arm template cannot be edited in the azure portal there it is opened as read only mode. We need to download the arm template and then open in VScode.

There we can edit it and then upload to the Azure portal.

Going forward we will open the bash azure CLI and then deploy this edited arm template to create the virtual machine with the changes which we made in arm template.

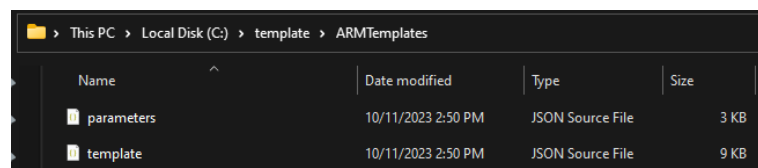
▼ Step 1:

Download arm template

Unzip the arm template file.

Store the arm template file and the parameter file in the C drive by creating the template folder.

Open VSCode in the folder where the arm template file and parameter file is located.



Name	Date modified	Type	Size
parameters	10/11/2023 2:50 PM	JSON Source File	3 KB
template	10/11/2023 2:50 PM	JSON Source File	9 KB

▼ Step 2:

Go to the OS disk section in parameters.

Make changes in its value by saying that standardSSD_LRS. If we just write Standard then azure will create HDD drive.

Scroll below and go the username and password section and change its value to whatever username and password it wants.

```
},
  "adminUsername": {
    "value": "Abhinav"
  },
  "adminPassword": {
    "value": " "
  }
}
```

```
},
  "osDiskType": {
    "value": "StandardSSD_LRS"
  },
  "virtualMachineSize": {
    "value": "Standard_D2s_v3"
  }
}
```

Save and close the VS CODE.

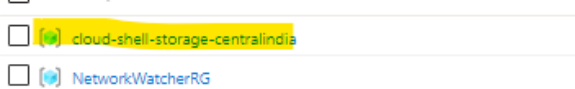
▼ Step 3 :

We need to deploy the template file using the portal.

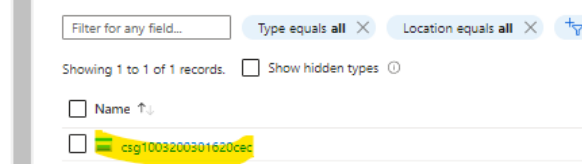
Go to the portal and check the resource group.

Go to the resource group which was created above while setting azure bash cli.

When you enter that resource group, on its overview page you will see the only one resource which will obviously be the storage account.



This is under resource group of storage class click on the highlighted one it will take you to the over view page of recourse group.



This is the resource under the resource group page. open it and it will take you to the overview page of the actual resource.

▼ Step 4:

Click on the storage account recourse and open it.

Under its overview page you will see a option file share. Click on it.

We will see that there is a single file share in the file share. Click on that as well.

Then this will show you a directory which is inside the file share.

Click on add directory and then give the directory name as template.

Resource group > Storage account resource> file share> a file share resource> directory.

File service

Large file share	Disabled
Active Directory	Not configured
Default share-level permissions	Disabled
Soft delete	Enabled (7 days)
Share capacity	5 TiB

This option you will see on the resource overview page click on it and it will open a file share service.

File share settings

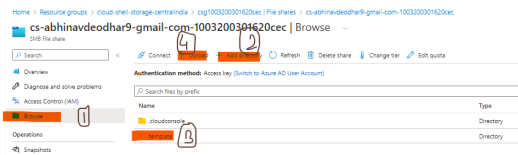
Active Directory (SMB): [Not configured](#) Default share-level permissions: [Disable](#)

Search file shares by prefix (case-sensitive)

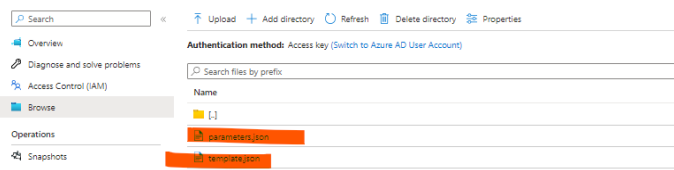
Name

cs-abhinavdeodh

This is the file share service open it and you will see a directory and there will be upload option above.



Created a template directory.



Uploaded files which were edited in VS CODE.

▼ Step 5:

Click on Browse option and we will see two files which were uploaded by us.

Go to azure cli and open bash.

Before passing any comment in azure cli go to the resource group and create one new resource group for the VM machine.

This VM machine which we are going to create is by arm template. We can also create the resource group by BACH CLI but portal is more convenient.

Now we will type the commands in bash.

```
cd clouddrive
cd template
#we are now in template folder which we have created above.
dir
#this will list the files that are there in the template folder.
az deployment group create --resource-group optimised-vm-rg --template-file template.json --parameters parameter.json
#we here said that to create a new deployment inside the resource group we just created and for that
#use the template file and parameter file
```

```
abhinav [~/clouddrive/template] $ az deployment group create --resource-group optimised-vm-rg --template-file template.json --parameters parameters.json
Running ..
```

Now we can go and check in the Resource group and there will be resources created of VM. You can go to the VM machine and under the navigation menu go to the disk and you will see standard ssd. which is what we changed.