**-------------Hybrid Multi Cloud Computing--------------**

TerraForm Git Jenkins AWS

Hybrid Multi Cloud Task1 Given Mr.Vimal Daga Sir



**Task Description :-**

Task 1 : Have to create/launch Application using Terraform

1. Create the key and security group which allow the port 80.

2. Launch EC2 instance.

3. In this Ec2 instance use the key and security group which we have created in step 1.

4. Launch one Volume (EBS) and mount that volume into /var/www/html

5. Developer have uploded the code into github repo also the repo has some images.

6. Copy the github repo code into /var/www/html

7. Create S3 bucket, and copy/deploy the images from github repo into the s3 bucket and change the permission to public readable.

8 Create a Cloudfront using s3 bucket(which contains images) and use the Cloudfront URL to update in code in /var/www/html

Optional

1) Those who are familiar with jenkins or are in devops AL have to integrate jenkins in this task wherever you feel can be integrated

2) create snapshot of ebs

Above task should be done using terraform

**Pre-requisites For This Project :-**

- Base OS in my case it is Windows 10 and On the Top of Windows 10 I am using Red Hat Enterprise Linux (RHEL 8 ) with the Help of Virtual Box

- Inside Rhel we need some setup to be ready like we need Jenkins , and In this Rhel 7 Os I have installed TerraForm and Aws CLI . need to Configure the aws

In this os and I have created profile so u can also that in explanation Below

- First I have Tried EveryThing on my BasOs that is Windows & then had created its file as maybel.tf and pushed to github and then pulled at rhel and repeated whole work again there after this whole thing I explained in this Documentation

Do Please checl it till lasT .

So Let’s Start , Here I am Starting From my terraform file ie. maybel.tf and I will be explaining in Between

**For Better view Open in Accessibility mode or Download it .**

**Provider and linking to my profile that I created :-**

provider "aws" {

profile = "abhinew"

region = "ap-south-1"

**Creating Security group with allowing Port 80 For http and Port 22 For ssh :-**

resource "aws\_security\_group" "sec\_grp" {

name = "sec\_grp"

description = "Allows SSH and HTTP"

vpc\_id = "vpc-0d8f9265"

ingress {

description = "SSH"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = [ "0.0.0.0/0" ]

}

ingress {

description = "HTTP"

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = [ "0.0.0.0/0" ]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

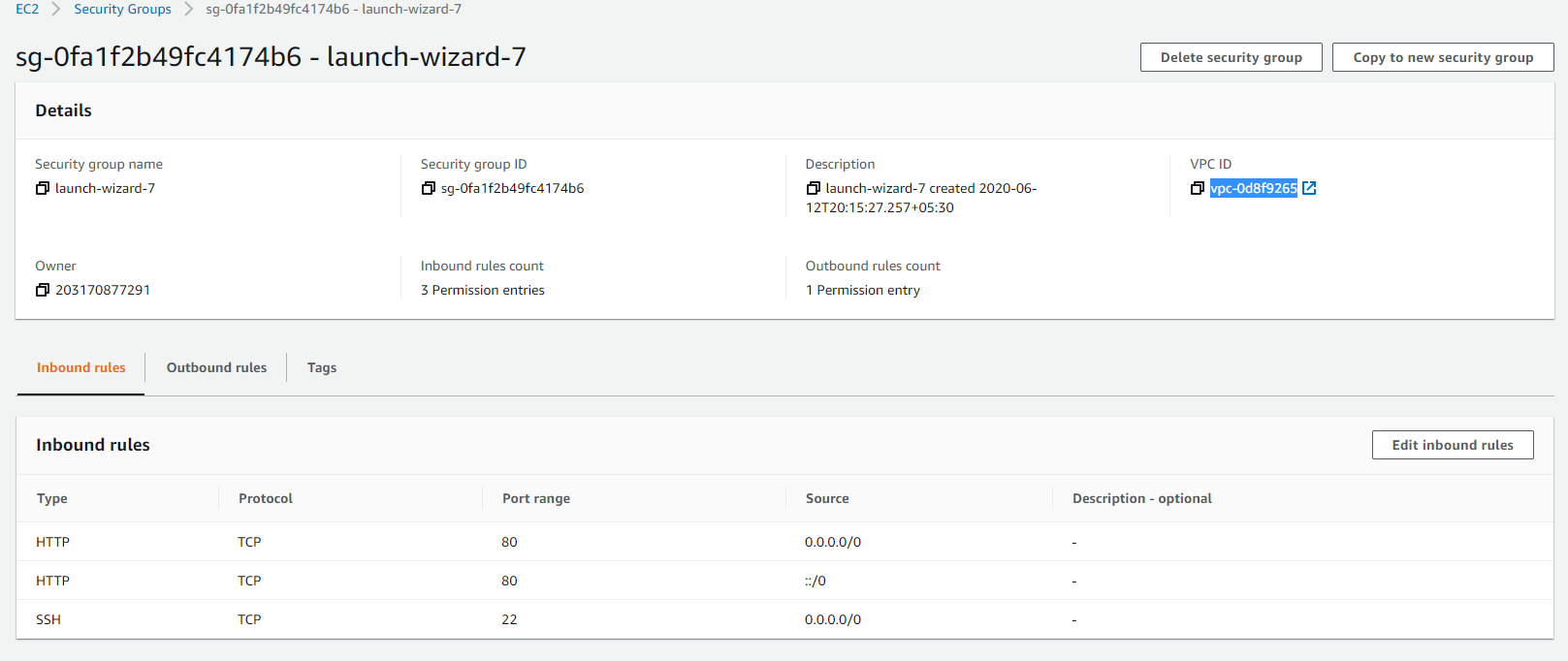
}

tags = {

Name = "sec\_grp"

}

}



Creating key-pairs For Furthur Use :-

resource "tls\_private\_key" "mykey" {

algorithm = "RSA"

}

resource "aws\_key\_pair" "generated\_key" {

key\_name = "mykey"

public\_key = "${tls\_private\_key.mykey.public\_key\_openssh}"

depends\_on = [

tls\_private\_key.mykey

]

