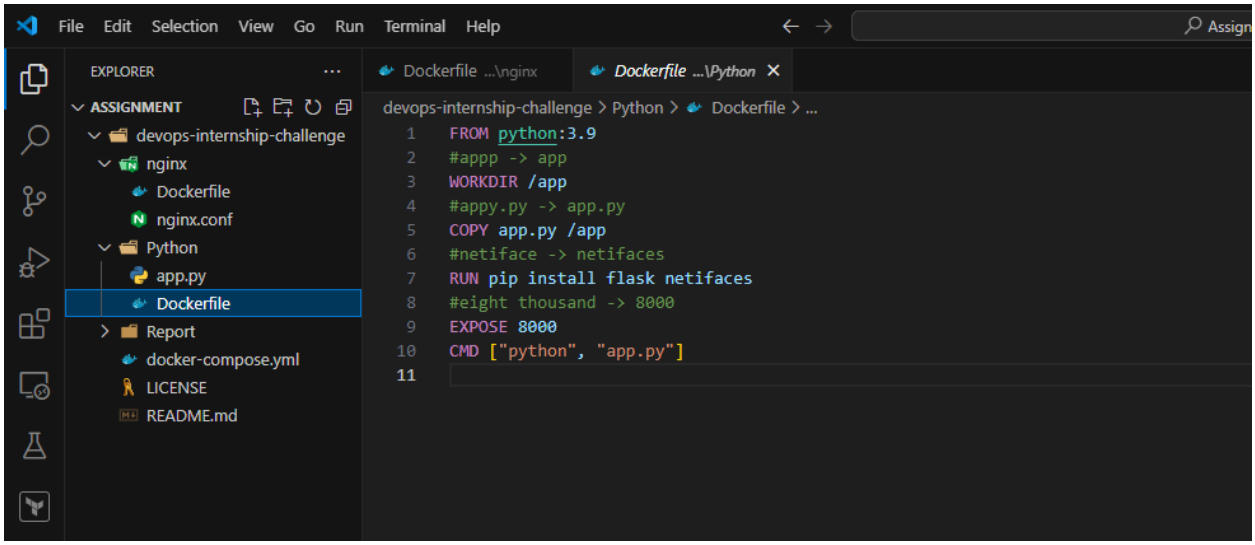
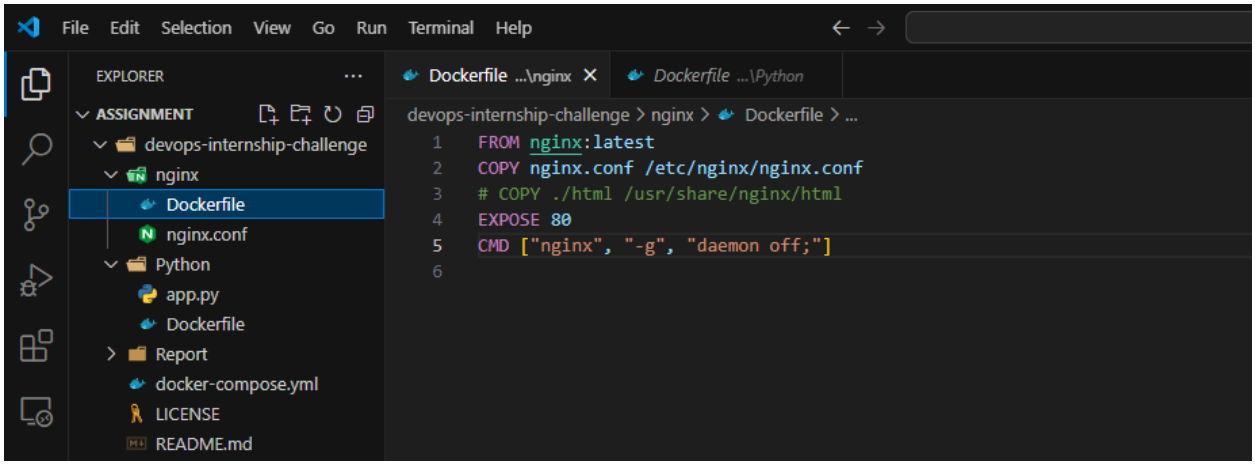
Assignment:

