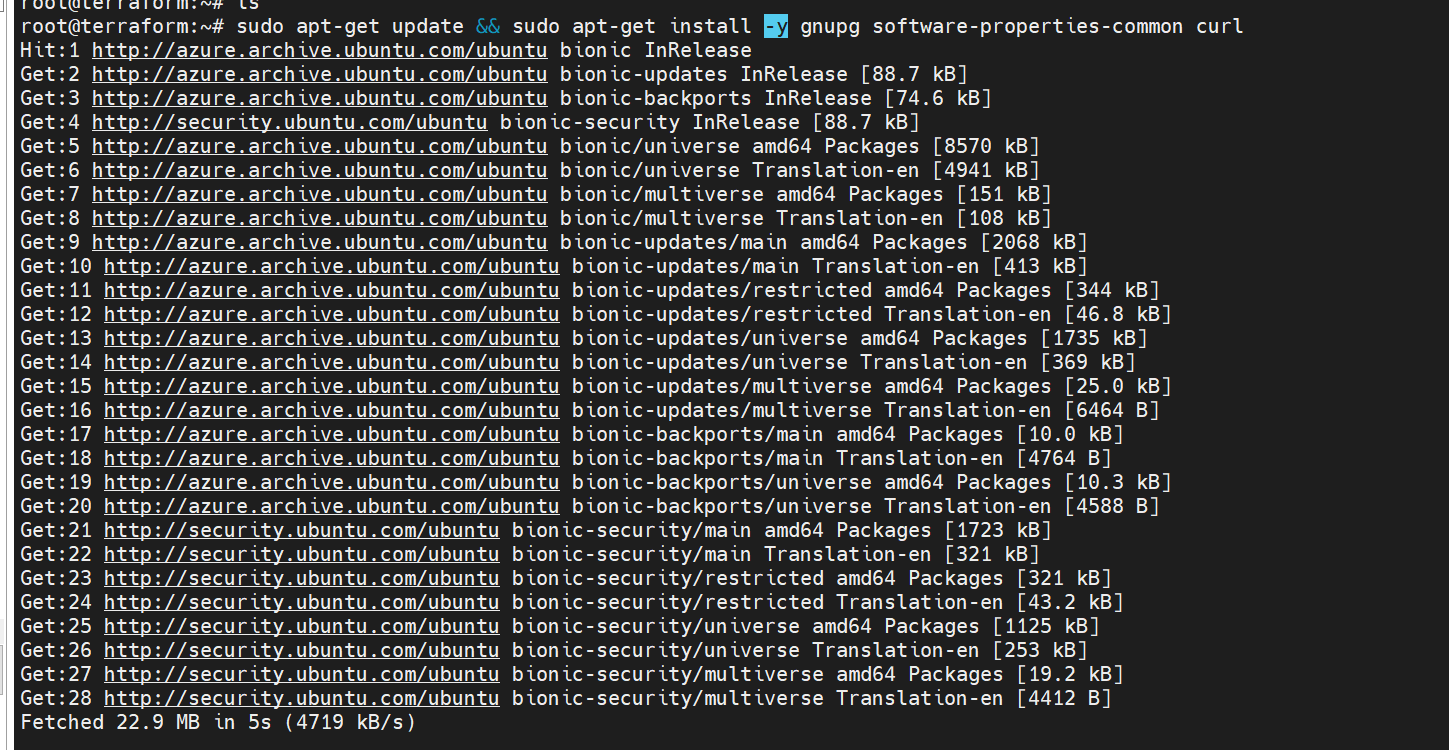
TERRAFORM INSTALLATIONS

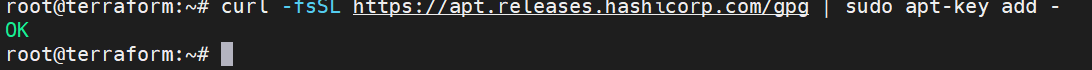
First of all download the terraform on your virtual machine.

sudo apt-get update && sudo apt-get install -y gnupg software-properties-common curl

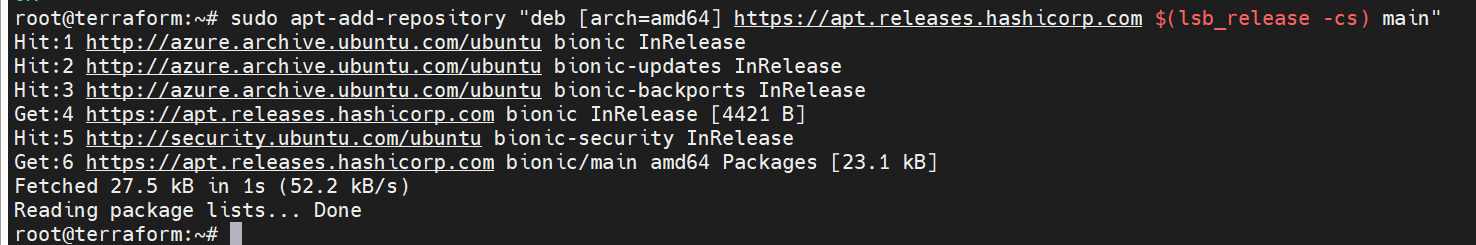


Add the HashiCorp [GPG key](https://apt.releases.hashicorp.com/gpg).

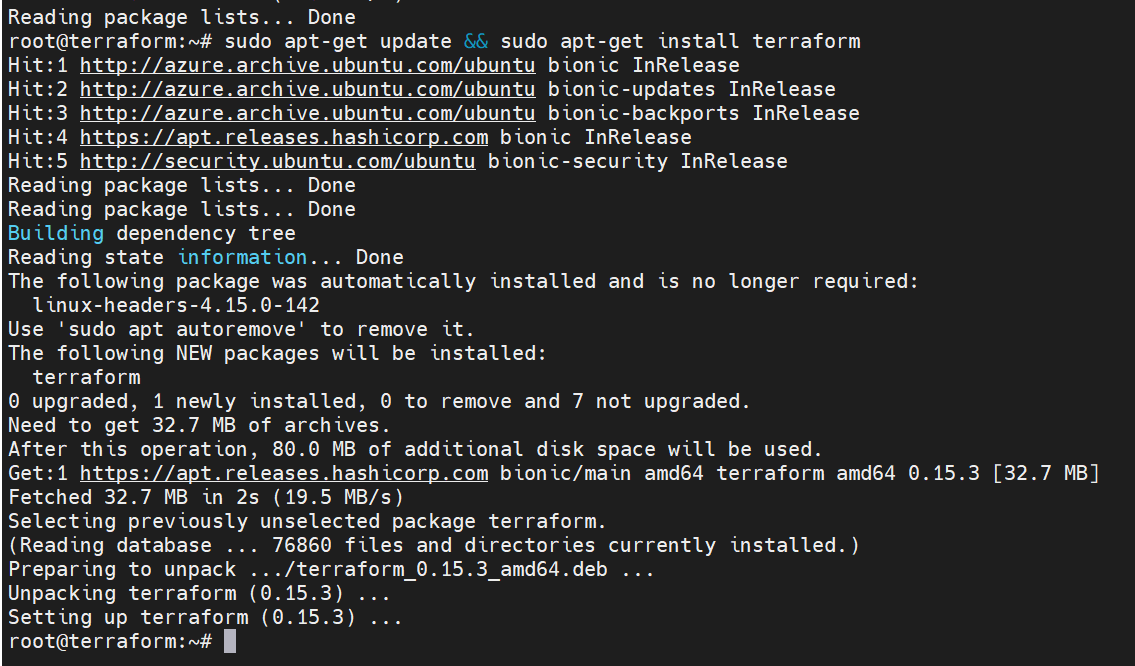
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -



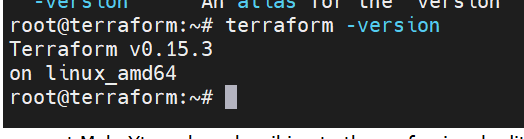
sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb\_release -cs) main"



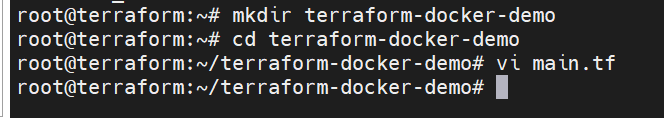
sudo apt-get update && sudo apt-get install terraform

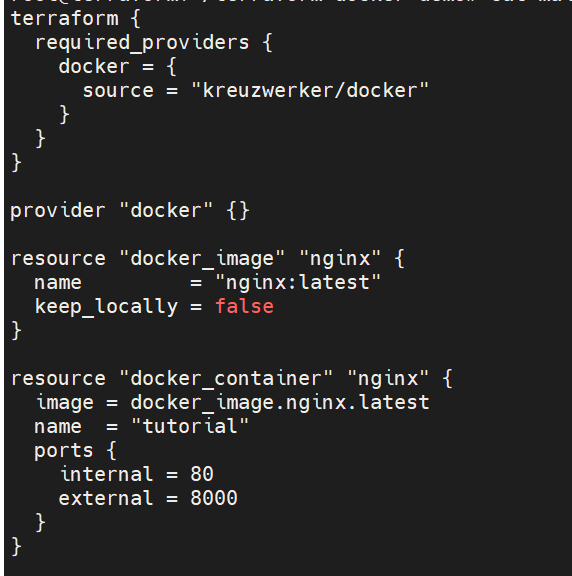


Whether is it installed or not to check terraform -version



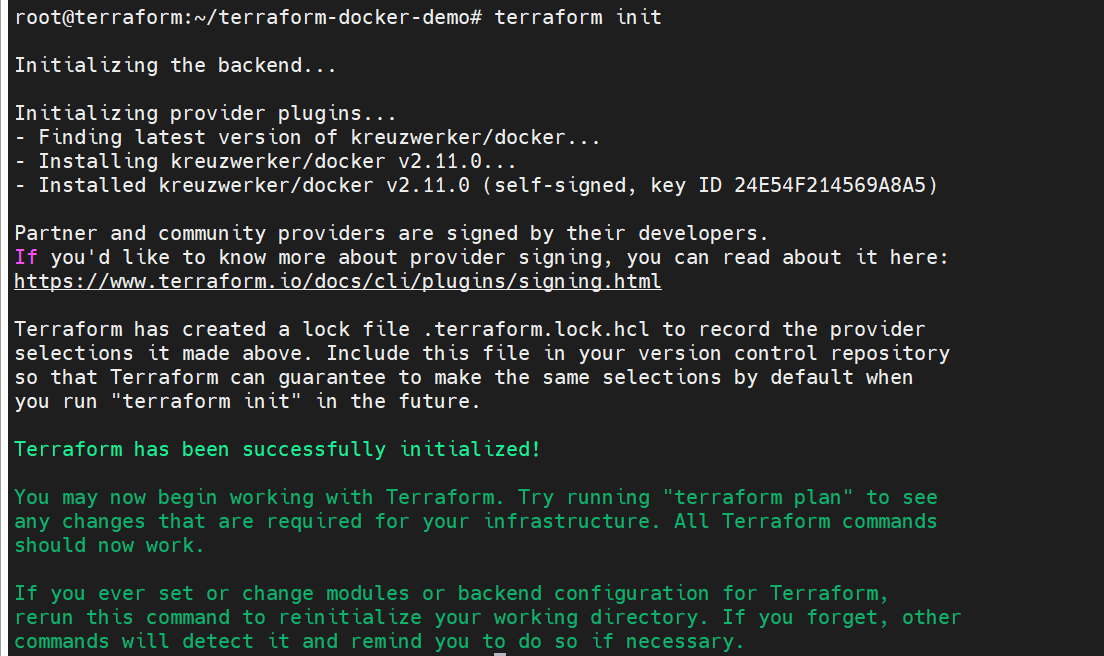
Create one directory inside your root directory as a name terraform-docker-demo and create a file main.tf and inside the main.tf I wrote like download ngnix from docker hub.