}

resource "local\_file" "key-file" {

content = "${tls\_private\_key.mykey.private\_key\_pem}"

filename = "mykey.pem"

depends\_on = [

tls\_private\_key.mykey

]

}

In this screenshot here are 3 keys but the key that crated automatically by terraform us mykey u can match from code also



Creating Instance Using Above Details with some installed softwares like Httpd , Git.:-

resource "aws\_instance" "AbhiOs1" {

ami = "ami-0447a12f28fddb066"

instance\_type = "t2.micro"

key\_name = aws\_key\_pair.generated\_key.key\_name

security\_groups = [ aws\_security\_group.sec\_grp.name ]

provisioner "remote-exec" {

connection {

agent = "false"

type = "ssh"

user = "ec2-user"

private\_key = "${tls\_private\_key.mykey.private\_key\_pem}"

host = "${aws\_instance.AbhiOs1.public\_ip}"

}

inline = [

"sudo yum install httpd php git -y",

"sudo systemctl restart httpd",

"sudo systemctl enable httpd",

]

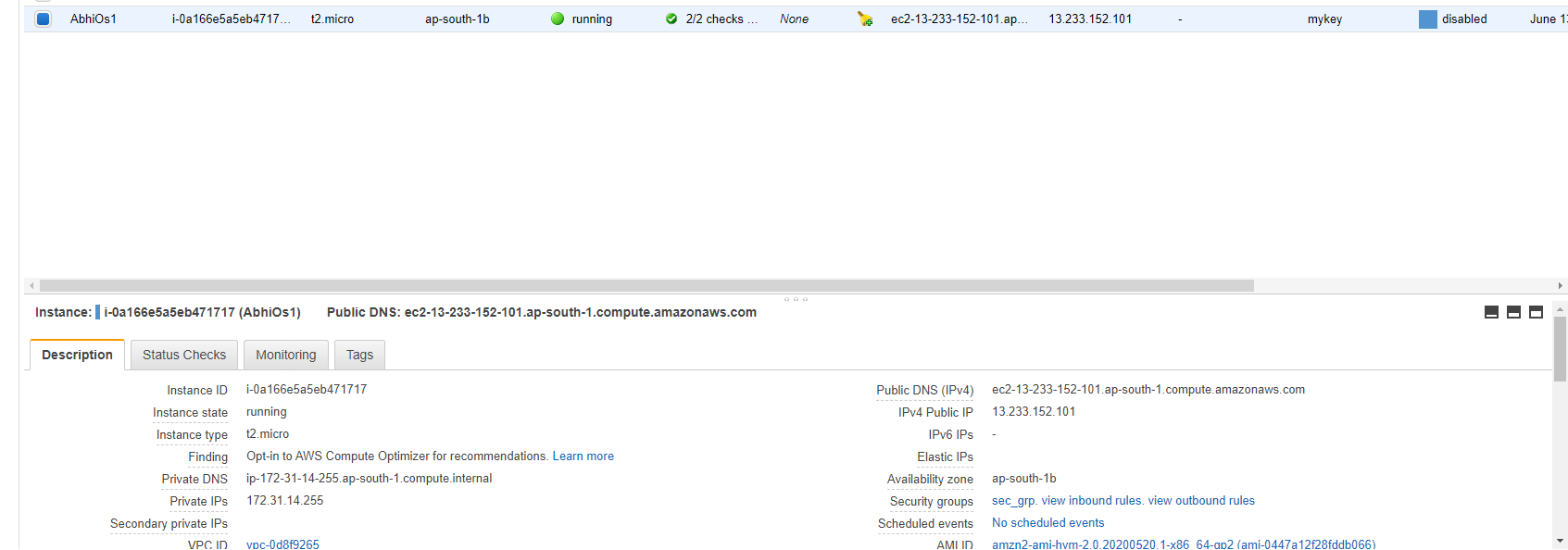
}

tags = {

Name = "AbhiOs1"

}

}



**Creating EBS Volume :-**

resource "aws\_ebs\_volume" "myebs1" {

availability\_zone = aws\_instance.AbhiOs1.availability\_zone

size = 1

tags = {

Name = "ebsvol"

}

}

**Attaching Volume :-**

resource "aws\_volume\_attachment" "attach\_ebs" {

device\_name = "/dev/sdh"

volume\_id = aws\_ebs\_volume.myebs1.id

instance\_id = aws\_instance.AbhiOs1.id

force\_detach = true

}

**Getting ip of instance on screen :-**

output "myip" {

value = aws\_instance.AbhiOs1.public\_ip

**}**

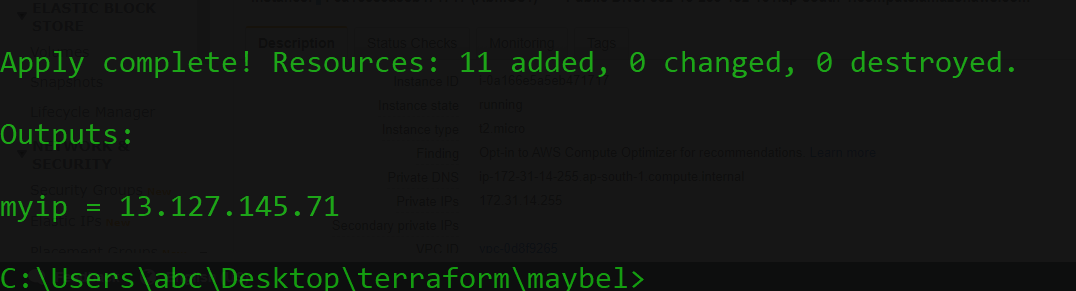
resource "null\_resource" "nullip" {

provisioner "local-exec" {

command = "echo ${aws\_instance.AbhiOs1.public\_ip} > publicip.txt"

}

}



**Now Connecting Using ssh & Downloading the Code ..:-**

connection {

agent = "false"

type = "ssh"

user = "ec2-user"

private\_key = "${tls\_private\_key.mykey.private\_key\_pem}"

host = "${aws\_instance.AbhiOs1.public\_ip}"

