

AZURE INTERNSHIP TRAINING :

1. Create a Resource Group (Using Portal)

- Sign in to Azure Portal.
- Select Resource groups.
- Select Create.
- Enter the following values:
 - Subscription: Select the Subscription.
 - Resource group : Enter the name to the resource group.
 - Region : Select a location , For me , it's India.
- Select Review + Create.
- Select Create and Refresh the Page.
- The Resource Group is Created.

The screenshot shows the Microsoft Azure portal interface for managing Resource Groups. The URL in the browser is `portal.azure.com/?l=en.en-in#blade/HubsExtension/BrowseResourceGroups`. The page title is "Resource groups". A search bar at the top right contains the placeholder "Search resources, services, and docs (G+ /)". Below the title, there are buttons for "Create", "Manage view", "Refresh", "Export to CSV", "Open query", "Assign tags", and "Feedback". There are also filter buttons for "Subscription == all" and "Location == all", along with an "Add filter" button. The main content area displays one record: "myResourcGroup" (Subscription: Azure for Students, Location: Central India). The page includes navigation links like "Home >" and "List view".

2. Create a vnet/subnet inside the resource group.

- Go to Azure Portal.
- Create Virtual Network.
- Fill in the Details:
 - Subscription: Select the subscription.
 - Select the resource group inside which you want to create your virtual network.

- Instance details:
 - Name: Name you want to give to the virtual network instance.
 - Region : Select The Region.
- Set the IP address you want to set up or keep it as default according to your need.
- For adding subnets , there is a default subnet already there. To add more subnets.
- Click on Add subnet. Provide the Details and that is done.
- Now you change the security settings. Set up tags and you are good to go.
- Click on Review + Create .
- The vnet and subnet is created inside the resource group.

The screenshot shows the Microsoft Azure portal interface for a resource group named 'myResourcGroup'. The top navigation bar includes the Azure logo, a search bar, and user information (ceccse.1802058@gmail.com, DEFAULT DIRECTORY). The main content area displays the 'Overview' tab for the resource group. Key details shown include:

- Subscription:** Azure for Students
- Subscription ID:** df4329de-8e60-490b-90d1-5d28df3d8d92
- Deployments:** 1 Succeeded
- Location:** Central India
- Tags:** Click here to add tags

The left sidebar lists other tabs: Activity log, Access control (IAM), Tags, Resource visualizer, Events, Deployments, Security, Policies, Properties, Locks, Cost analysis, and Cost Management.

The 'Resources' section at the bottom shows one record found, with a table listing:

Name	Type	Location
vnet1	Virtual network	Central India

Name	IPv4	IPv6	Available IPs	Delegated
default	10.0.0.0/24	-	251	-
subnet1	10.0.1.0/24	-	251	-

3. Create a Virtual Machine OS: Ubuntu Focal (using portal).Explore Availability sets/zones.

- Go to Azure Portal.
- Click on Virtual Machines down there.
- Click on Create + Virtual Machines.
- In the basics Tab , Add the subscription and correct resource group name . In the instance details , choose the correct image and add the virtual machine name. Rest you can leave default or choose as per your choice.
- User administrator account select SSH public key.
- Add the username and leave all the things to default.
- Select Review + Create .
- Select Create .
- Download the private key and create the resource.
- The VM is created . Now go to Resource and setup the connection to the virtual machine.

Microsoft Azure

Home > Virtual machines >

Create a virtual machine

Validation passed

Basics

Subscription	Azure for Students
Resource group	myResourcGroup
Virtual machine name	VMazure
Region	Central India
Availability options	Availability zone
Security type	Standard
Image	Ubuntu Server 20.04 LTS - Gen2
Size	Standard B2s (2 vcpus, 4 GiB memory)
Authentication type	SSH public key
Username	azureuser
Key pair name	VMazure_key
Public inbound ports	SSH
Azure Spot	No

Disks

OS disk type	Premium SSD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Data disks	1
Delete data disk with VM	1 disk enabled
Ephemeral OS disk	No

Networking

Virtual network	vnet1
Subnet	default (10.0.0.0/24)
Public IP	(new) VMazure-ip
Accelerated networking	Off

Actions

- Create
- < Previous
- Next >
- Download a template for automation

Microsoft Azure

Home >

CreateVm-canonical.0001-com-ubuntu-server-focal-2-20220...

Deployment

Search (Cmd +/)

Delete Cancel Redeploy Refresh

We'd love your feedback! →

Your deployment is complete

Deployment name: CreateVm-canonical.0001-com-ubuntu-server-f...
 Subscription: Azure for Students
 Resource group: myResourcGroup
 Start time: 21/2/2022, 4:16:42 pm
 Correlation ID: 4d7f56eb-692e-48c0-b9a2-ed7c9cfab2a2

Deployment details (Download)

Next steps

- Setup auto-shutdown Recommended
- Monitor VM health, performance and network dependencies Recommended
- Run a script inside the virtual machine Recommended

Go to resource Create another VM

Microsoft Defender for Cloud
 Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Work with an expert
 Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

VMazure_key.pem

Show all X

Connect to the Virtual Machine :

Run the following command :

```
ssh -i <path-to-the-private-key-file> username@public-ip-address
```

```
PP-C02FX1QZMD6M:~ janvi.intern$ cd Downloads
[PP-C02FX1QZMD6M:Downloads janvi.intern$ ssh -i VMazure_key.pem azureuser@20.204.174.2]
[32
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.11.0-1028-azure x86_64)

 * Documentation:  https://help.ubuntu.com
```