After saving the main.tf file then when we are using first time first of all initialize the terraform

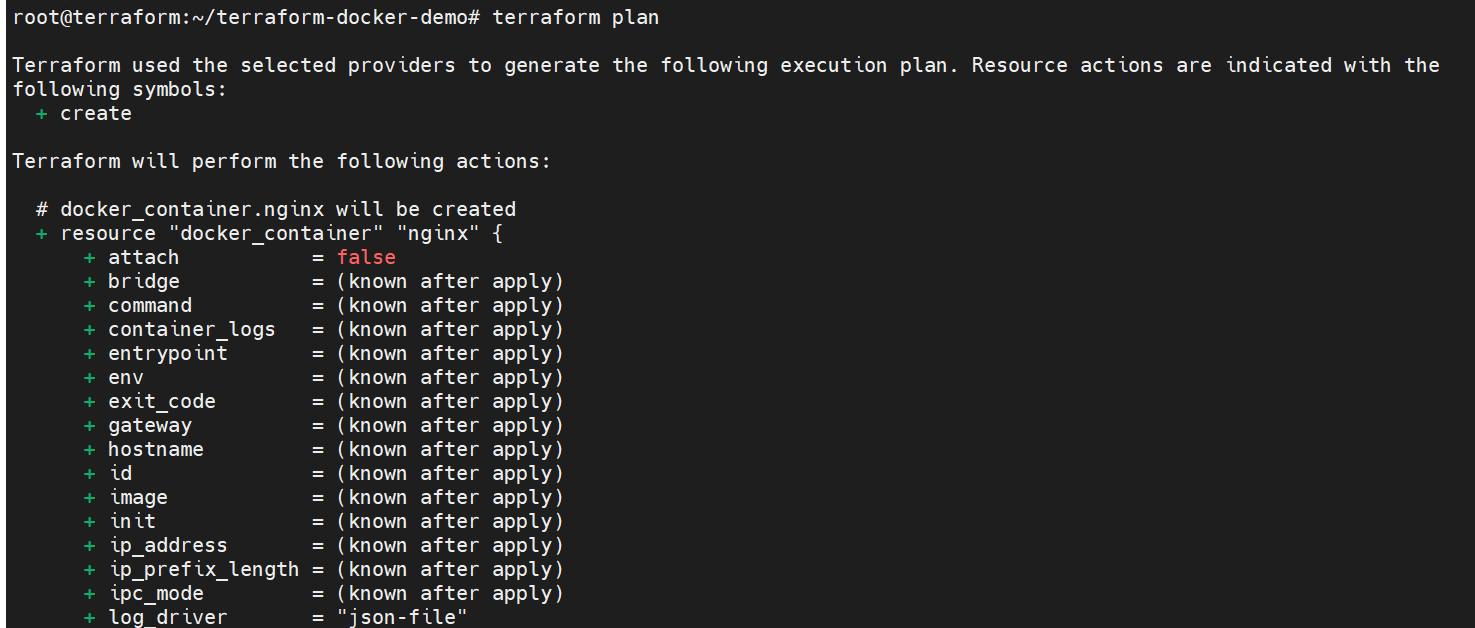
terraform init



After we can init and we can check validate with help of terraform validate

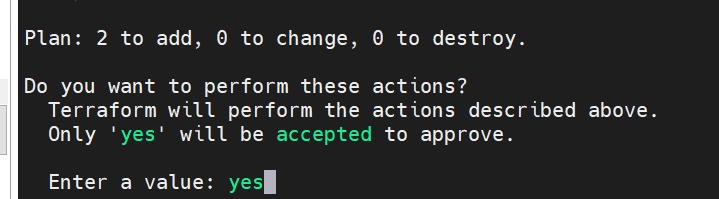


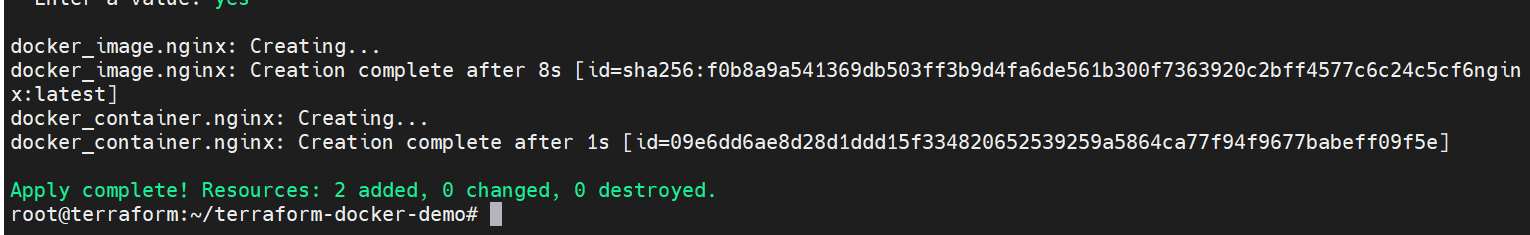
Then after successfully we can check plan like dry run terraform plan



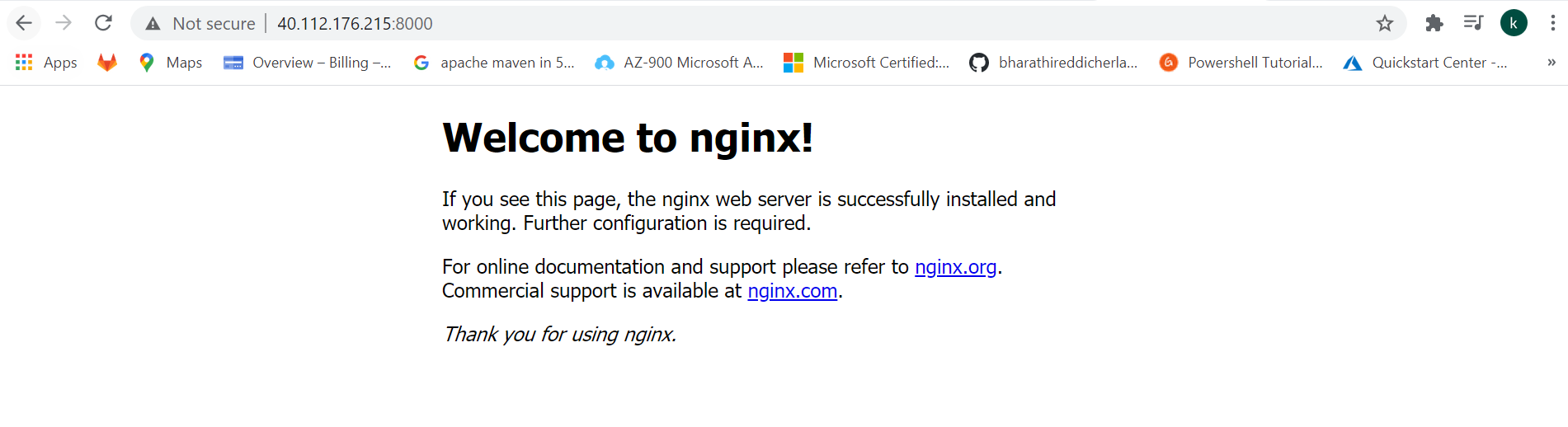
Then finally we can fire terraform apply then it will give the output and apply all the updates.

Terraform apply





Then we can access the ngnix in the browser by the help of 40.1121.76.215:8000



Then we need to install azure-cli by using below caommand

Azure-cli install

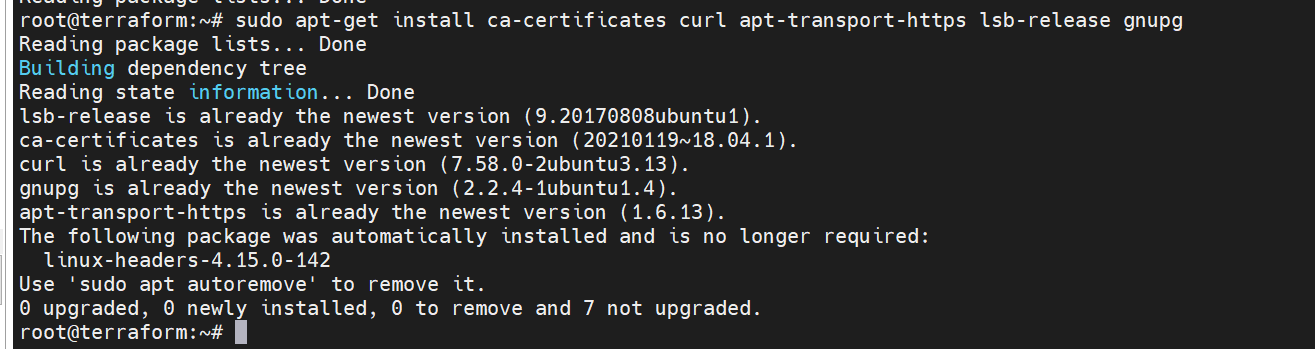
sudo apt-get update

sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg

curl -sL https://packages.microsoft.com/keys/microsoft.asc |

gpg --dearmor |

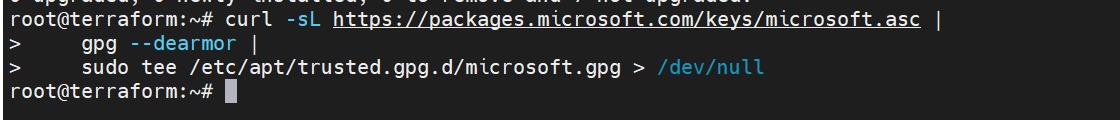
sudo tee /etc/apt/trusted.gpg.d/microsoft.gpg > /dev/null