}

provisioner "remote-exec" {

inline = [

"sudo mkfs.ext4 /dev/xvdh",

"sudo mount /dev/xvdh /var/www/html",

"sudo rm -rf /var/www/html/\*",

"sudo git clone https://github.com/a1-s2/Hybrid\_Cloud\_Task1.git /var/www/html"

]

}

}

**Creatin Bucket :-**

resource "aws\_s3\_bucket" "terra-buckcbuck" {

bucket = "abhibucketnew"

acl = "public-read"

versioning {

enabled = true

}

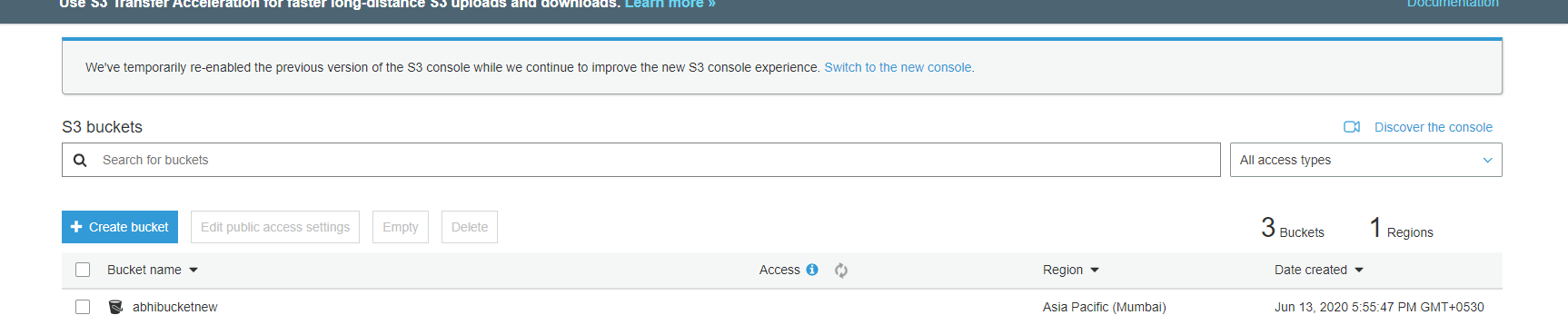
tags = {

Name = "my-new-buck"

Environment = "Dev"

}

}



**Creating CloudFront :-**

resource "aws\_cloudfront\_distribution" "tera-cloufront1" {

origin {

domain\_name = "abhibucketnew.s3.amazonaws.com"

origin\_id = "S3-abhibucketnew"

custom\_origin\_config {

http\_port = 80

https\_port = 80

origin\_protocol\_policy = "match-viewer"

origin\_ssl\_protocols = ["TLSv1", "TLSv1.1", "TLSv1.2"]

}

}

enabled = true

default\_cache\_behavior {

allowed\_methods = ["DELETE", "GET", "HEAD", "OPTIONS", "PATCH", "POST", "PUT"]

cached\_methods = ["GET", "HEAD"]

target\_origin\_id = "S3-abhibucketnew"

forwarded\_values {

query\_string = false

cookies {

forward = "none"

}

}

viewer\_protocol\_policy = "allow-all"

min\_ttl = 0

default\_ttl = 3600

max\_ttl = 86400

}

restrictions {

geo\_restriction {

restriction\_type = "none"

}

}

viewer\_certificate {

cloudfront\_default\_certificate = true

}

}



**This is Used For Auto Webpage Open in FireFox Browser :-**

resource "null\_resource" "nullremote" {

depends\_on = [

null\_resource.nullmount,

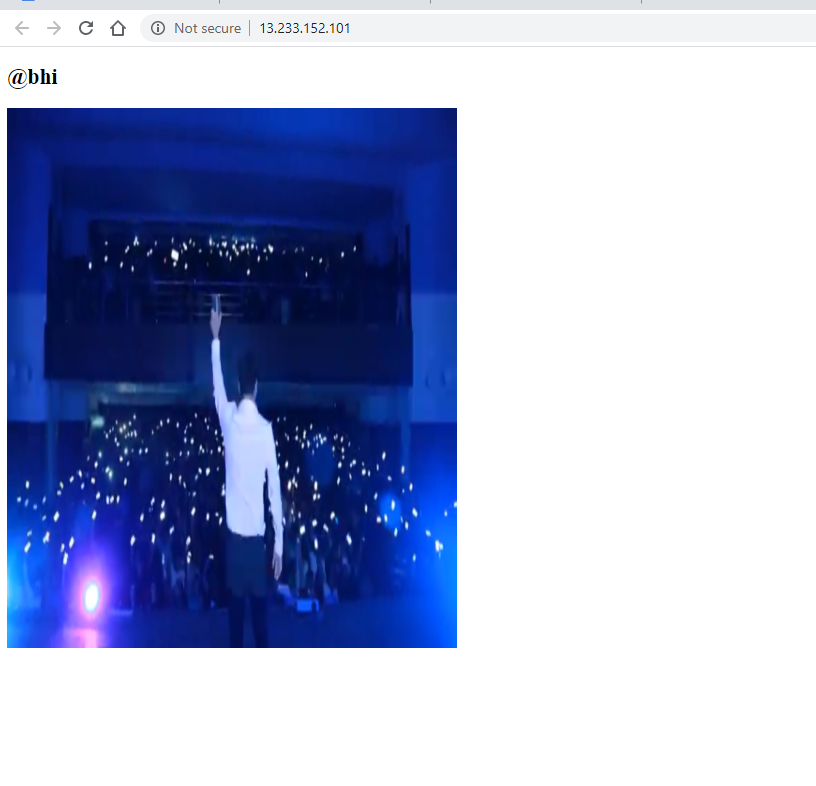
]