devops-qoala-assignment-Aditya-Singh-21ucc011

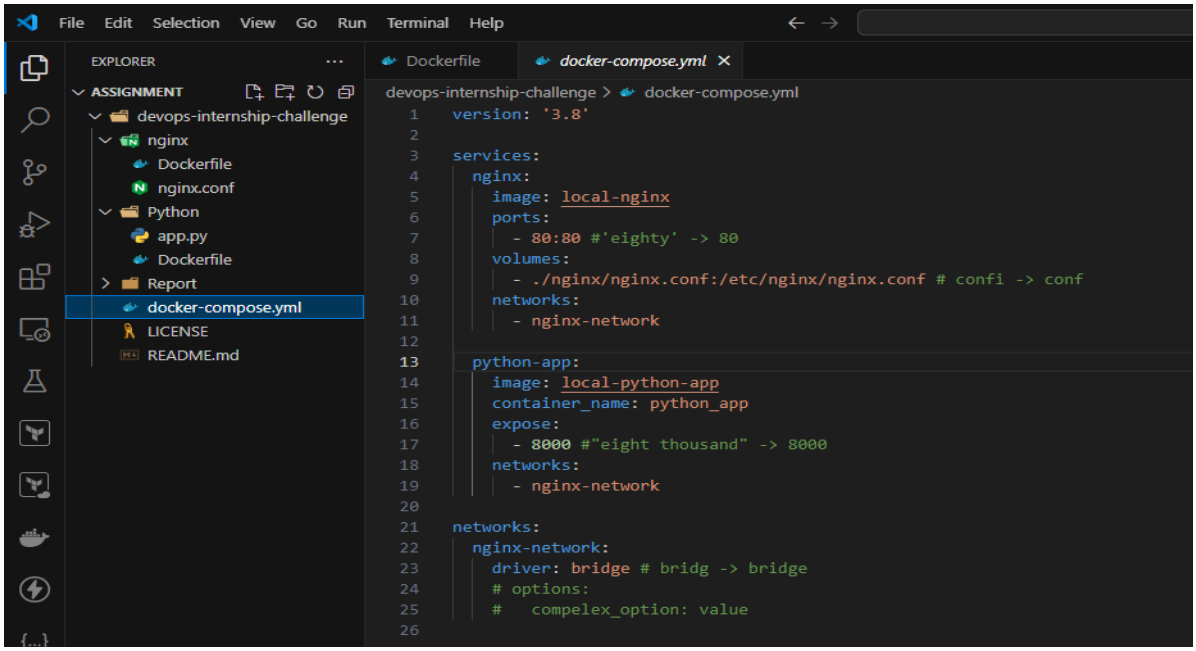
Approach to the Problem:

1. First we debug the dockerfile of nginx, Python and docker.compose.yml file

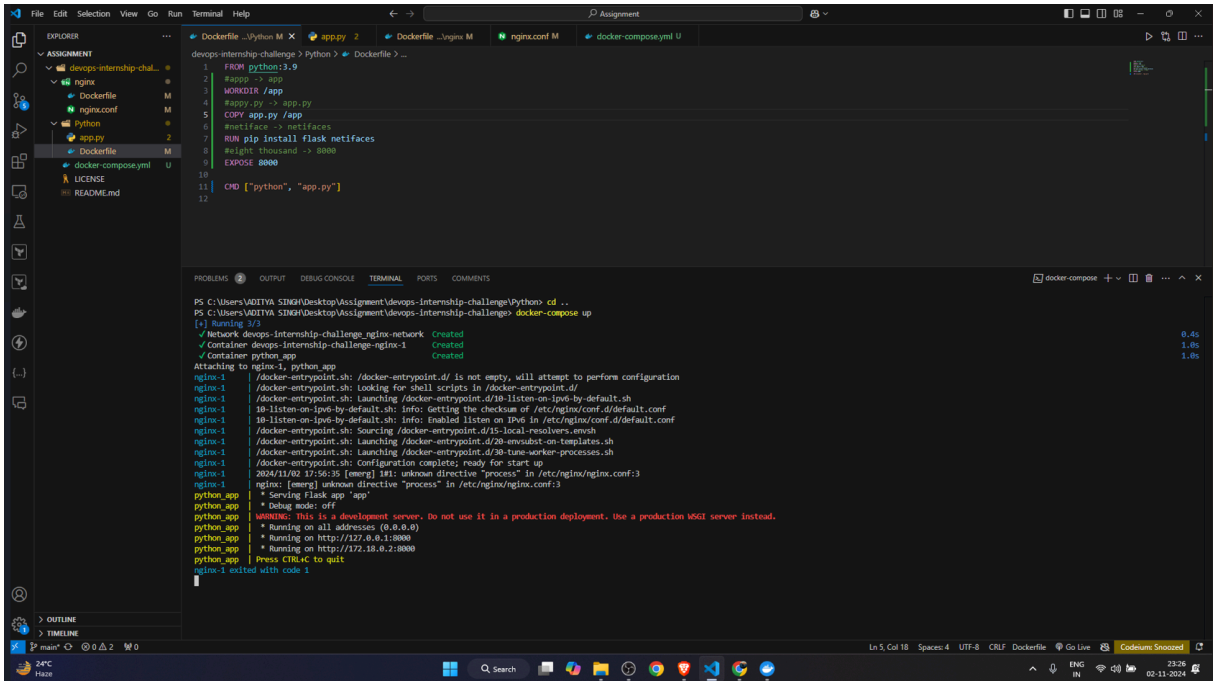




2. We then debug the compose file and then changed the extension from .yaml to .yml

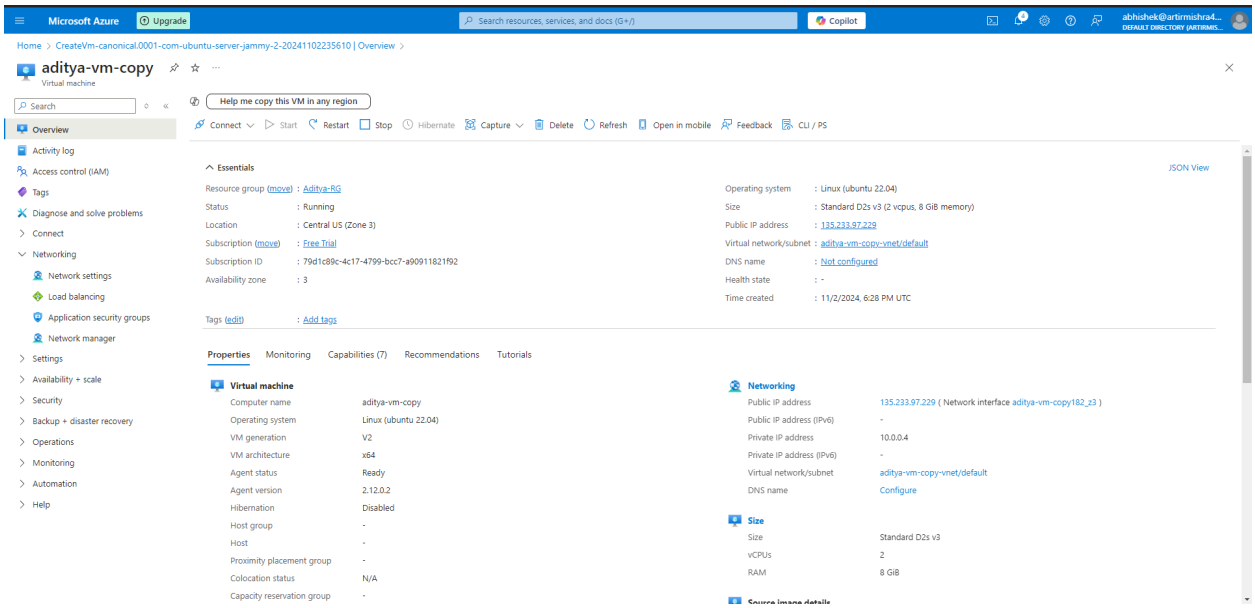


3. We first build the docker images of the nginx and the run the command docker-compose up to run the application locally on port: 80