4. Setup a Flask app inside the VM which prints the hostname of the VM.

- To create a Flask application install python , pip , flask on the machine.
 - sudo apt install -y python3-pip
 - sudo apt install python3-flask
- Create a directory.
- Create a python file in the directory.
- Add the code to the file.
- Export the app as a flask app using export flask=app.py

```
● ○ ● 📁 Downloads — azureuser@VMazure: ~/flask — ssh -i VMazure_key.pem azureuser@20.204.174...
azureuser@VMazure:~$ mkdir flask
azureuser@VMazure:~$ cd flask
azureuser@VMazure:~/flask$ touch app.py
azureuser@VMazure:~/flask$ nano app.py
azureuser@VMazure:~/flask$ export flask=app.py
```

- Run the app using flask run.

```
azureuser@VMazure:~/flask$ flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
[* Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [21/Feb/2022 15:52:38] "GET / HTTP/1.1" 200 -
```

- It will provide the url. Curl that url to the another terminal and you will get the desired output.

```
[azureuser@VMazure:~$ curl http://127.0.0.1:5000/
Hostname : VMazureazureuser@VMazure:~$ ]
```

5. Create an SP with contributor access on the subscription.

- Go to the Azure Active Directory . Then Go to App Registrations.
- Go to New Registration to create a Service Principal(SP) .
- Fill the form with the required information to create a SP and Click on create.
Your SP is created.

Now give this SP the contributor access. For this follow the below steps:

- Go to the homepage of azure portal and select your subscription plan.
- Go to add role assignment and select your SP and choose role as Contributor and then click on review and assign and your contributor will have the contributor access.

The screenshot shows the Azure Active Directory section of the Azure portal. The left sidebar has 'Manage' selected, with 'App registrations' highlighted. The main area shows a table of applications. A message banner at the top right informs users that starting June 30th, 2020, no new features will be added to the Azure Active Directory Authentication Library (ADAL) and Azure AD Graph, but technical support and security updates will continue. The table lists one application:

Display name	Application (client) ID	Created on	Certificates & secrets
firstSP	8f9807cb-aba9-42eb-... 21/2/2022		-

This was with the help of azure portal. We can also make it with the help of Azure CLI.

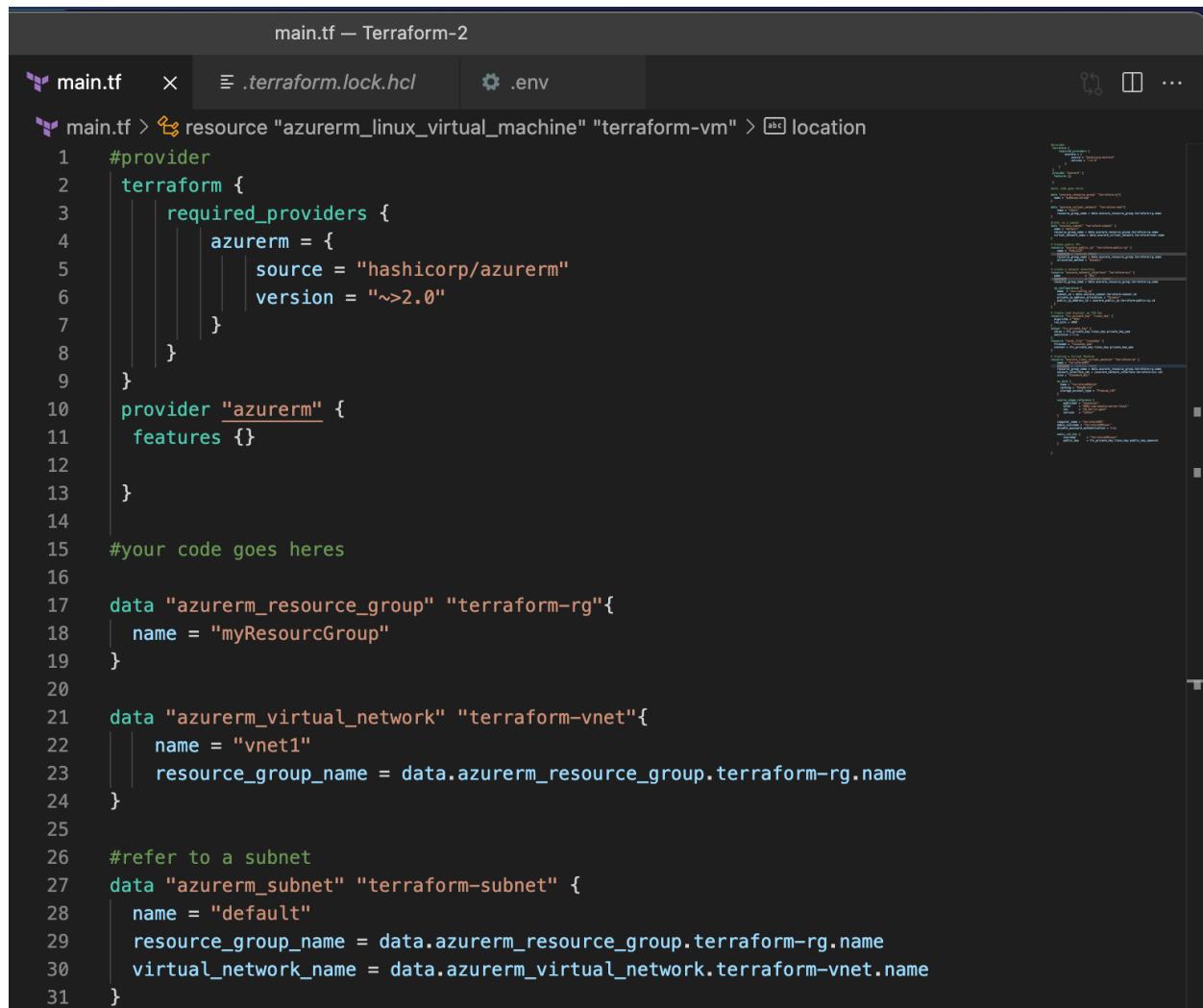
```
Bash Requesting a Cloud Shell. Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

janvi@Azure:~$ az ad sp create-for-rbac --name cliSP --role Contributor
In a future release, --scopes argument will become required for creating a role assignment. Please explicitly specify --scopes.
Creating 'Contributor' role assignment under scope '/subscriptions/df4329de-8e60-490b-90d1-5d28df3d8d92'
The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or check the credentials into your source control. For more information, see https://aka.ms/azadsp-cli
Terminal container button

{
  "appId": "cb4b33c3-9ebb-4b45-b354-54957b8e1fc8",
  "displayName": "cliSP",
  "password": "83_DMa3-a8f9htx09nL7cJ1QGly6JOIiLh",
  "tenant": "6a2cd28e-0fba-4acf-a8da-c867e0cce96"
}
janvi@Azure:~$ 
```