provisioner "local-exec" {

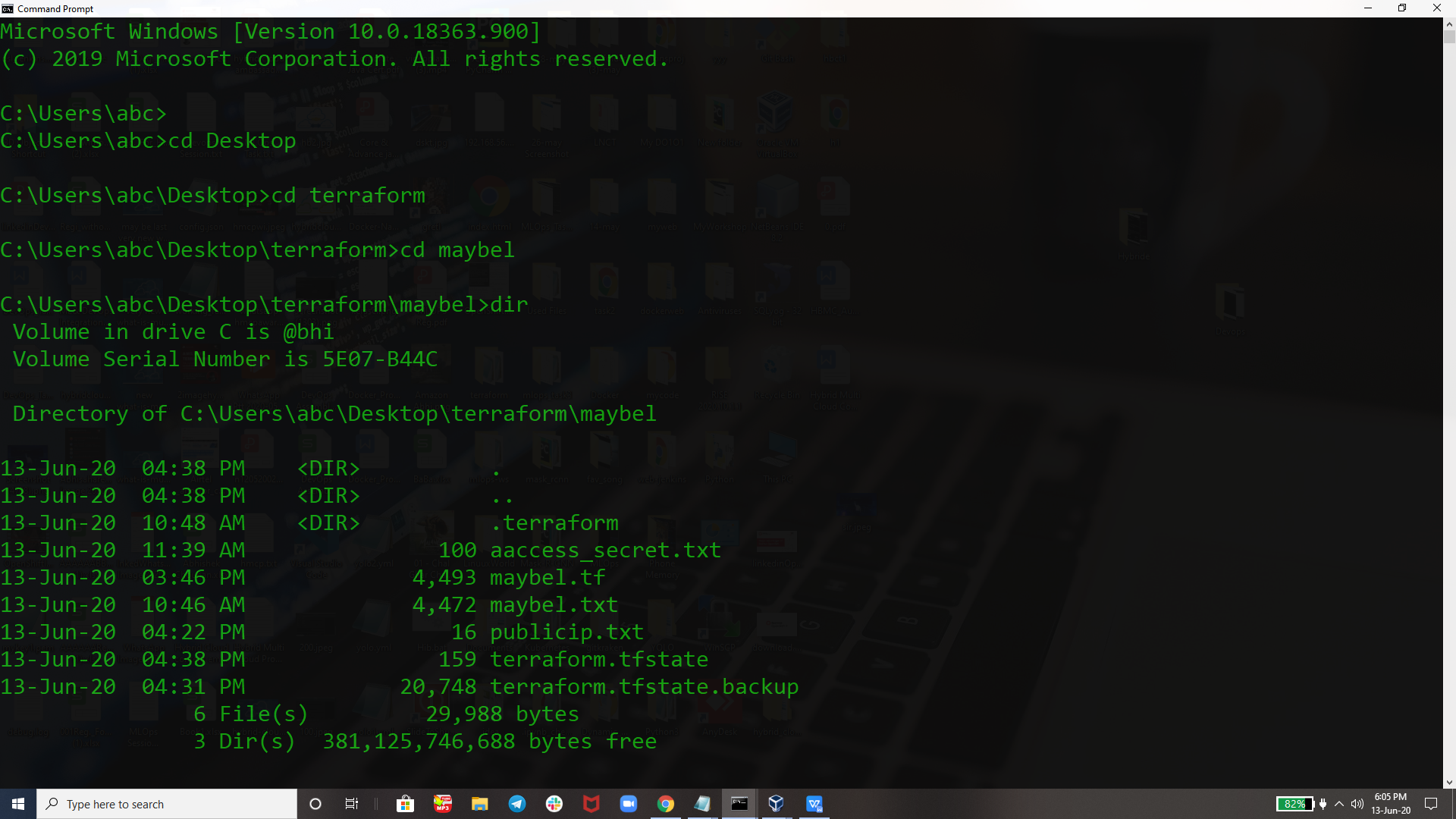
command = "firefox ${aws\_instance.os1.public\_ip}"