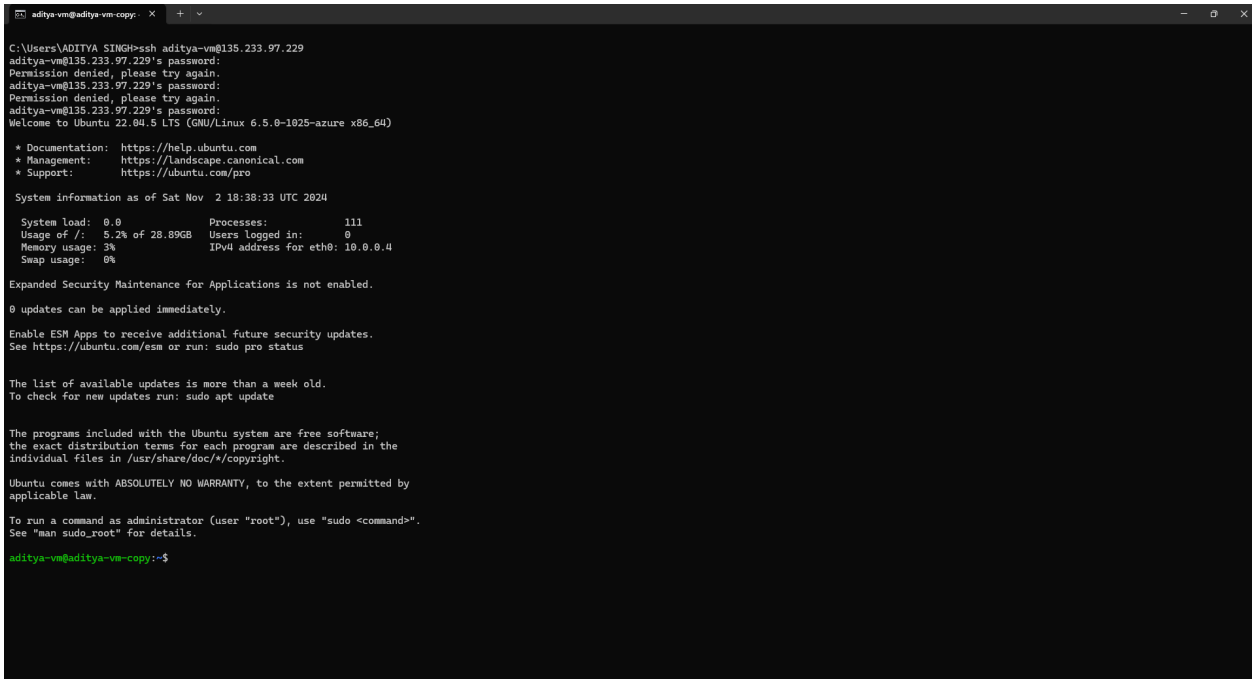


4. Attaching the Screenshot of the locally running application

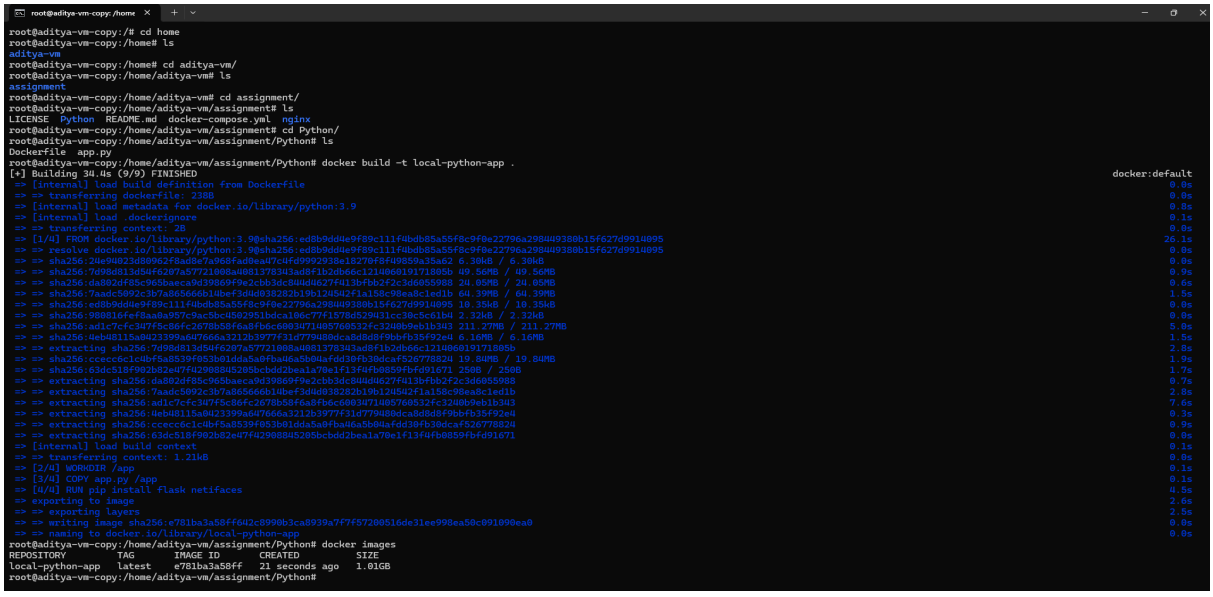
5. Now for the Bonus Points:
   Let's deploy it on Azure Cloud after creating a Virtual Machine