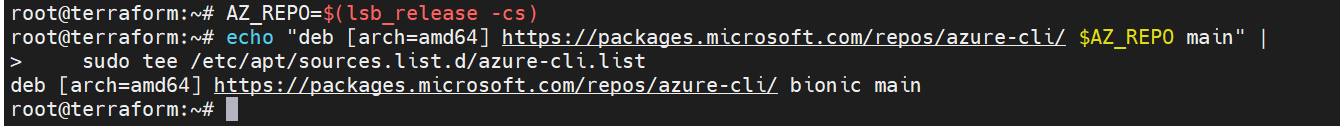
AZ\_REPO=$(lsb\_release -cs)

echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ\_REPO main" |

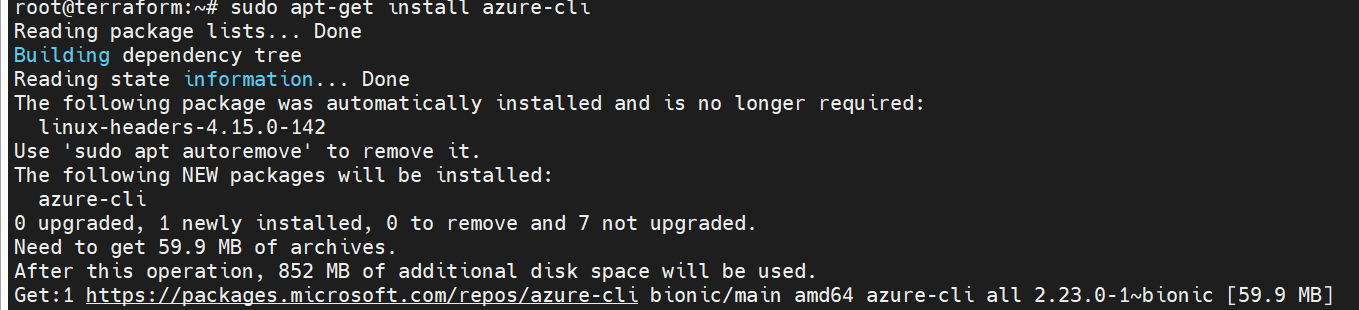
sudo tee /etc/apt/sources.list.d/azure-cli.list

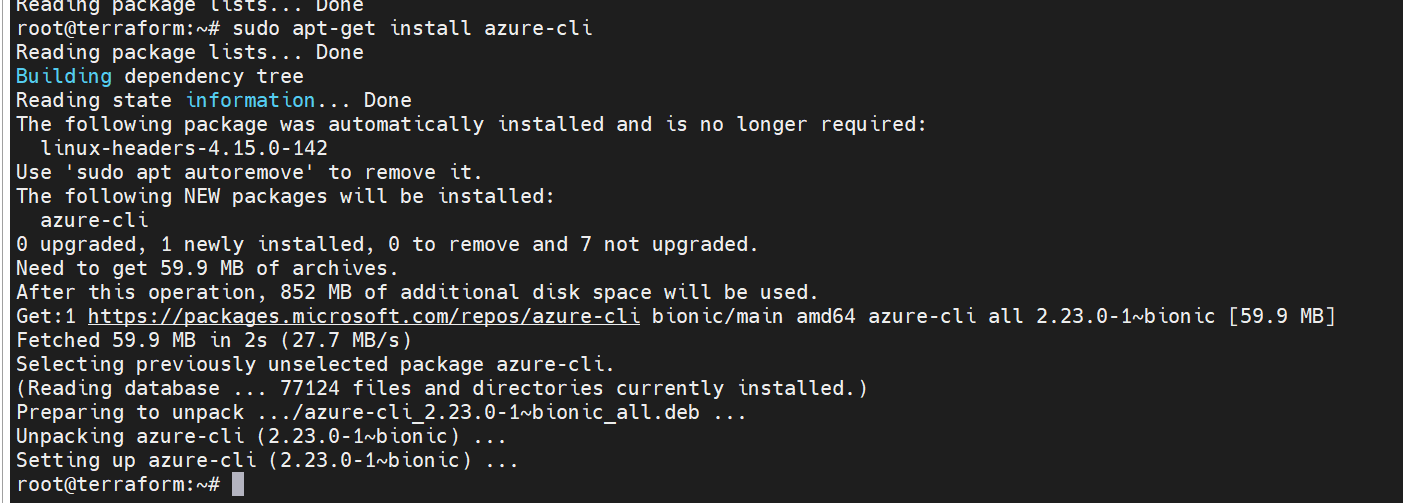


sudo apt-get update



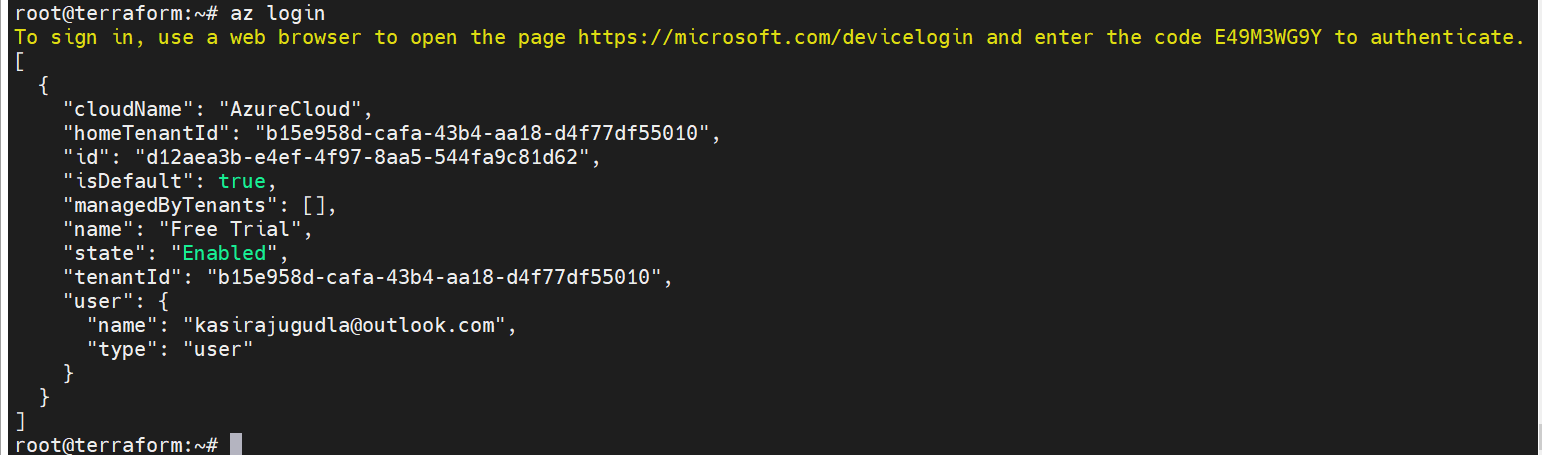
sudo apt-get install azure-cli



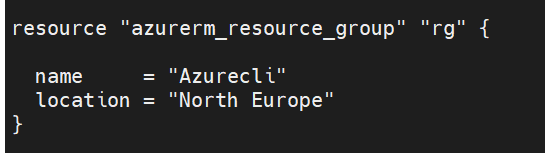


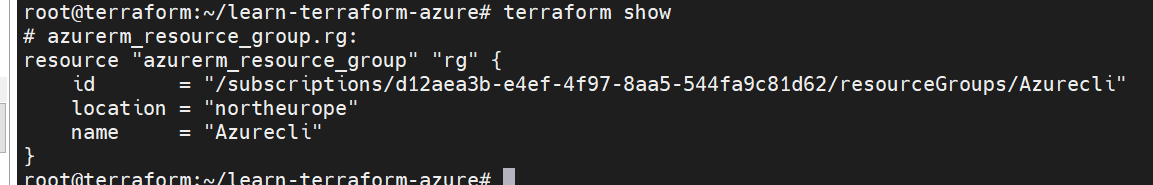
az login when you fire this one it it will show like this

browser window will open and you will be prompted to enter your Azure login credentials. After successful authentication, your terminal will display your subscription information