6. Create a VM in the same resource group, vnet, subnet using terraform using the SP in step 5 and setup the flask app in the second VM as well.

- Required:
 - Install and Open Visual Studio and install Hashicorp terraform extensions.
 - Instal Azure-CLI and terraform on the machine.
 - Link your Visual Studio account to azure account.
- Steps:
 1. Create a file name main.tf and write the terraform as below



The screenshot shows a code editor window with a dark theme. The title bar says "main.tf — Terraform-2". The editor tabs include "main.tf", ".terraform.lock.hcl", and ".env". The code itself is a Terraform configuration:

```
main.tf — Terraform-2

main.tf    X   .terraform.lock.hcl   .env   ...
```

```
main.tf > 1 resource "azurerm_linux_virtual_machine" "terraform-vm" > location
```

```
1 #provider
2 terraform {
3     required_providers {
4         azurerm = {
5             source = "hashicorp/azurerm"
6             version = "~>2.0"
7         }
8     }
9 }
10 provider "azurerm" {
11     features {}
12 }
13 }
14
15 #your code goes here
16
17 data "azurerm_resource_group" "terraform-rg"{
18     name = "myResourcGroup"
19 }
20
21 data "azurerm_virtual_network" "terraform-vnet"{
22     name = "vnet1"
23     resource_group_name = data.azurerm_resource_group.terraform-rg.name
24 }
25
26 #refer to a subnet
27 data "azurerm_subnet" "terraform-subnet" {
28     name = "default"
29     resource_group_name = data.azurerm_resource_group.terraform-rg.name
30     virtual_network_name = data.azurerm_virtual_network.terraform-vnet.name
31 }
```

main.tf — Terraform-2

main.tf .terraform.lock.hcl .env

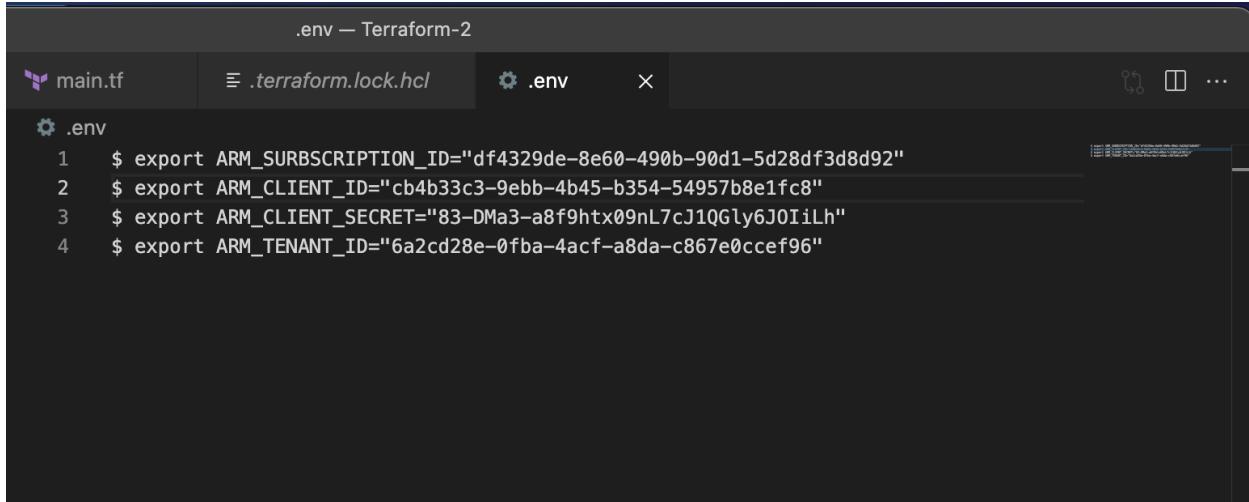
main.tf > resource "azurerm_linux_virtual_machine" "terraform-vm" > location

```
33 # Create public IPs
34 resource "azurerm_public_ip" "terraform-public-ip" {
35   name = "PublicIP"
36   location = "Central India"
37   resource_group_name = data.azurerm_resource_group.terraform-rg.name
38   allocation_method = "Dynamic"
39 }
40
41 # create a network interface
42 resource "azurerm_network_interface" "terraform-nic" {
43   name          = "Nic"
44   location      = "Central India"
45   resource_group_name = data.azurerm_resource_group.terraform-rg.name
46
47   ip_configuration {
48     name  = "nic-config_ip"
49     subnet_id = data.azurerm_subnet.terraform-subnet.id
50     private_ip_address_allocation = "Dynamic"
51     public_ip_address_id = azurerm_public_ip.terraform-public-ip.id
52   }
53 }
54
55 # Create (and display) an SSH Key
56 resource "tls_private_key" "linux_key" {
57   algorithm = "RSA"
58   rsa_bits = 4096
59 }
60 output "tls_private_key" {
61   value = tls_private_key.linux_key.private_key_pem
62   sensitive = true
63 }
64 resource "local_file" "linuxkey" {
65   filename = "linuxkey.pem"
66   content = tls_private_key.linux_key.private_key_pem
67 }
68 }
```