}

}

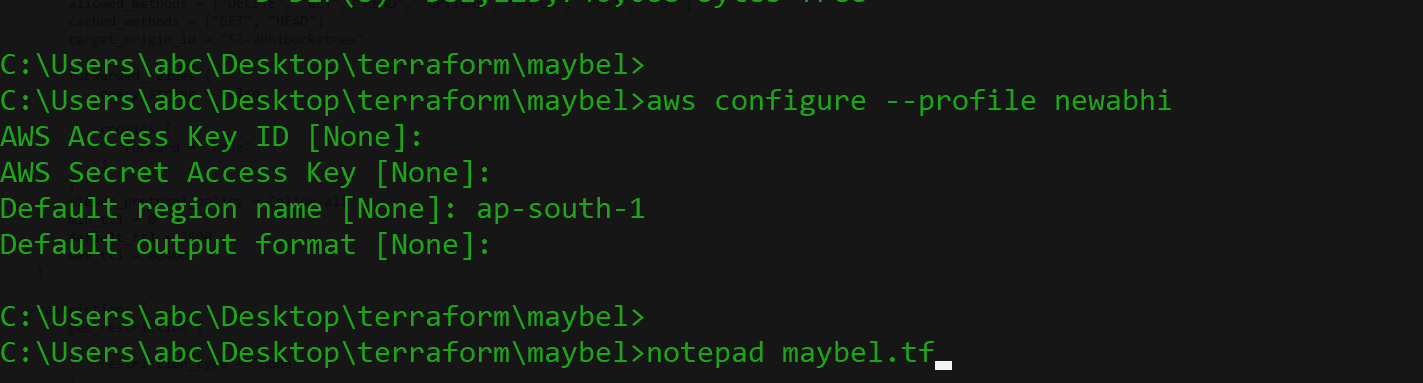


**Going Inside The Directory where we have our .tf File :-**



**Configuring Profile and Opening our TerraForm Configuration File :-**

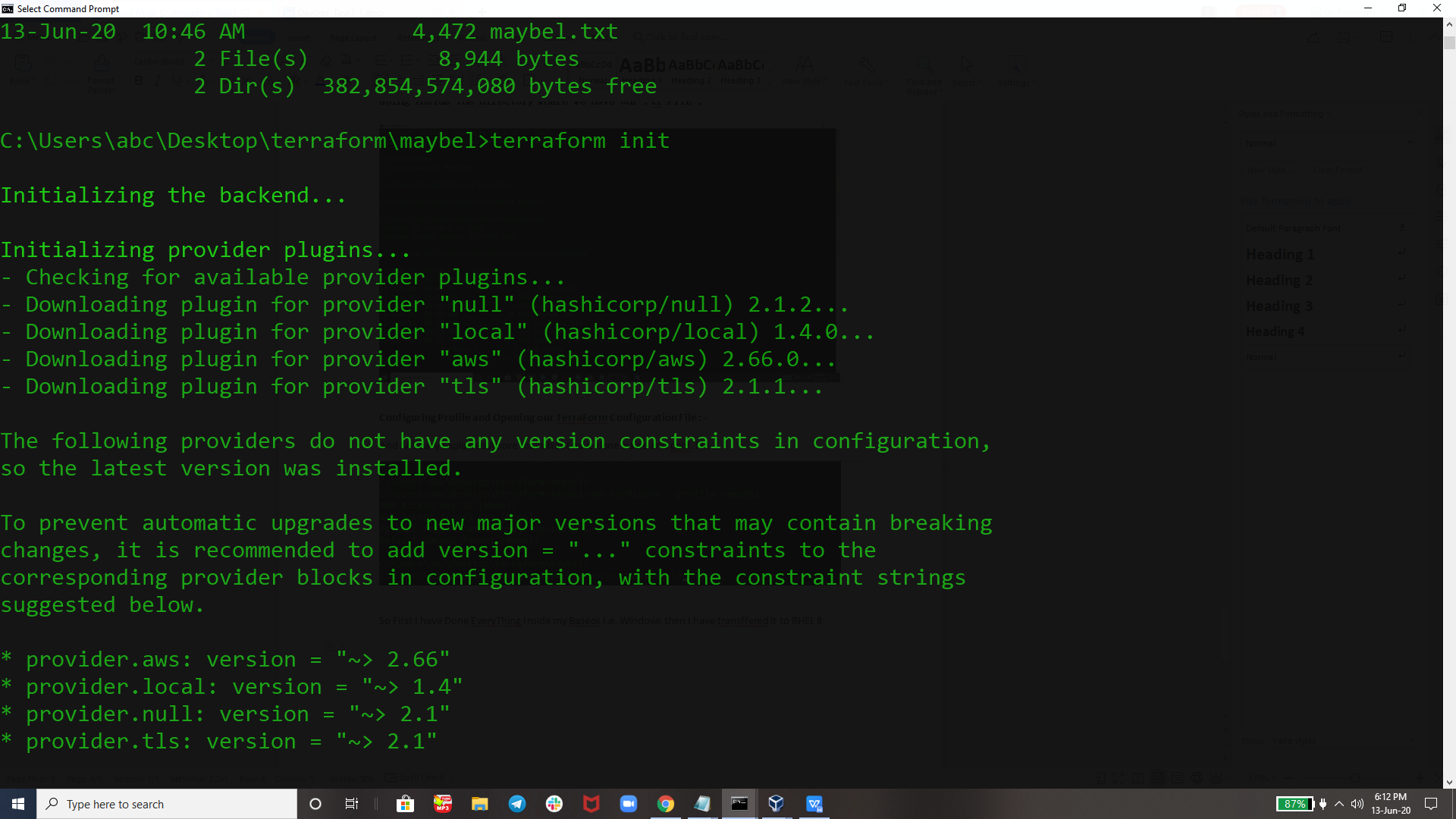
And I Already Explained Above that what I have done inside and conf. file



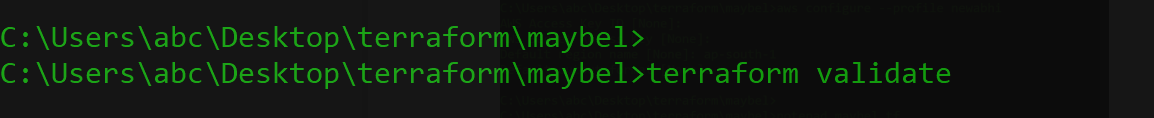
So First I have Done EveryThing Inside my Baseos I.e. Windows then I have transffered it to RHEL 8

As we Are Executing this Conf. File First Time so we we need some plugin according to this conf. File

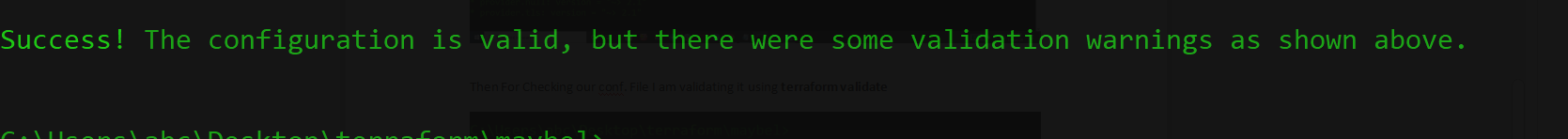
So we have to run **terraform init c**ommand.it will download all the plugin required for executing this conf file..



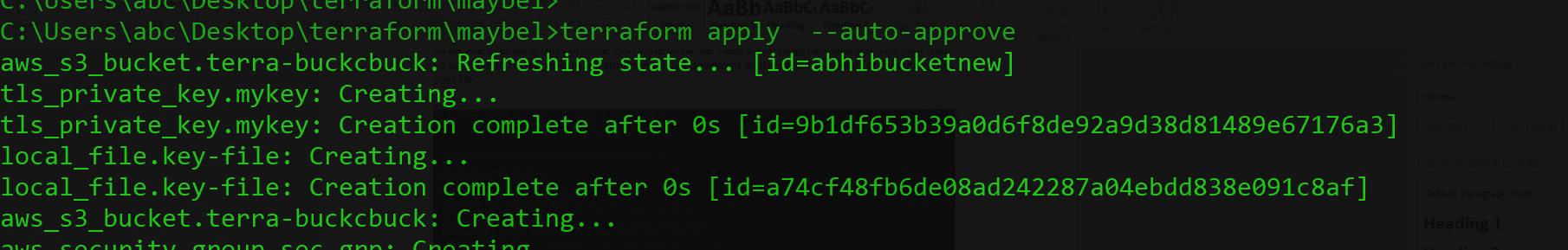
Then For Checking our conf. File I am validating it using **terraform validate**