New resource group in azure: we can add the resource group name and location

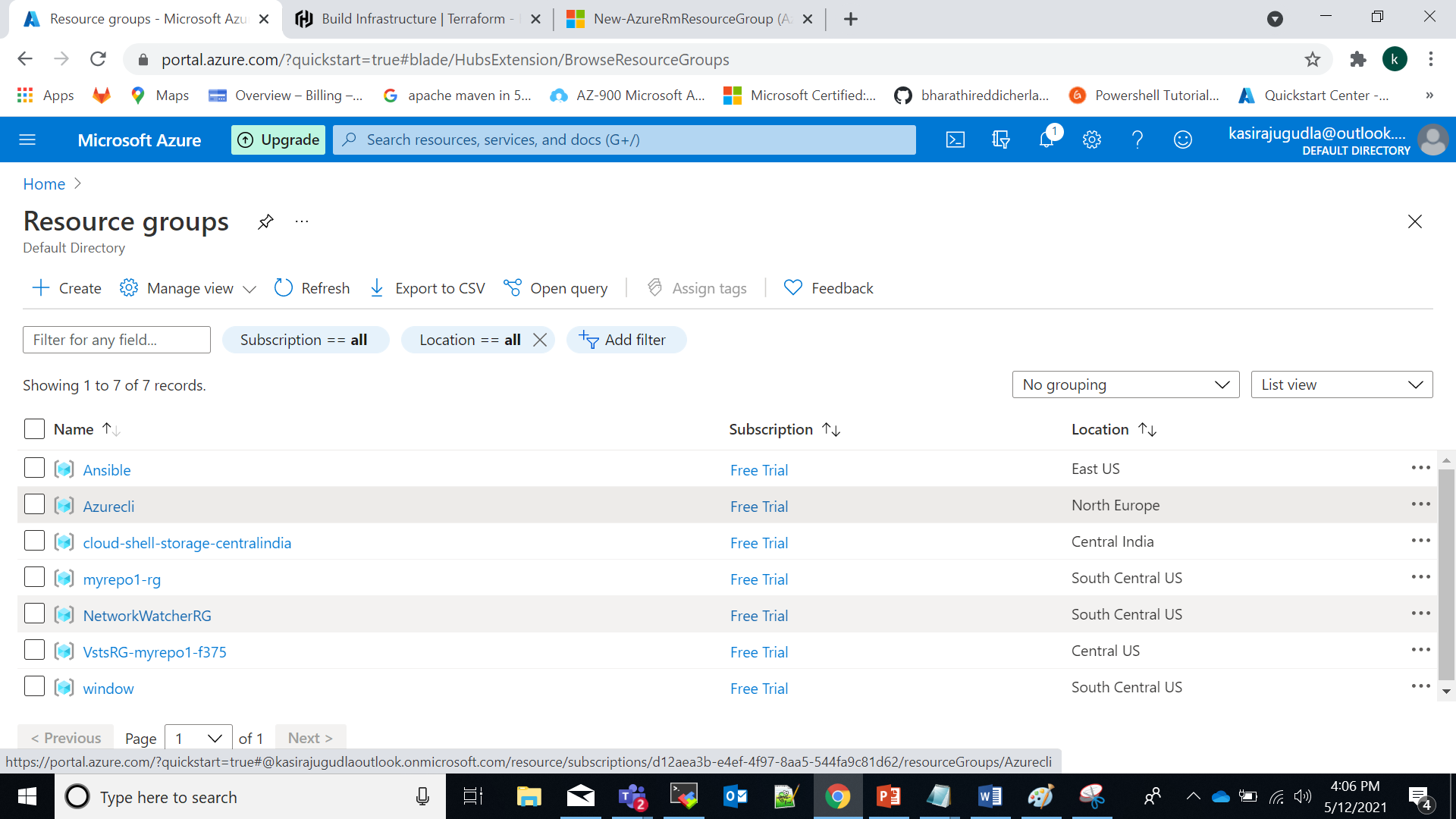




terraform init

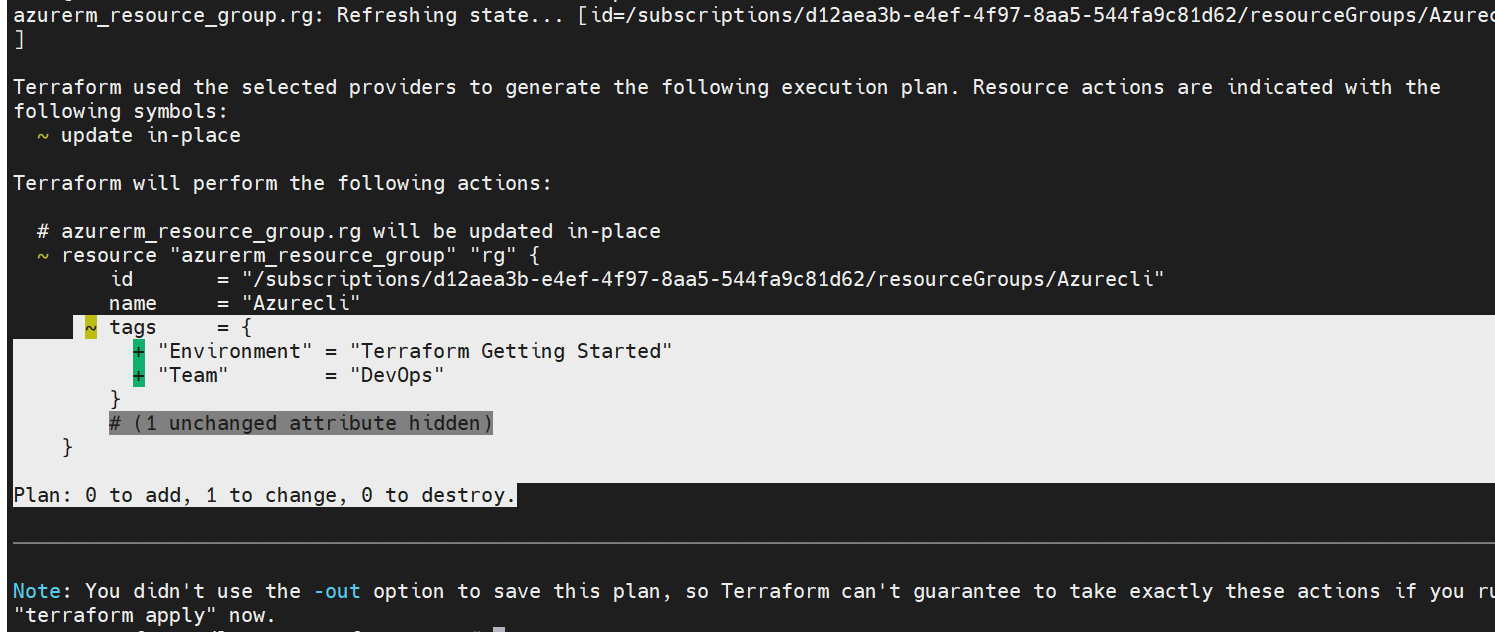
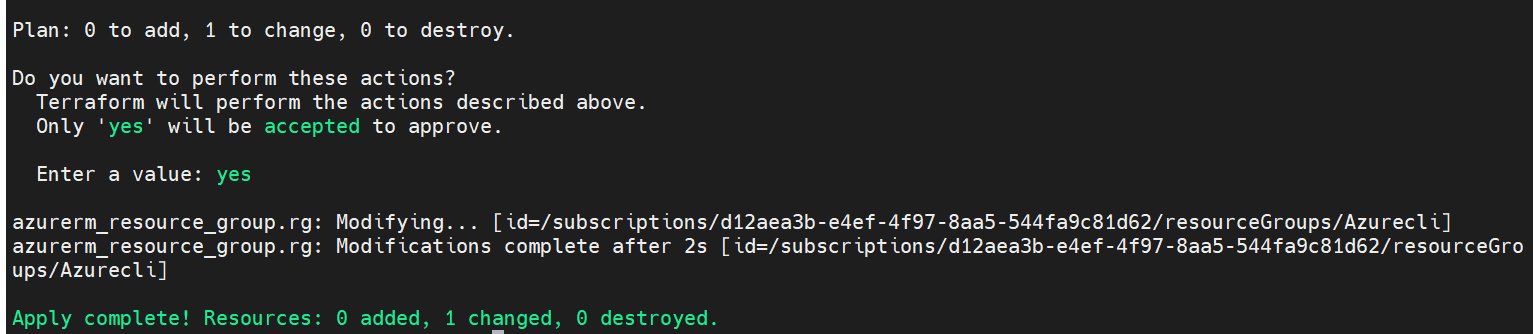
terraform plan

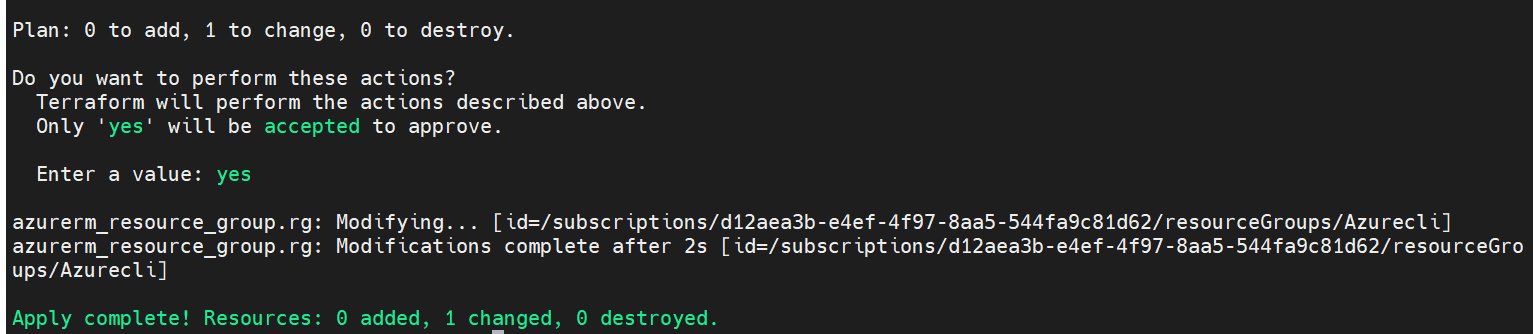
terraform apply

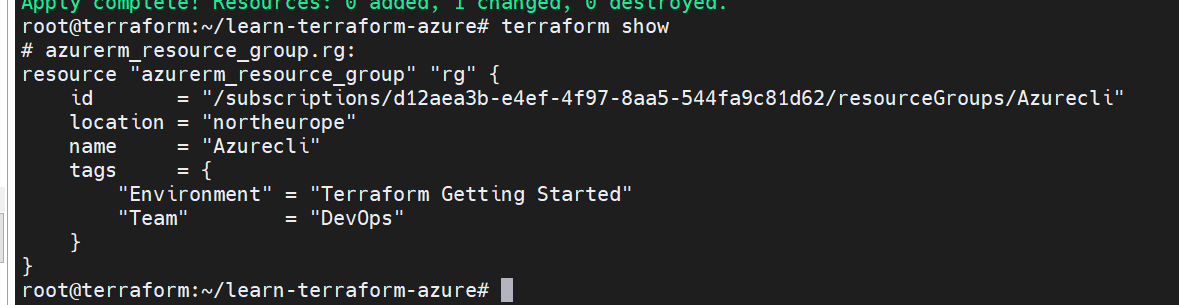


[Change Infrastructure:](https://learn.hashicorp.com/tutorials/terraform/azure-change?in=terraform/azure-get-started)

[Your plan output indicates that the resource will be updated in place with the ~ symbol beside the resource group. Your new resource attributes, indicated with the + symbol, will be added to the resource group.](https://learn.hashicorp.com/tutorials/terraform/azure-change?in=terraform/azure-get-started)

[ ](https://learn.hashicorp.com/tutorials/terraform/azure-change?in=terraform/azure-get-started)







Create resource dependencies:

Azure requires the following underlying resources before you can deploy a virtual machine:

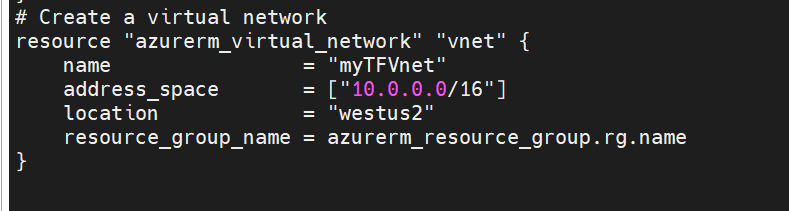
Resource group

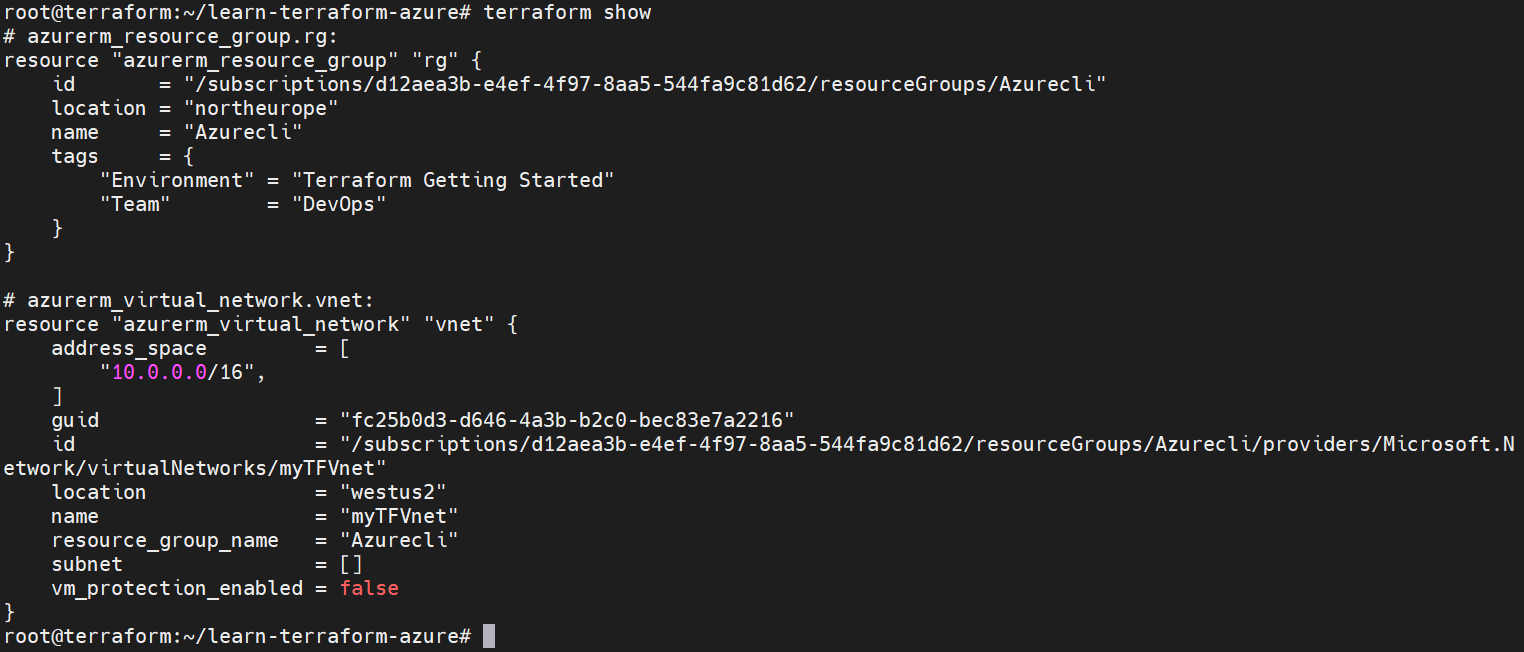
* Virtual network
* Subnet
* Network security group
* Network interface

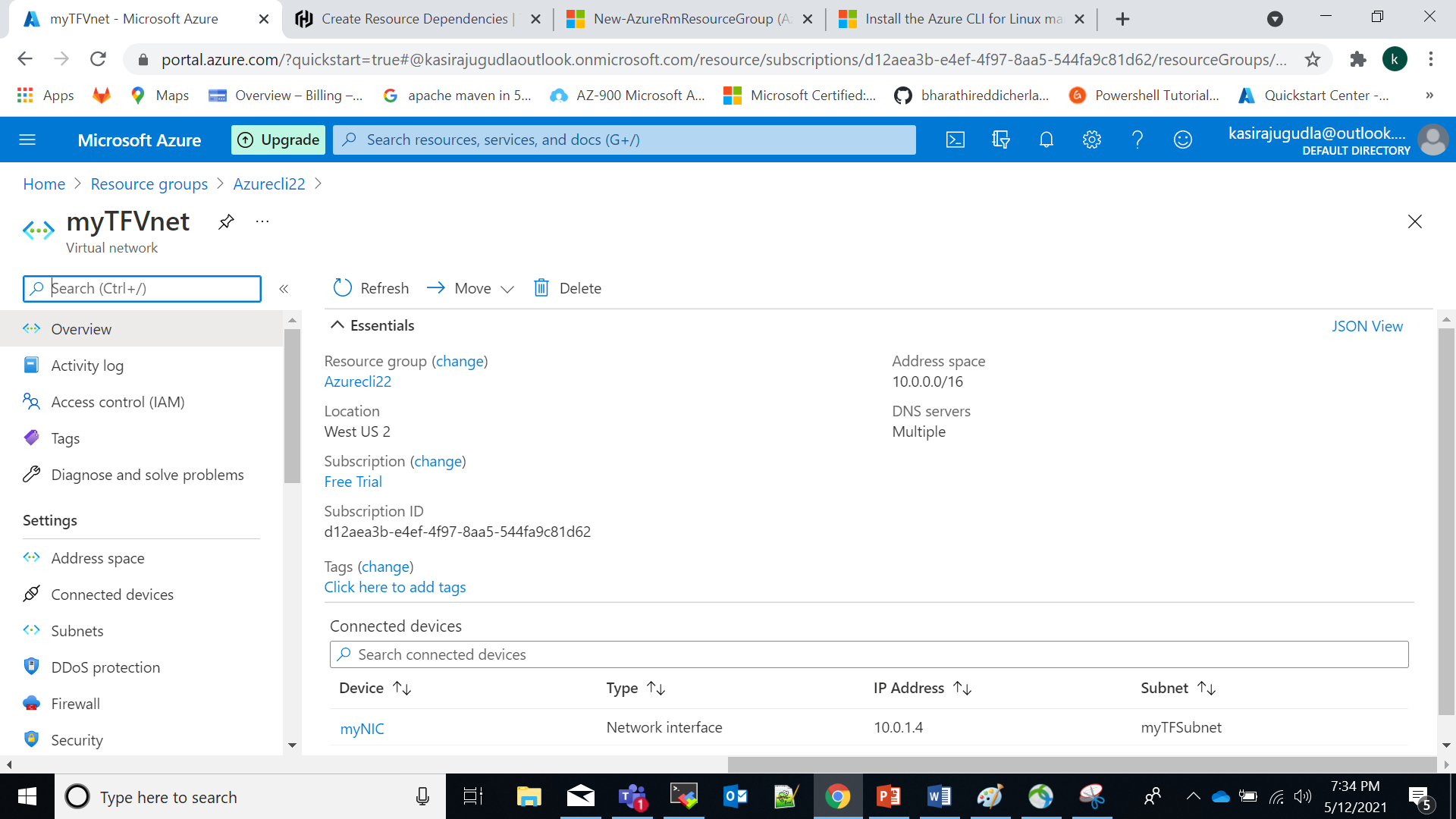
Your deployment will also include a public IP address as well as an explicitly opened port 22 for SSH access.

In your main.tf file, add the resource block below, which creates a virtual network for your virtual machine.

Virtual network: if we want to create new virtual network we can use this script







# Create public IP

resource "azurerm\_public\_ip" "publicip" {

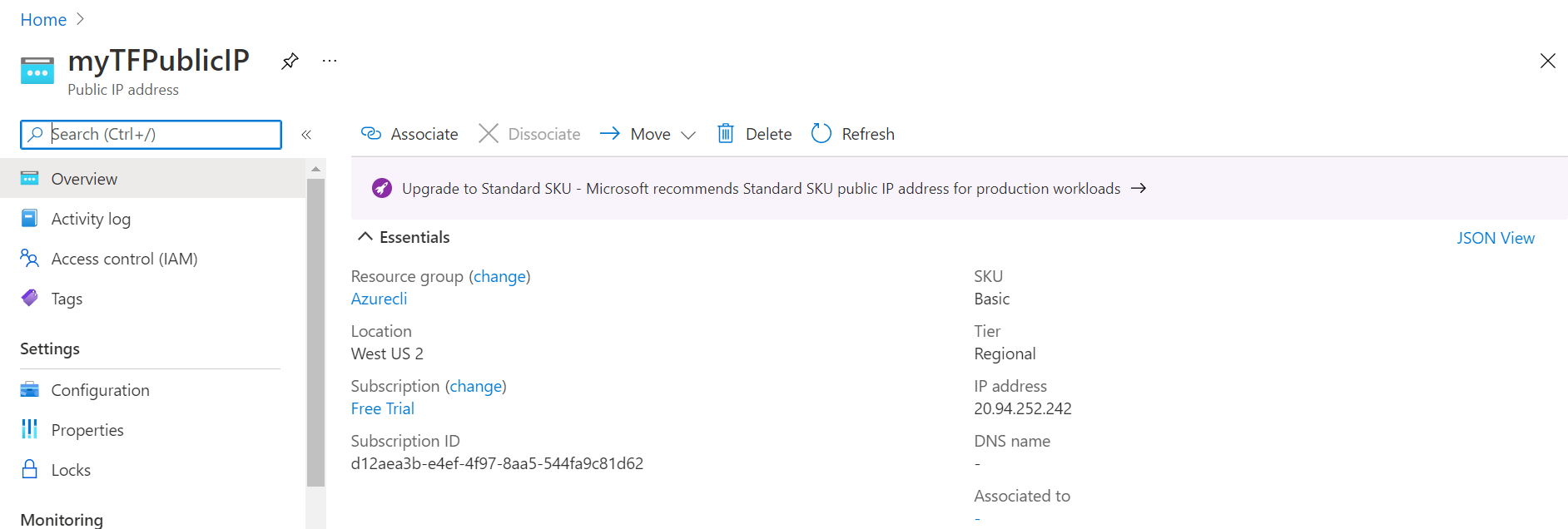
name = "myTFPublicIP"

location = "westus2"

resource\_group\_name = azurerm\_resource\_group.rg.name

allocation\_method = "Static"