The screenshot shows a code editor window with a dark theme. The title bar says "main.tf — Terraform-2". The tabs include "main.tf", ".terraform.lock.hcl", and ".env". The code itself is a Terraform configuration for creating a Linux virtual machine in Azure. It defines a resource block for "azurerm_linux_virtual_machine" named "terrafrom-vm". The VM has a name of "terraformVM2", is located in "Central India", and uses a "Standard_B1s" size. It has an OS disk named "terraformOSdisk" with "ReadWrite" caching and "Premium_LRS" storage account type. The source image reference is set to Canonical's Ubuntu Server Focal version. The VM has a computer name of "terraformVM2", an admin user named "TerraformVM2user", and disabled password authentication. An admin SSH key is defined with a username of "TerraformVM2user" and a public key from a TLS private key. The entire configuration ends with a closing brace at line 100.

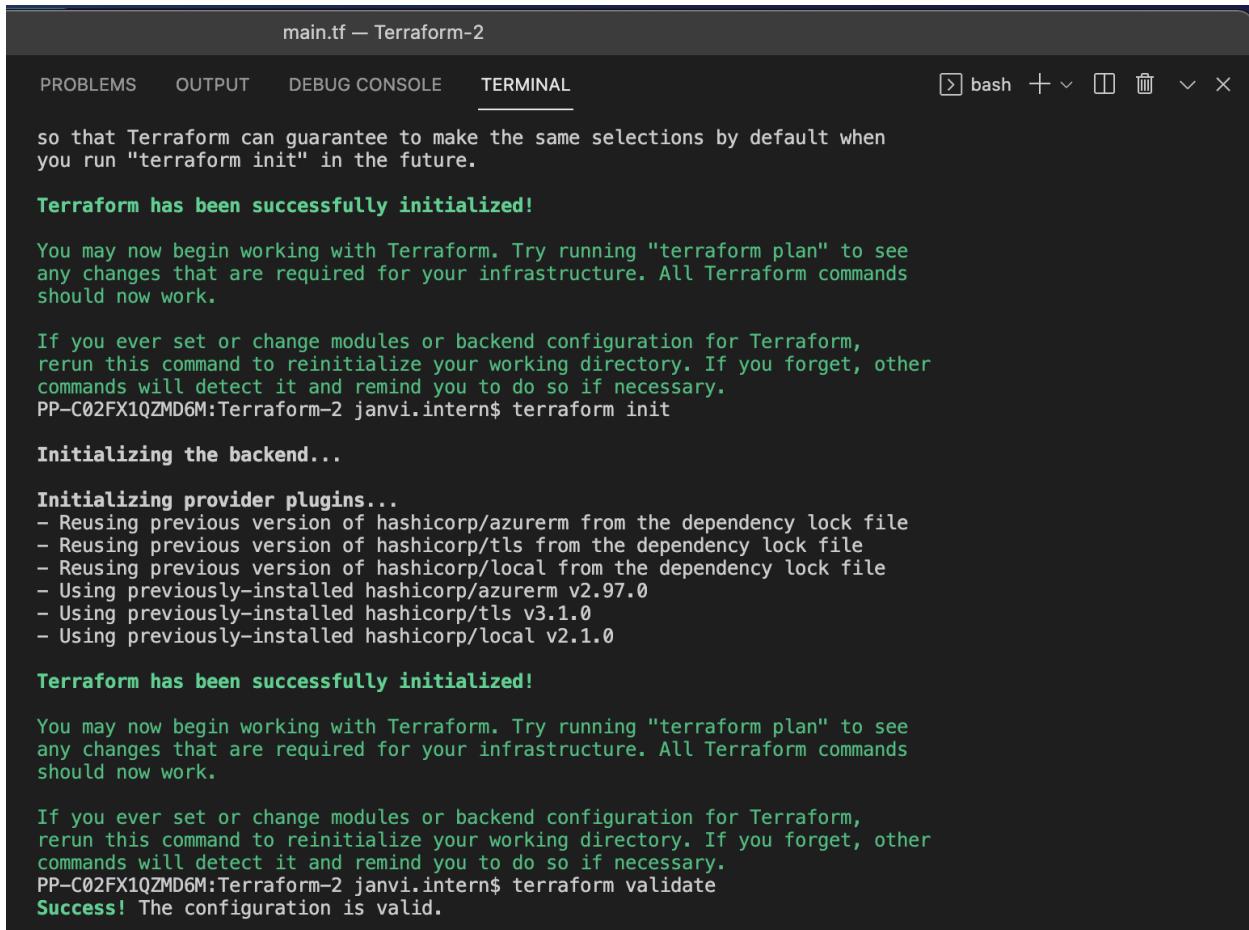
```
main.tf
main.tf > resource "azurerm_linux_virtual_machine" "terrafrom-vm" > location
69 # Creating a Virtual Machine
70 resource "azurerm_linux_virtual_machine" "terrafrom-vm" [
71   name = "terraformVM2"
72   location = "Central India"
73   resource_group_name = data.azurerm_resource_group.terraform-rg.name
74   network_interface_ids = [azurerm_network_interface.terraform-nic.id]
75   size = "Standard_B1s"
76
77   os_disk {
78     name = "terraformOSdisk"
79     caching = "ReadWrite"
80     storage_account_type = "Premium_LRS"
81   }
82
83   source_image_reference {
84     publisher = "Canonical"
85     offer     = "0001-com-ubuntu-server-focal"
86     sku       = "20_04-lts-gen2"
87     version   = "latest"
88   }
89
90   computer_name = "terraformVM2"
91   admin_username = "TerraformVM2user"
92   disable_password_authentication = true
93
94   admin_ssh_key {
95     username      = "TerraformVM2user"
96     public_key    = tls_private_key.linux_key.public_key_openssh
97   }
98
99
100 ]
```