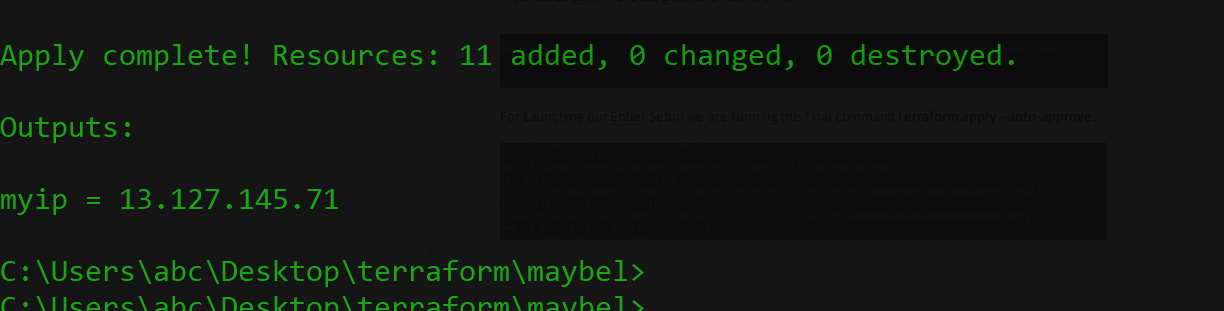
It comes as conf. File looks good and has no error.



For Launching our Entier Setup we are running this final command **terraform apply --auto-approve .**



Its Been Succesful and giving output ip of our instance saying that ur entire setup has been succesfully launched.



Now For Automation the Things I am using Jenkins & I will Explain EveryThing Their ..

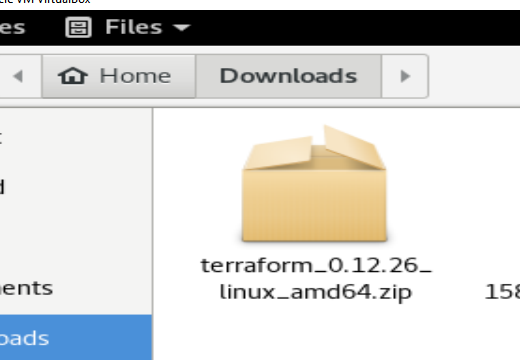
So running everything in Rhel 8 we need some setup to be ready like

- Terraform

- Aws CLI

- Jenkins

So I have downloaded terraform zip file ..

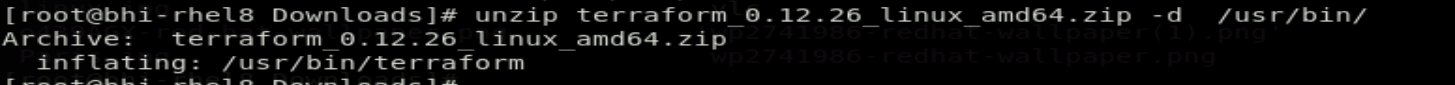


Then we need to unzip for getting terraform to be executable and set its path..