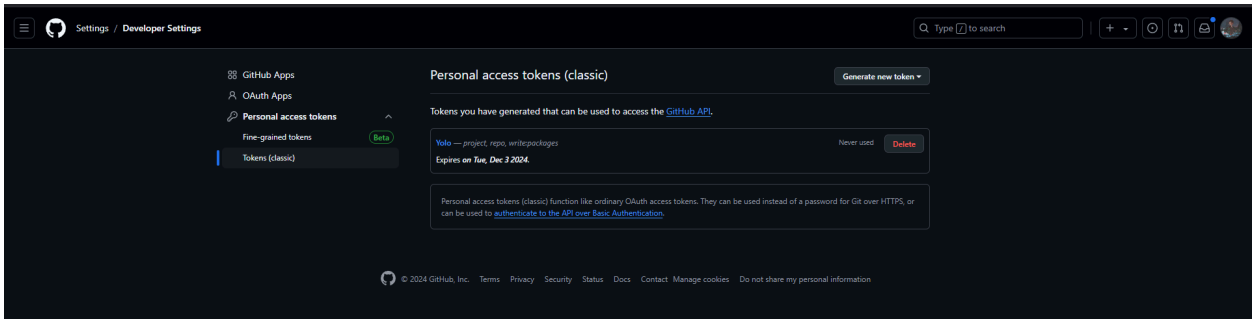
6. Now we will access the SSH on cmd by the following command
   ssh [user_name]@[Public-IP]:



7. Now we follow the same set of instructions on Virtual Machine
   a. Install Docker
   b. Install Docker Compose



8. Now we git clone the repo and the problem I faced was that I was not able to access the Private repo, So for that I created a token from GitHub to access the Private Repositories.



9. Docker Images created of both the nginx and Python app

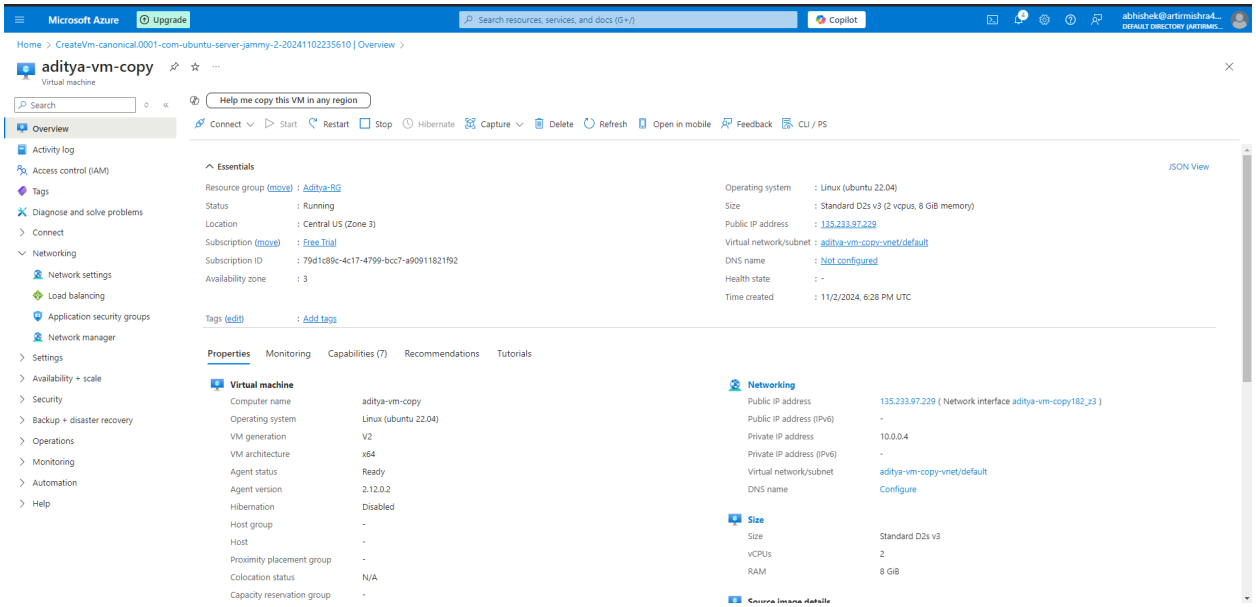
10. We deployed it on Virtual Machine and generated the Public IP and Hit it directly on the server but we did not create the Virtual Machine directly from the portal, we created it using Terraform(IaC) tool.