2. Create a env file and provide the credentials related to the service principle and add them



```
.env — Terraform-2
main.tf .terraform.lock.hcl .env X
$ export ARM_SUBSCRIPTION_ID="df4329de-8e60-490b-90d1-5d28df3d8d92"
$ export ARM_CLIENT_ID="cb4b33c3-9ebb-4b45-b354-54957b8e1fc8"
$ export ARM_CLIENT_SECRET="83-DMa3-a8f9htx09nL7cJ1QGly6J0IiLh"
$ export ARM_TENANT_ID="6a2cd28e-0fba-4acf-a8da-c867e0cceef96"
```

3. Run the command terraform init to initialize the terraform.
4. Do Terraform validate to check if configuration is right and then run terraform plan.



```
main.tf — Terraform-2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash + ∨ ×
so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.
PP-C02FX1QZMD6M:Terraform-2 janvi.intern$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/azurerm from the dependency lock file
- Reusing previous version of hashicorp/tls from the dependency lock file
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/azurerm v2.97.0
- Using previously-installed hashicorp/tls v3.1.0
- Using previously-installed hashicorp/local v2.1.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.
PP-C02FX1QZMD6M:Terraform-2 janvi.intern$ terraform validate
Success! The configuration is valid.
```

5. When the VM is created , Do terraform plan and then do terraform apply . It will give the below output.

```

janvi.intern — TerraformVM2user@terraformVM2: ~/app — bash — 131x60
pp_002FX10ZMP6M: Terraform-2 janvi.intern$ terraform apply
tls_private_key_linux.key: Refreshing state... [id=a31096a868f6fac798dc1f05e5ecda82f3f891]
local_file.linuxkey: Refreshing state... [id=e1de16d84bd21c98b63f20a7d63c23e1298a699]
azurerm_public_ip.terraform-public-ip: Refreshing state... [id=/subscriptions/df4329de-8e60-490b-90d1-5d28df3d8d92/resourceGroups/m
yResourceGroup/providers/Microsoft.Network/publicIPAddresses/PublicIP]
azurerm_network_interface.terraform-nic: Refreshing state... [id=/subscriptions/df4329de-8e60-490b-90d1-5d28df3d8d92/resourceGroups/m
yResourceGroup/providers/Microsoft.Network/networkInterfaces/Nic]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply":

# azurerm_network_interface.terraform-nic has changed
~ resource "azurerm_network_interface" "terraform-nic" {
  id                               = "/subscriptions/df4329de-8e60-490b-90d1-5d28df3d8d92/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/Nic"
  name                            = "Nic"
  + tags                           = {}
  # (9 unchanged attributes hidden)
}
# (1 unchanged block hidden)

# azurerm_public_ip.terraform-public-ip has changed
~ resource "azurerm_public_ip" "terraform-public-ip" {
  id                               = "/subscriptions/df4329de-8e60-490b-90d1-5d28df3d8d92/resourceGroups/myResourceGroup/providers/Microsoft.Network/publicIPAddresses/PublicIP"
  + ip_tags                        = {}
  name                            = "PublicIP"
  + tags                           = {}
  # (9 unchanged attributes hidden)
}

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the followi
ng plan may include actions to undo or respond to these changes.

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following s
ymbols:
+ create

Terraform will perform the following actions:

# azurerm_linux_virtual_machine.terraform-vm will be created
+ resource "azurerm_linux_virtual_machine" "terraform-vm" {
  + admin_username                = "TerraformVM2user"
  + allow_extension_operations    = true
  + computer_name                 = "terraformVM2"
  + disable_password_authentication = true
  + extensions_time_budget        = "PT1H30M"
  + id                             = (known after apply)
  + location                       = "centralindia"
  + max_bid_price                 = -1
  + name                           = "terraformVM2"
  + network_interface_ids         = [
    + "/subscriptions/df4329de-8e60-490b-90d1-5d28df3d8d92/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkI
nterfaces/Nic",
    + admin_ssh_key {
      + public_key = <<-EOT
        ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACQC52oXhInK2Tb7Ev9gfiQXMuYNmwRqhgyCv6akgd9/1xVKK4fz8ZLdY0xheow3KyFcUce8B7rJ3mL14
Bh6QzoWgS2dQ5B88Ed6p7D/MP0uTVz4c491NCv0N/WIccu65fJ0uNyADCEV6GM9m17Vpr//MJHE/Q+oFQk63a1V9ESeMStpqTheAdWfdEadwXvV6NmQnQn9KCQSpGvg
0gnDyB0BUMXX9en420+cfmajdXNeHljax/Ip/af+RrykVC90PX5TuglThIEt1EM1qp4xk6a3tJMfr/8vj0vc7nf4fHGs0jZkaJP3oRsA+mV7v3S8l1ntpdk762vNrNdU
GfHubavlcDpl/LfOLNbbegomMyVbeKdrZ88cyjyAsvjV14yg3Kv1QNCK4xEmZk20M1J0P5ePjfMbY+fuW7HubM+Y483ZmrLEpfv4Tjne7K3emURF2aB161ljDM-T4z/e
mHfEcXbjV1+8+zdx0zvZds0EDY1jaxsc1J0SM+sbylyn9bJ0JGF9p3N5gJ20gBjhOfWORGgTI8s7vRWFX0zT82gEaXjP0BFyQuhbV/vd7q+aKmfV2zjWAbghHt8WdpFR
zrkLn0sKZCnbC1Y4E9Jxn10zfMyBIswKgS3jWucKbWYwmM3zo/p1aDeZsYjzvw==

      EOT
      + username = "TerraformVM2user"
    }
    + os_disk {
      + caching           = "ReadWrite"
      + disk_size_gb     = (known after apply)
      + name              = "terraformOSdisk"
      + storage_account_type = "Premium_LRS"
      + write_accelerator_enabled = false
    }
    + source_image_reference {
      + offer            = "0001-com-ubuntu-server-focal"
      + publisher        = "Canonical"
      + sku              = "20_04-lts-gen2"
      + version          = "latest"
    }
  }
}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_linux_virtual_machine.terraform-vm: Creating...
azurerm_linux_virtual_machine.terraform-vm: Still creating... [10s elapsed]
azurerm_linux_virtual_machine.terraform-vm: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.terraform-vm: Creation complete after 22s [id=/subscriptions/df4329de-8e60-490b-90d1-5d28df3d8d92/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/terraformVM2]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