}



# Create Network Security Group and rule

resource "azurerm\_network\_security\_group" "nsg" {

name = "myTFNSG"

location = "westus2"

resource\_group\_name = azurerm\_resource\_group.rg.name

security\_rule {

name = "SSH"

priority = 1001

direction = "Inbound"

access = "Allow"

protocol = "Tcp"

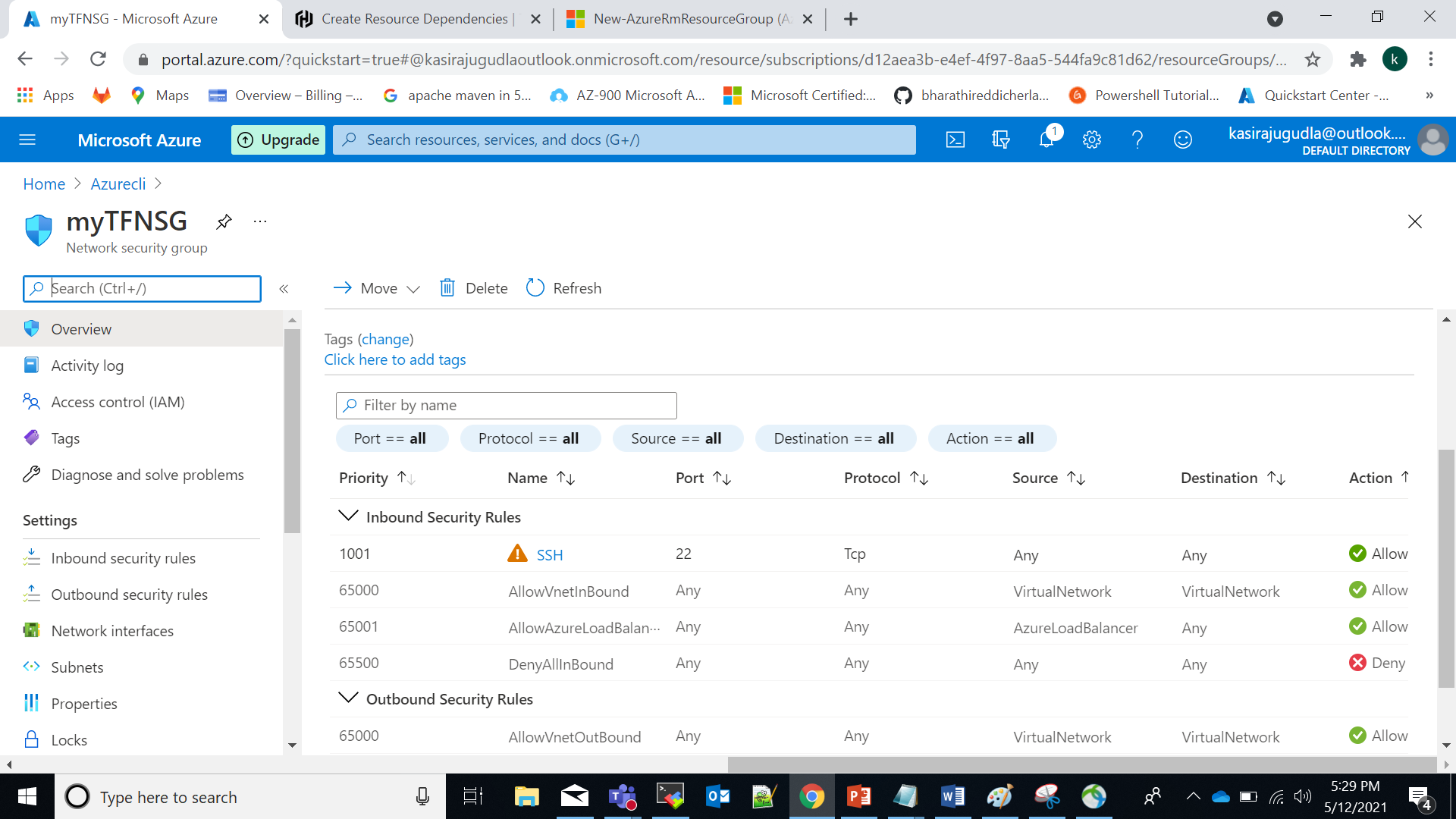
source\_port\_range = "\*"

destination\_port\_range = "22"

source\_address\_prefix = "\*"

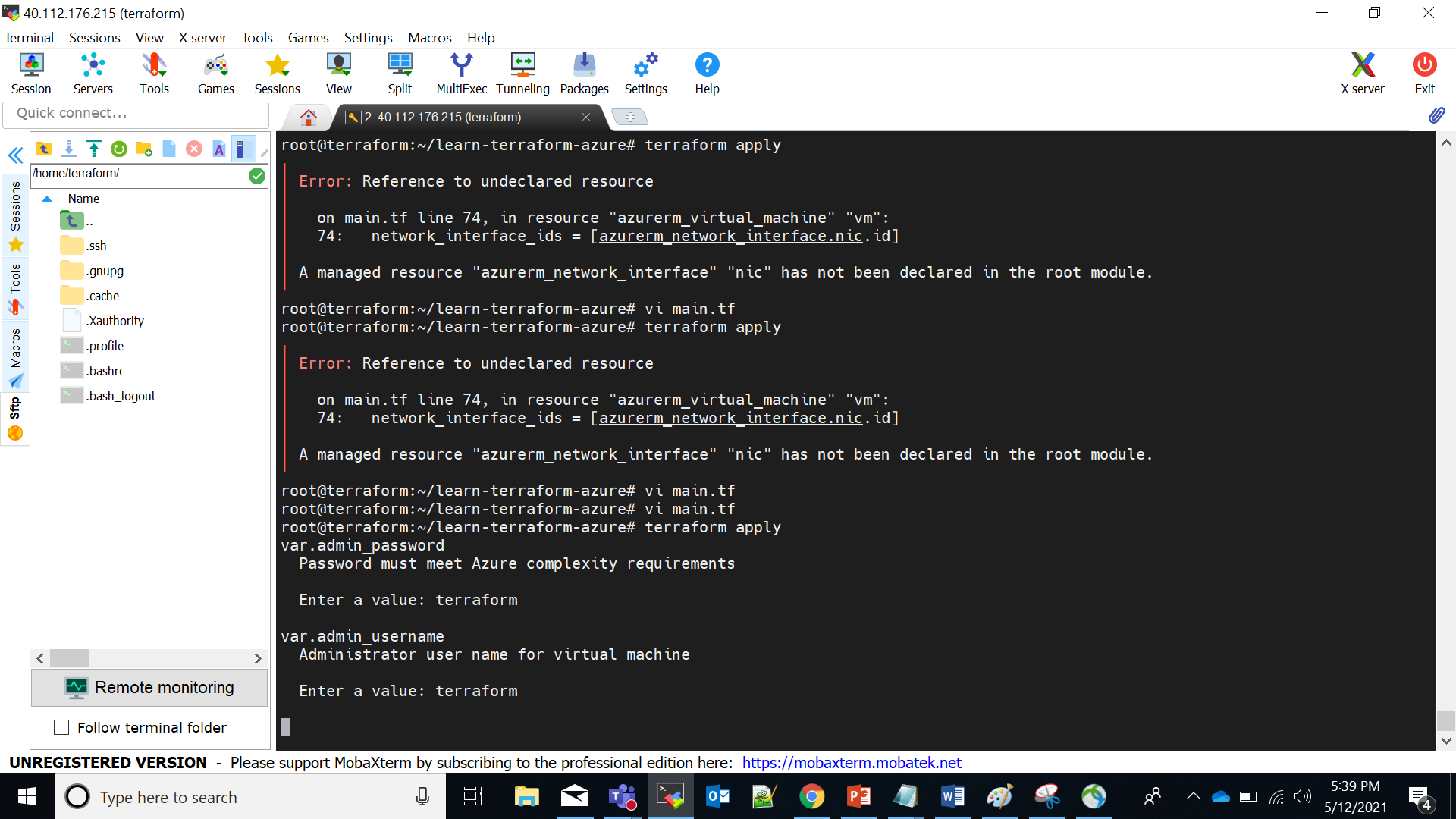
destination\_address\_prefix = "\*"

}



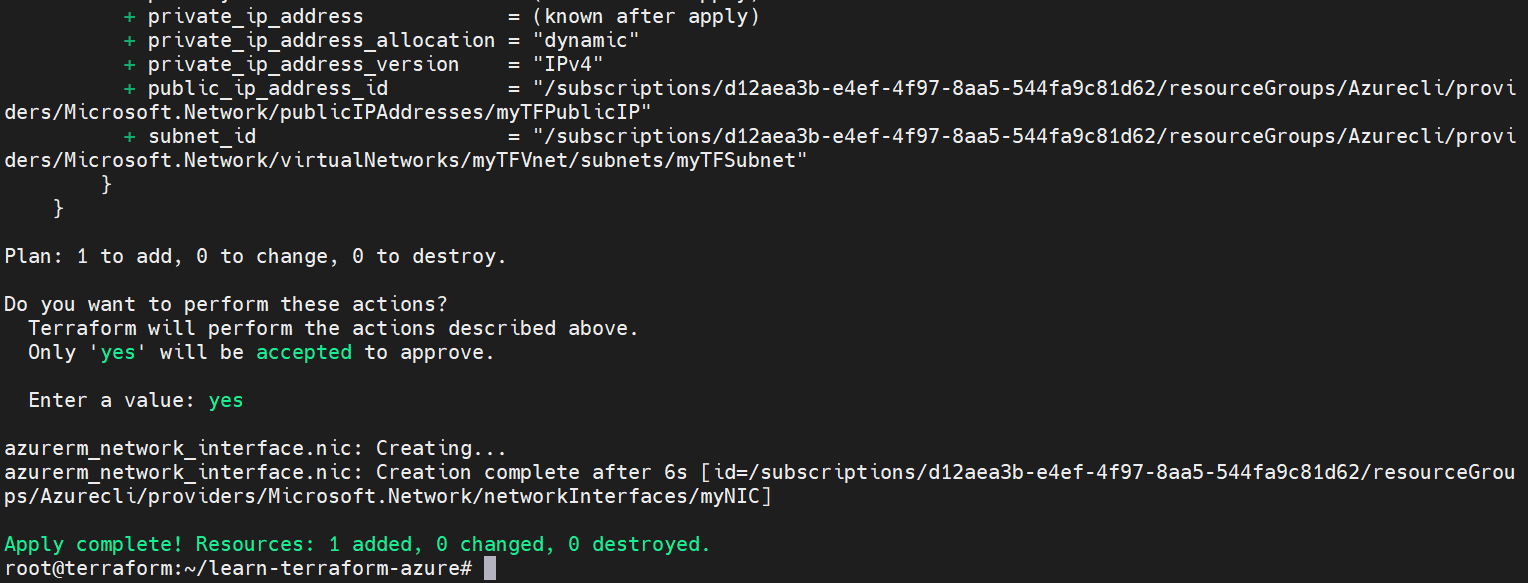
////////////////////////////////////////