Terraform zip file



Now unzip it and set its path



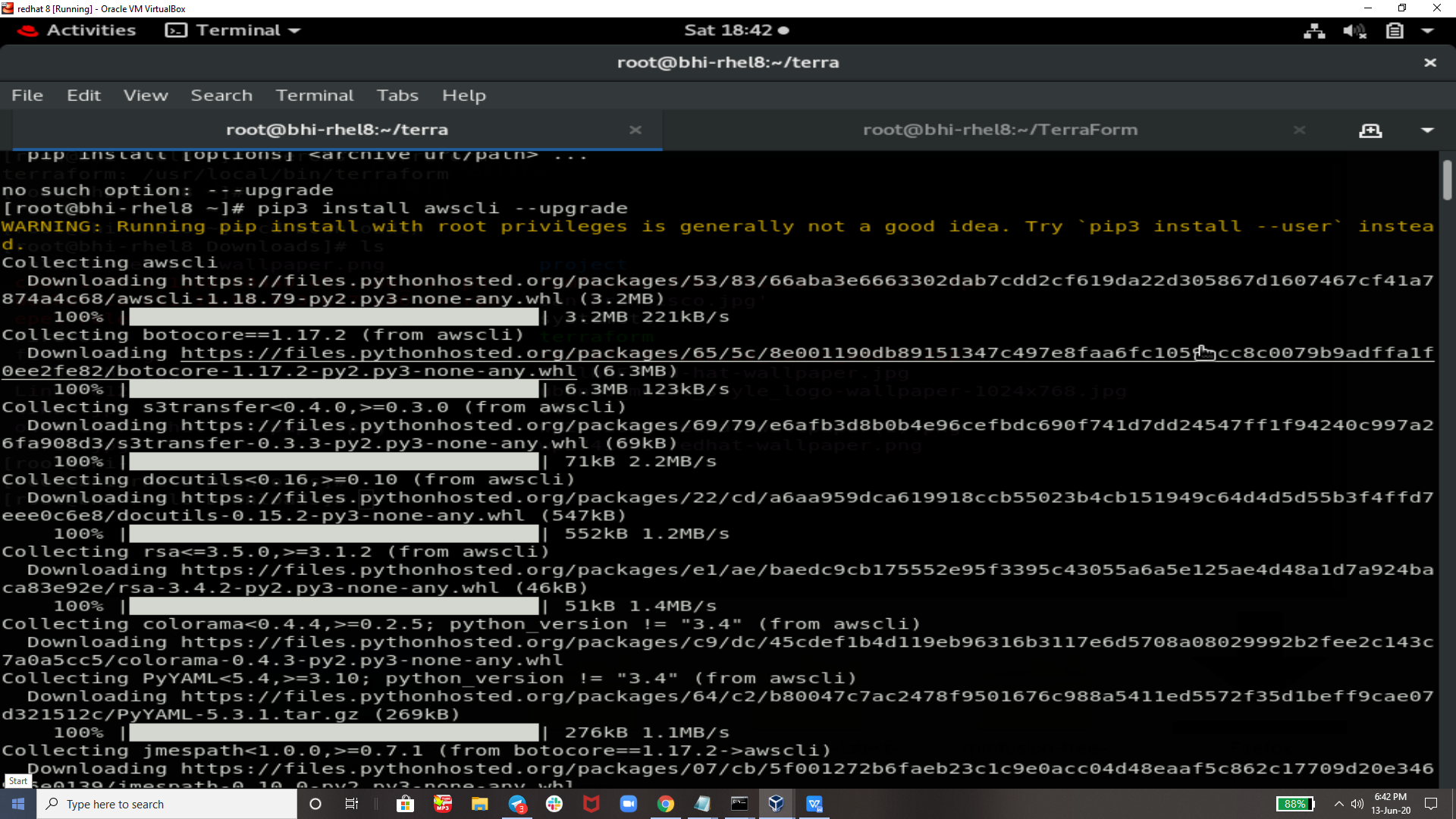
Now we have terraform executable file



**Using chmod +x terraform** command to make it executable



Now installing aws cli

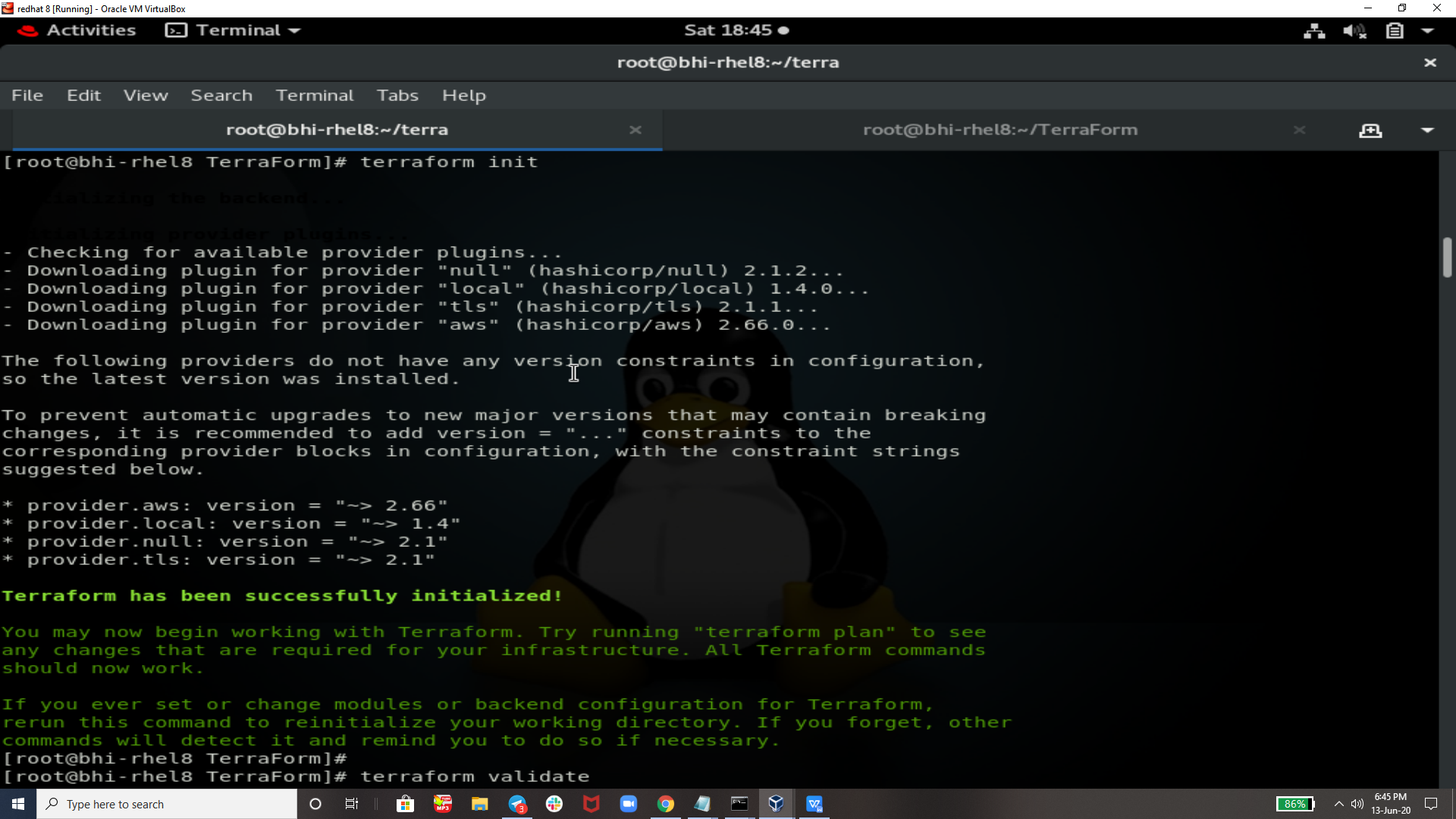




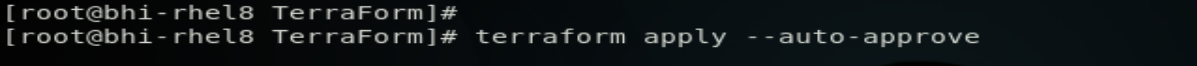