```

- To create the flask app we need to follow the exact procedure we have done in task 4.

```
TerraformVM2user@terraformVM2:~/app$ flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

7. See the output of the flask app here.

```
[TerraformVM2user@terraformVM2:~$ curl http://127.0.0.1:5000/
Hostname is terraformVM2TerraformVM2user@terraformVM2:~$ ]
```

7. Add an NSG rule to allow inbound traffic only on ssh port and the port on which the flask app is running (using portal)

1. Go to the Azure Portal and select the network resource group of the vm.
2. Under the inbound security rules , click on add to add the NSG rule for SSH port 22 and fill the details as below .
3. Do the same for the flask app which was running on other and create another NSG rule to inbound the traffic on that port.
4. The NSG rule is created and the traffic is now allowed on ssh and the flask app only.

Priority	Name	Port	Protocol	Source	Destination
300	SSH	22	TCP	Any	Any
1010	Port_5000	5000	Any	Any	Any

8. Create a load balancer, to balance between the 2 VMs (flask app), so that it prints the hostnames of both the VMs in round robin.

Pre-Requisites:

1. Apache server

2. Configure apache with the flask app .

Steps to create a load balancer:

1. Go to the azure portal.
2. Search load balancer.
3. Go to the load balancer page and create a load balancer.
4. Provide the details .
5. Add the Frontend IP Configuration.
6. In the backend pool , we will need to add the Virtual machine we want to set our load balancer on.
7. Add the load balancing rule and the health probe rule so that load balancer route traffic to another vm if one of the vm goes down.
8. Once all this is setup , the load balancer IP will be able to access our VM's flask app on refresh.

You can see all the configurations in the images below.

The screenshot shows the Azure Load Balancer configuration page for a resource group named 'myResourceGroup'. The 'Essentials' section displays the following details:

Setting	Value
Resource group (move)	myResourceGroup
Location	Central India
Subscription (move)	Azure for Students
Subscription ID	df4329de-8e60-490b-90d1-5d28df3d8d92
SKU	Basic
Tags (edit)	Click here to add tags

The 'Configure high availability and scalability for your applications' section includes three main features:

- Balance IPv4 and IPv6 addresses**: Native dual-stack endpoints help meet regulatory requirements and address the fast-growing number of devices in mobile and IoT. [Learn more](#)
- Build highly reliable applications**: Load Balancer improves application uptime by routing traffic to healthy nodes. [Learn more](#)
- Secure your networks**: Control network traffic and protect private networks using built-in network address translation (NAT). [Learn more](#)

Navigation links on the left include: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Frontend IP configuration, Backend pools, Health probes, Load balancing rules, Inbound NAT rules, Properties, Locks), Monitoring (Insights, Alerts), Automation (Tasks (preview), Export template), Support + troubleshooting (New Support Request), and Home > Load Balancer.

Home > JloadBalancer

JloadBalancer | Frontend IP configuration

Load balancer

Search (Cmd+ /) < + Add Refresh Give feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings

Frontend IP configuration

Filter by name... Name ↑ IP address ↑ Rules count ↑

Name	IP address	Rules count
ILoadBalancer	20.198.116.81 (iploadbalancer)	1

X

Home > JloadBalancer

JloadBalancer | Backend pools

Load balancer

Search (Cmd+ /) < + Add Refresh Give feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings

Frontend IP configuration Backend pools Health probes Load balancing rules

Filter by name... Backend pool = all Resource Name = all Resource Status = all IP address = all Network interface = all Availability zone = all Group by Backend pool

Backend pool	Resource Name	Resource Status	IP Address	Network interface	Availability zone	Rules count
BackendLoadBalancer	terraformVM2	Running	10.0.0.5	Nic		1
BackendLoadBalancer	VMazure	Running	10.0.0.4	vmazure538		1

X

Home > JloadBalancer

JloadBalancer | Health probes

Load balancer

Search (Cmd+ /) < + Add Refresh Give feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings

Frontend IP configuration Backend pools Health probes Load balancing rules

Filter by name... Name ↑ Protocol ↑ Port ↑ Used By ↑

Name	Protocol	Port	Used By
HProbe1	TCP	80	LoadBalancerRule

X

Home > JloadBalancer

JloadBalancer | Load balancing rules

Load balancer

Search (Cmd+ /) < + Add Refresh Give feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings

Frontend IP configuration Backend pools Health probes Load balancing rules

Filter by name... Name ↑ Load balancing rule ↑ Backend pool ↑ Health probe ↑

Name	Load balancing rule	Backend pool	Health probe
LoadBalancerRule	LoadBalancerRule (TCP/80)	BackendLoadBalancer	HProbe1

X

Microsoft Azure Search resources, services, and docs (G+) cecse.1802058@gmail... DEFAULT DIRECTORY

Home > JloadBalancer >

IPLoadBalancer

Type: Public
Public IP address: iploadbalancer (20.198.116.81)
Create new

Used by:

The list of load balancing rules, inbound NAT rules, inbound NAT pools, and outbound rules using this IP address.

Name	Type
LoadBalancerRule	Load balancing rule

Microsoft Azure Search resources, services, and docs (G+) cecse.1802058@gmail... DEFAULT DIRECTORY