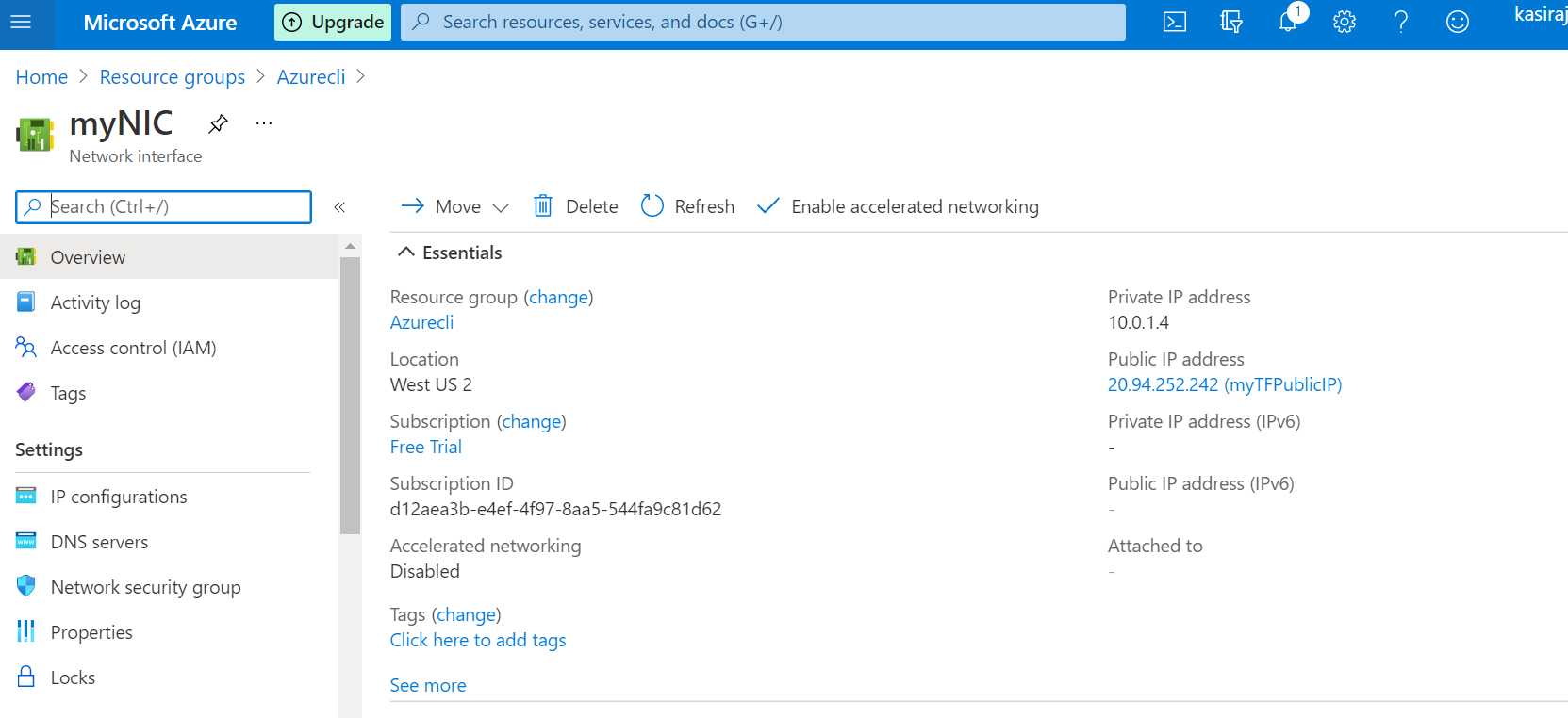
Passing username and value



//////

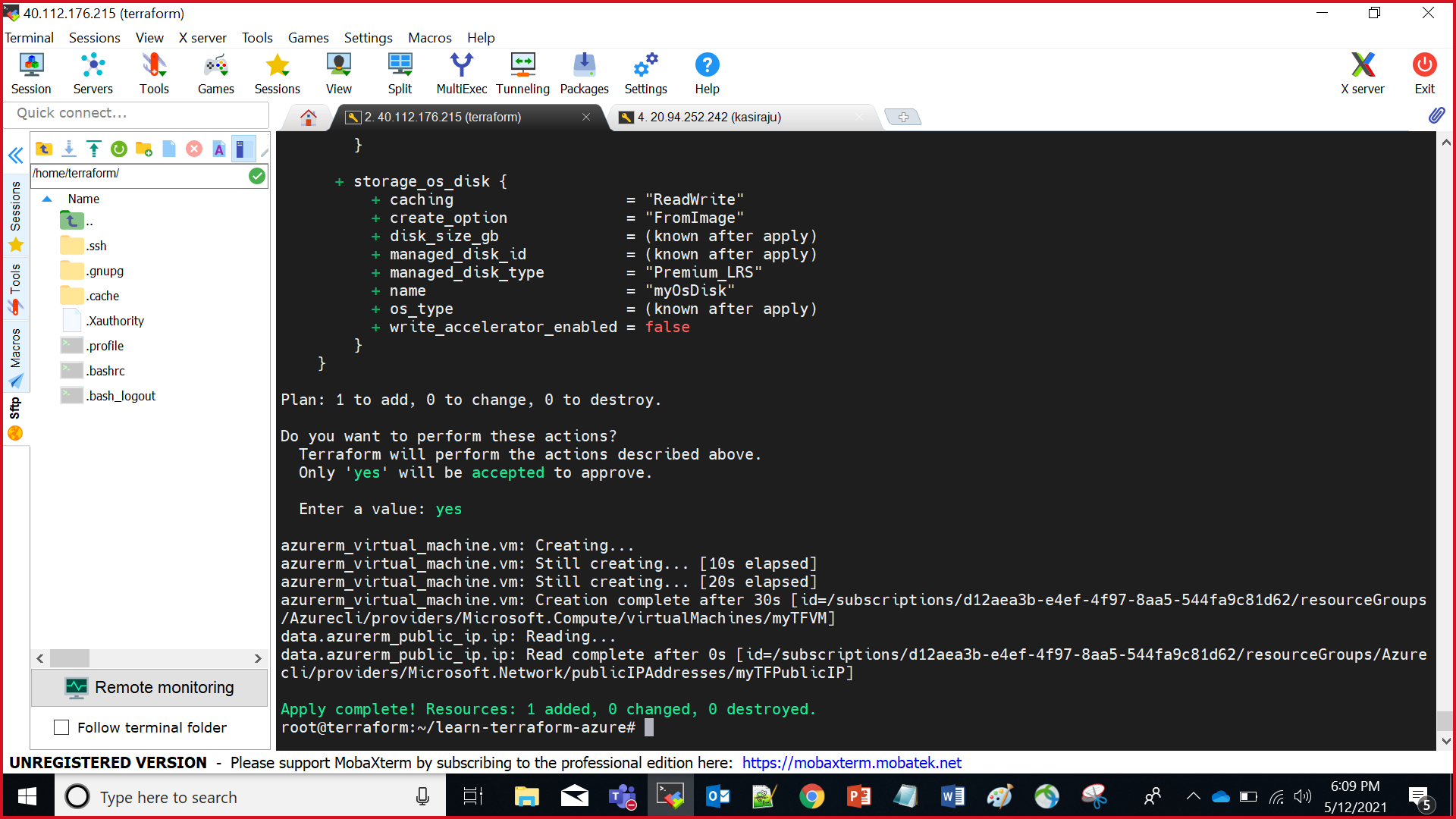
If we want to add Azure network interface