Steps:

1. Read the documentation of Azure Virtual Machine Creation
2. Used Terraform for the same purpose.
3. terraform init -> terraform plan -> terraform apply

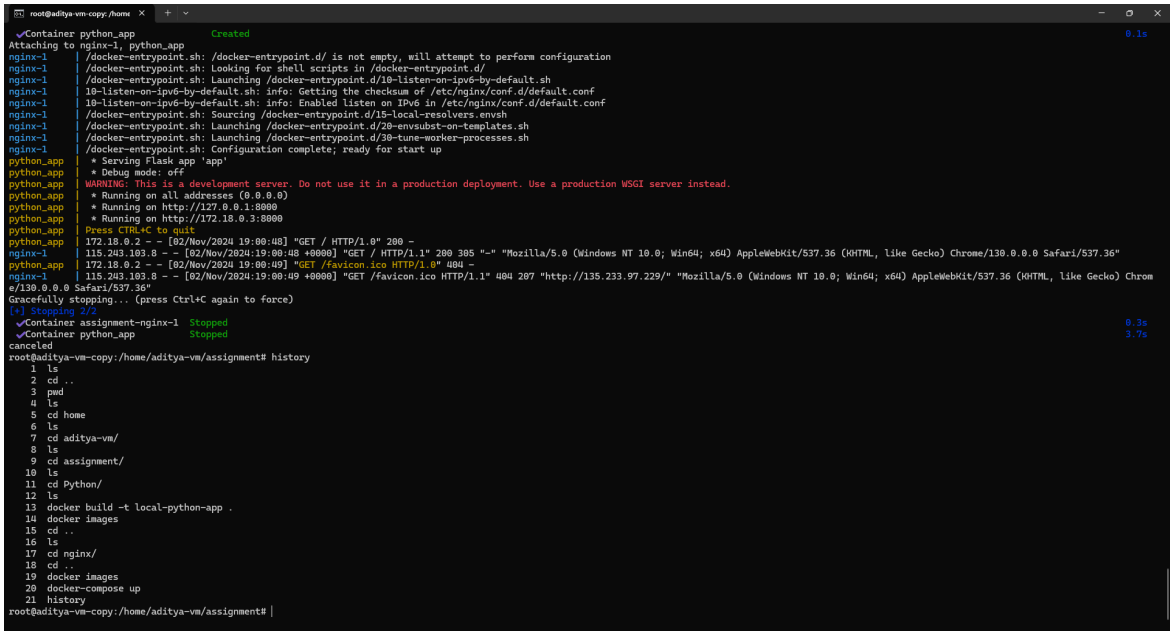
```
resource "azurerm_virtual_machine" "main" {
  name                  = "${var.prefix}-vm"
  location              = azurerm_resource_group.example.location
  resource_group_name   = azurerm_resource_group.example.name
  network_interface_ids = [azurerm_network_interface.main.id]
  vm_size               = "Standard_DS1_v2"

  # Uncomment this line to delete the OS disk automatically when deleting the VM
  # delete_os_disk_on_termination = true

  # Uncomment this line to delete the data disks automatically when deleting the VM
  # delete_data_disks_on_termination = true

  storage_image_reference {
    publisher = "Canonical"
    offer     = "0001-com-ubuntu-server-jammy"
    sku       = "22_04-lts"
    version   = "latest"
  }
  storage_os_disk {
    name              = "myosdisk1"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }
  os_profile {
    computer_name  = "hostname"
    admin_username = "testadmin"
    admin_password = "Password1234!"
  }
  os_profile_linux_config {
    disable_password_authentication = false
  }
  tags = {
    environment = "staging"
  }
}
```

Virtual Machine:



11. Final Result:

12. Attaching the screenshot of the commands used for the process:



——————————Thank You——————————————————————————