Home > JloadBalancer >

LoadBalancerRule

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances. Only backend instances that the health probe considers healthy receive new traffic.

Name: LoadBalancerRule
IP Version: IPv4
Frontend IP address: IPLoadBalancer (20.198.116.81)
Backend pool: BackendLoadBalancer
Protocol: TCP
Port: 80
Backend port: 80
Health probe: HProbe1 (TCP:80)
Session persistence: None
Idle timeout (minutes): 4
Floating IP: Disabled

Microsoft Azure Search resources, services, and docs (G+ /) 9 Help Settings User cecse.1802058@gmail... DEFAULT DIRECTORY

Home > JloadBalancer >

HProbe1 ...

JloadBalancer

Name: HProbe1

Protocol: TCP

Port: 80

Interval: 5 seconds

Unhealthy threshold: 3 consecutive failures

Used by: LoadBalancerRule

Microsoft Azure Search resources, services, and docs (G+)

Home > JloadBalancer >

BackendLoadBalancer

JloadBalancer

Name	BackendLoadBalancer
Virtual network	vnet1 (myResourcGroup)
Associated to	Virtual machines
IP Version	<input checked="" type="radio"/> IPv4 <input type="radio"/> IPv6

Virtual machines

You can only attach virtual machines in centralindia that have a basic SKU public IP configuration or no public IP configuration. All virtual machines must be in the same availability set and all IP configurations must be on the same virtual network.

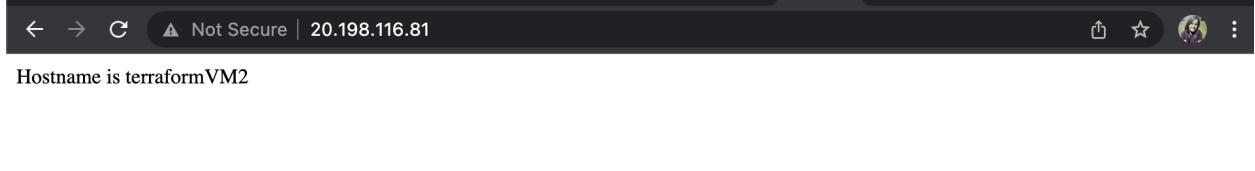
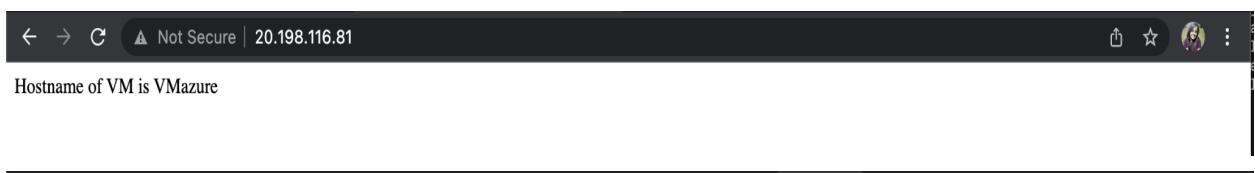
+ Add X Remove

Virtual machine	IP Configuration	Availability set
terraformVM2	nic-config_ip (10.0.0.5)	AVAILABILITYSET
VMazure	ipconfig1 (10.0.0.4)	AVAILABILITYSET

Used by

The list of load balancing rules, inbound NAT rules, and outbound rules using this backend pool.

Name	Type
LoadBalancerRule	Load balancing rule



9. Shut down one of the machines and check how the load balancer works.

When we shutdown one VM , the load balancer will send requests to the other VM and vice versa . Health Prob will check the health of the machine as per the time limit set by you and will forward the traffic accordingly on the basis of health of the machine.

The screenshot shows the Azure portal interface for managing a virtual machine. The main title bar includes the URL `portal.azure.com/?l=en-en#@ceccse1802058@gmail.onmicrosoft.com/resource/subscriptions/df4329de-8e60-490b-90d1-5d28...`. The top navigation bar has several items like SALTS, Untitled, SRE Int, How to, data sl, Netpla, AS HW, AZURE, terra, 20.198, and a user icon.

The main content area shows the details for a virtual machine named **terraformVM2**. The left sidebar has sections like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Networking, Connect, Disks, Size, Security, Advisor recommendations, Extensions + applications, Continuous delivery, Availability + scaling, Configuration, Identity, Properties, Locks), and a search bar.

Essentials section details:

- Resource group ([move](#)) **myResourceGroup**
- Status: Stopped (deallocated)
- Location: Central India
- Subscription ([move](#)) **Azure for Students**
- Subscription ID: df4329de-8e60-490b-90d1-5d28df3d8d92
- Tags ([edit](#)) [Click here to add tags](#)

Properties tab details:

Virtual machine	Networking
Computer name	Public IP address 20.198.116.81
Health state	Public IP address (IPv6) -
Operating system	Private IP address 10.0.0.5
Publisher	Private IP address (IPv6) -
Offer	Virtual network/subnet vnet1/default
Plan	DNS name Configure
VM generation	Host group None
Host	Size Standard D1v2

The status bar at the bottom shows the URL `Not Secure | 20.198.116.81`.

Hostname of VM is VMazure

10. Add a 10G data disk to the machine, and create a mount point to redirect the flask app logs to the mount point. Explore disk redundancy options.

For Both the Machines , Repeat the same steps:
I am gonna show this for the VMazure Virtual Machine:

1. Go to the Azure Portal .
2. Open the virtual Machine Page.
3. On the left menu , there is an option named disks, Select that.
4. Click on the create and attach a new data disk.
5. Select a 10 GB data size disk and provide the other details , and click save.

The screenshot shows the 'Disks' section of the VM Azure interface. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Networking, Connect, Disks selected), Size, Security, Advisor recommendations, Extensions + applications, and Continuous delivery. The main area has tabs for 'OS disk' and 'Data disks'. Under 'OS disk', it shows 'Swap OS disk' and a table for 'VMazure_OsDisk_1_74676588a4084063a76b7da1'. Under 'Data disks', it shows 'Create and attach a new disk' and 'Attach existing disks', with a table for 'VMazureAddedDataDisk'.

Now , we have successfully created the disk.

We need to create a mount point to redirect the flask app logs to the mount point so we are redirecting the data to this disk.

1. Connect to the VM and check all the disks present in the system using sudo fdisk -l
2. Format the disk by using sudo mkfs.ext4 /dev/sdc
3. Mount the flask app on the disk using sudo mount /dev/sdc /var/www/html/flask
4. Check if the flask app is mounted on that location using sudo lsblk.
5. In order to make these changes permanent , we will need to make the entry in the stab and for that we use sudo nano /etc/fstab and enter the following mount with the uuid of the location or disk where it needs to be mounted.
6. Now your mount is permanent , and you are good to go.