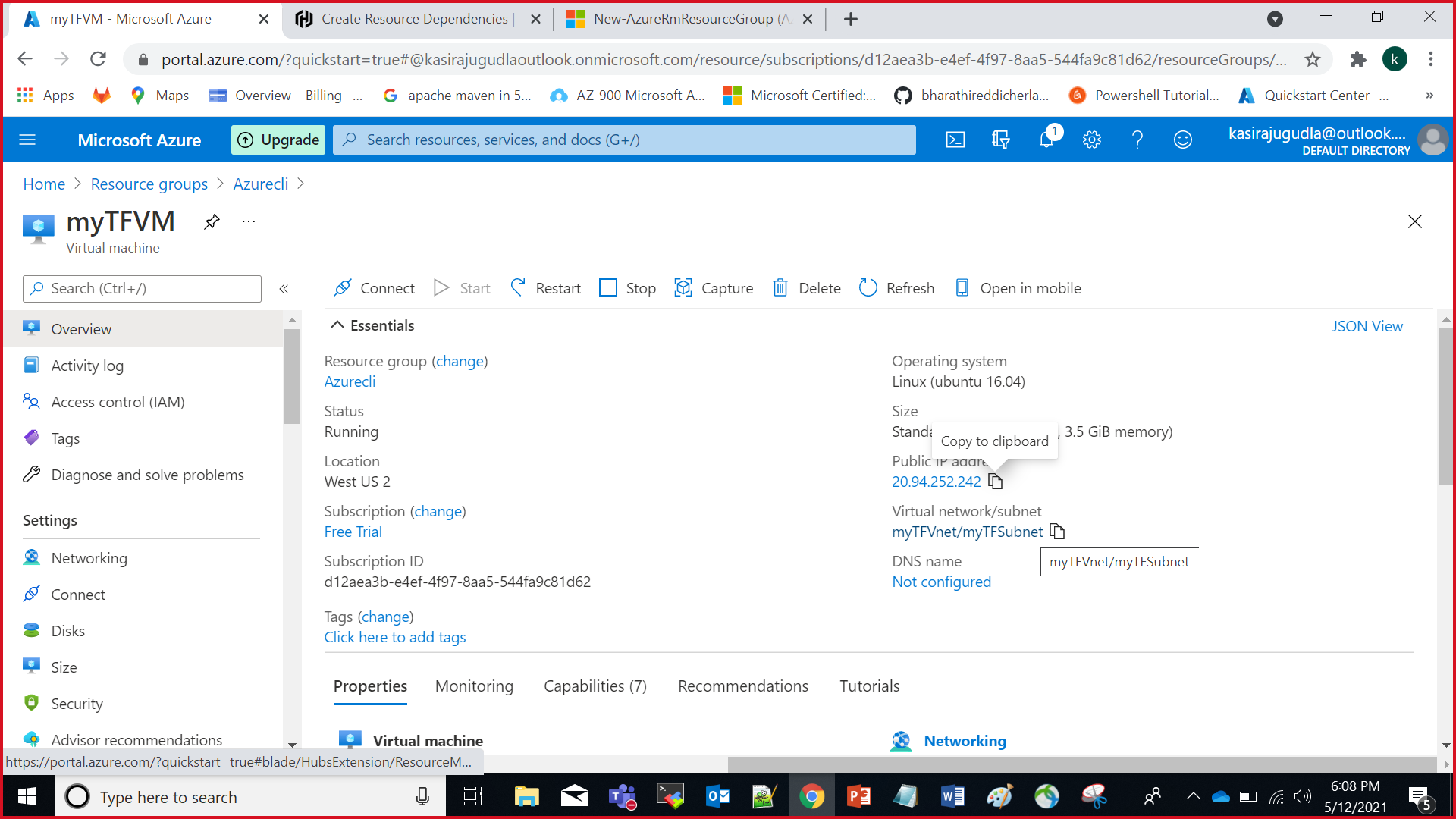


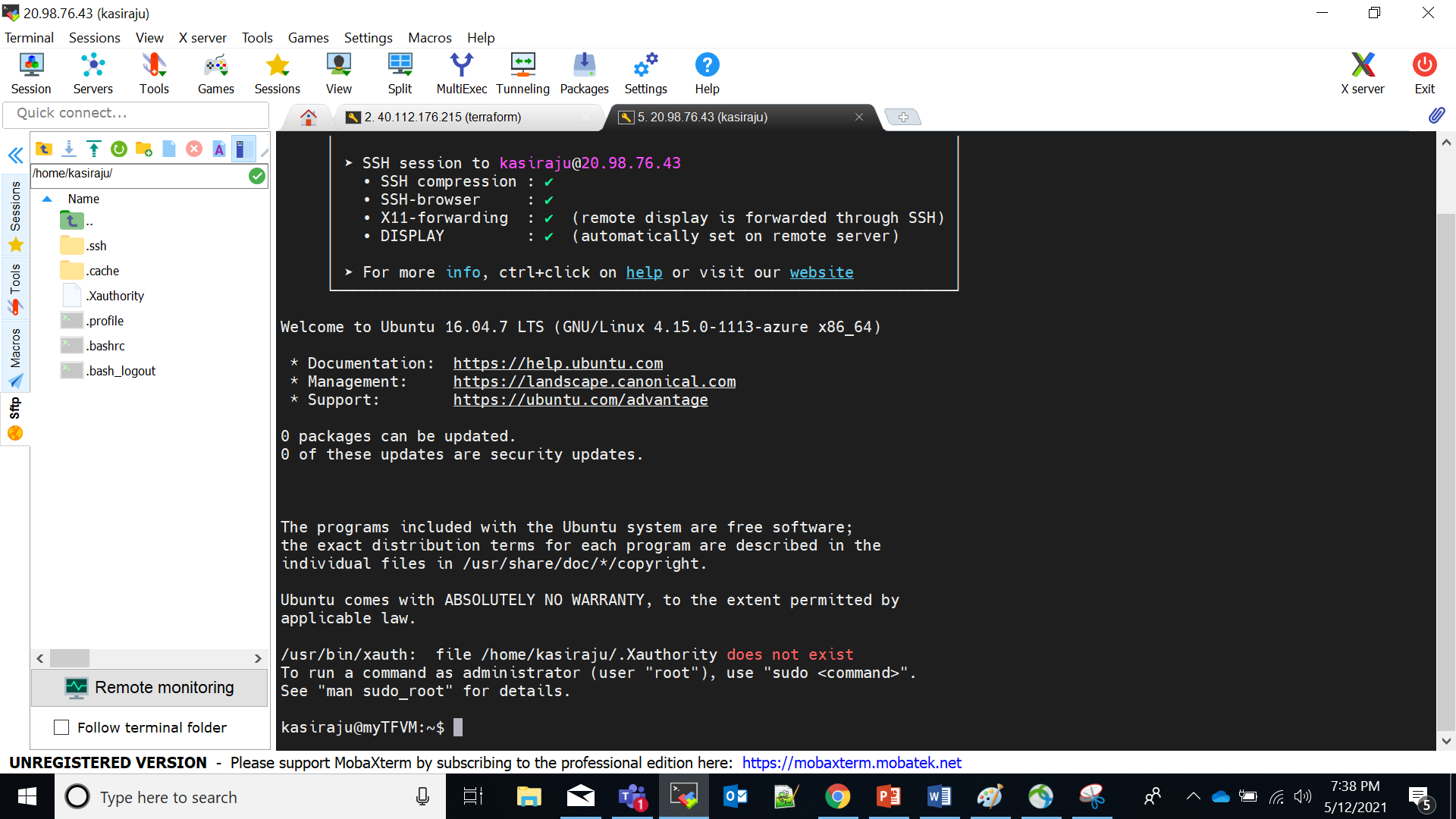


//////////////////////////////////

If we want to create vm we can use above all networks in only one main.tf file





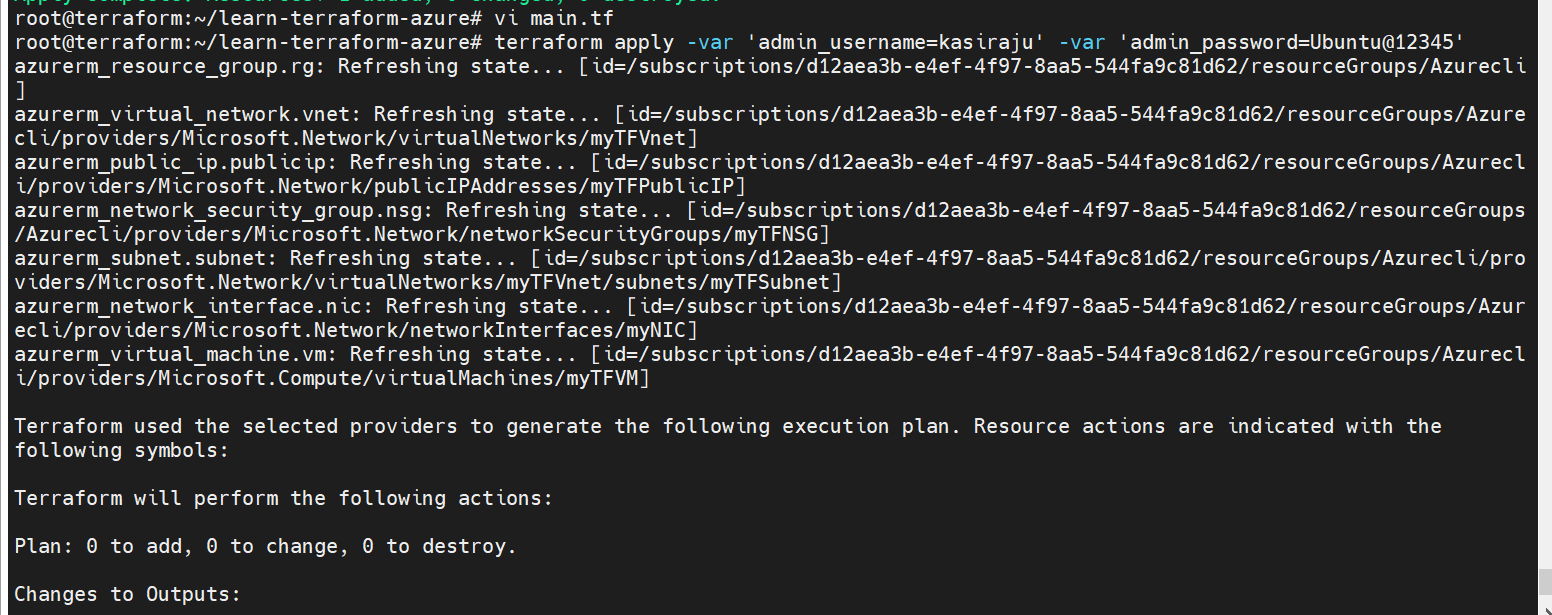


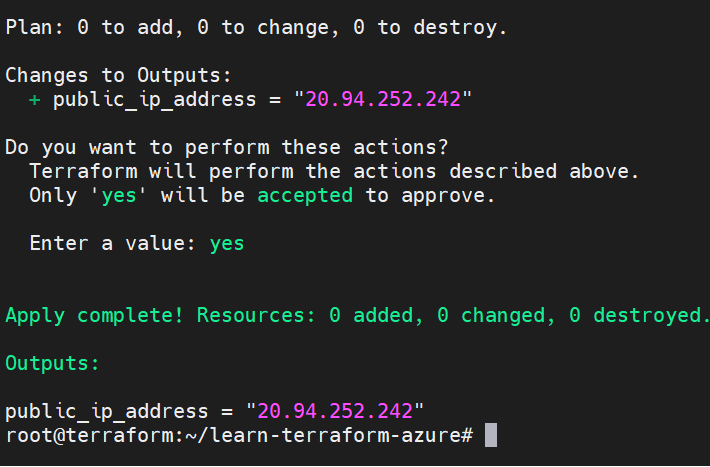
Query data with output variables:

output "public\_ip\_address" {

value = data.azurerm\_public\_ip.ip.ip\_address

}





And finally I was done destroying the all networks again I was create all network with resource group call Azurecli22