```

azureuser@VMazure:~$ sudo mkfs.ext4 /dev/sdc
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: eb62bd6a-389c-48fe-a397-9241d723de57
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
[...]
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

azureuser@VMazure:~$ sudo mount /dev/sdc /var/www/html/flask
azureuser@VMazure:~$ sudo lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0    7:0    0 67.2M  1 loop /snap/lxd/21835
loop1    7:1    0 61.9M  1 loop /snap/core20/1328
loop3    7:3    0 61.9M  1 loop /snap/core20/1361
loop4    7:4    0 43.6M  1 loop /snap/snappy/14978
loop5    7:5    0 67.9M  1 loop /snap/lxd/22526
sda     8:0    0   30G  0 disk
└─sda1   8:1    0 29.9G 0 part /
sda14   8:14   0   4M  0 part
└─sda15  8:15   0 100M 0 part /boot/efi
sdb     8:16   0   4G  0 disk
└─sdb1   8:17   0   4G  0 part /mnt
sdc     8:32   0   10G 0 disk /var/www/html/flask
azureuser@VMazure:~$ sudo blkid
/dev/sdb1: UUID="dc0417b-81d7-4aa1-8e85-e76bb1cdca1" TYPE="ext4" PARTUUID="8a7cb40d-01"
/dev/sda1: LABEL="cloudimg-rootfs" UUID="00b72c14-e32b-4e33-988c-002ba91aafec" TYPE="ext4" PARTUUID="8506d029-5fc8-4161-b338-d81d08859573"
/dev/sda15: LABEL_FATBOOT="UEFI" LABEL="UEFI" UUID="BA5D-627F" TYPE="vfat" PARTUUID="028e78d1-789e-4387-ad00-0d98b9d78558"
/dev/loop0: TYPE="squashfs"
/dev/loop1: TYPE="squashfs"
/dev/loop3: TYPE="squashfs"
/dev/loop4: TYPE="squashfs"
/dev/loop5: TYPE="squashfs"
/dev/sda14: PARTUUID="55eeccb43-34d8-47b2-8d2c-892203335842"
/dev/sdc: UUID="eb62bd6a-389c-48fe-a397-9241d723de57" TYPE="ext4"
azureuser@VMazure:~$ sudo nano /etc/fstab

```

11. Resize the VMs (via portal and terraform resp.)

Via Portal:

1. Go to the Azure portal .
2. Open the virtual machine page.
3. In the left menu , select size.
4. Pick a new size and select Resize.
5. Your VM will be resized as per your choice.

The screenshot shows the Azure portal interface for selecting a VM size. On the left, a sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Networking, Connect, Disks, and Size. The 'Size' option is currently selected and highlighted in grey. The main content area is titled 'VMazure | Size' and displays a table of VM sizes. The table includes columns for VM Size, Family, vCPUs, RAM (GiB), and Data disks. The first few rows shown are DS1_v2, D2s_v3, B2s, B1s, and B2ms. A tooltip at the top right states: 'If the virtual machine is currently running, changing its size will cause it to be restarted. Stopping the virtual machine may reveal additional sizes.' At the bottom of the table, there are sections for 'D-Series v4' and 'B-Series'. A note at the bottom right says: 'Prices presented are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Final charges will appear in your local currency in cost analysis and billing views. View Azure pricing calculator.'

12. Destroy all the resources. Delete the SP and extra users added if any.

To Destroy the Resources from terraform:

Do: `terraform destroy`

To destroy all the resources from the Portal.

Delete all the resources present there on the Portal.