Checking aws cli version that we have

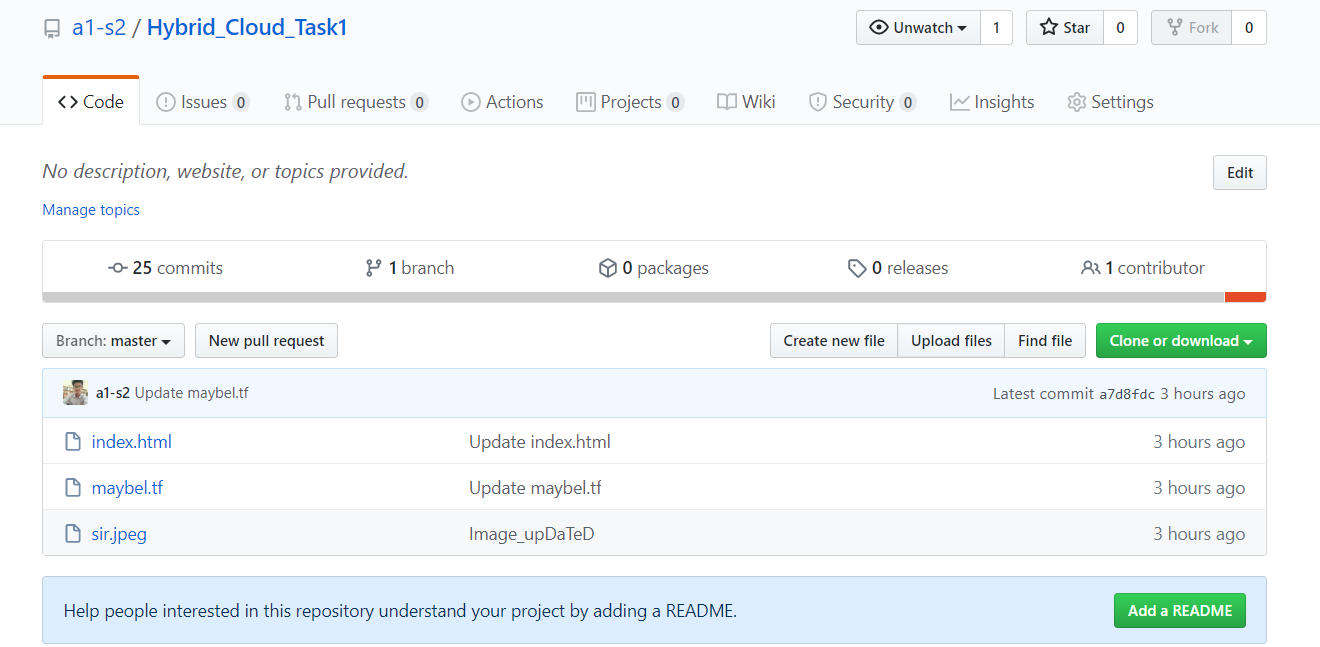


Now terraform init for installing plugin’s .



Terraform confi fine executing anf will give us output





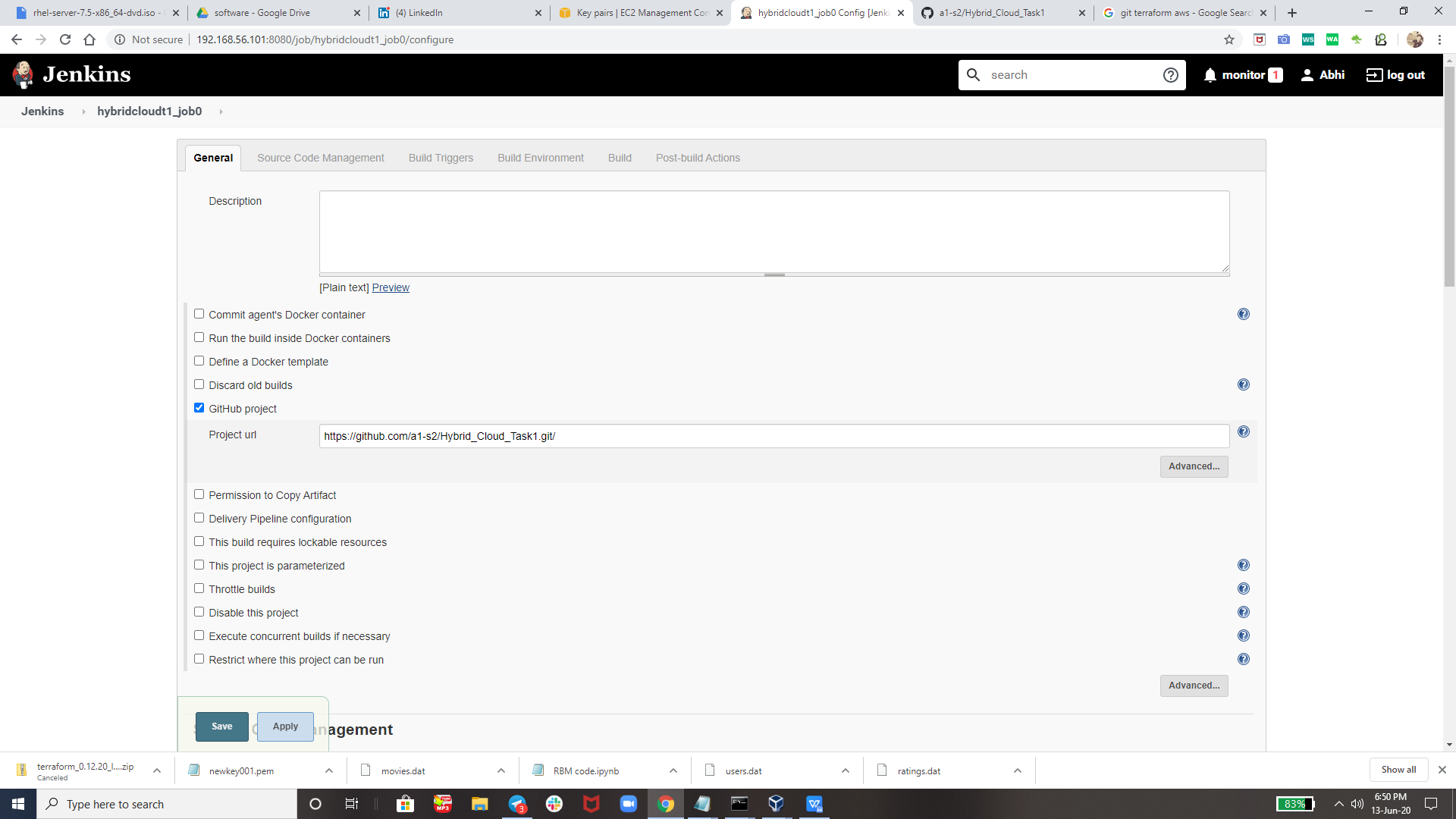
As Developer commited the code from git bash and its updated to github so jenkins job will trigger automatically

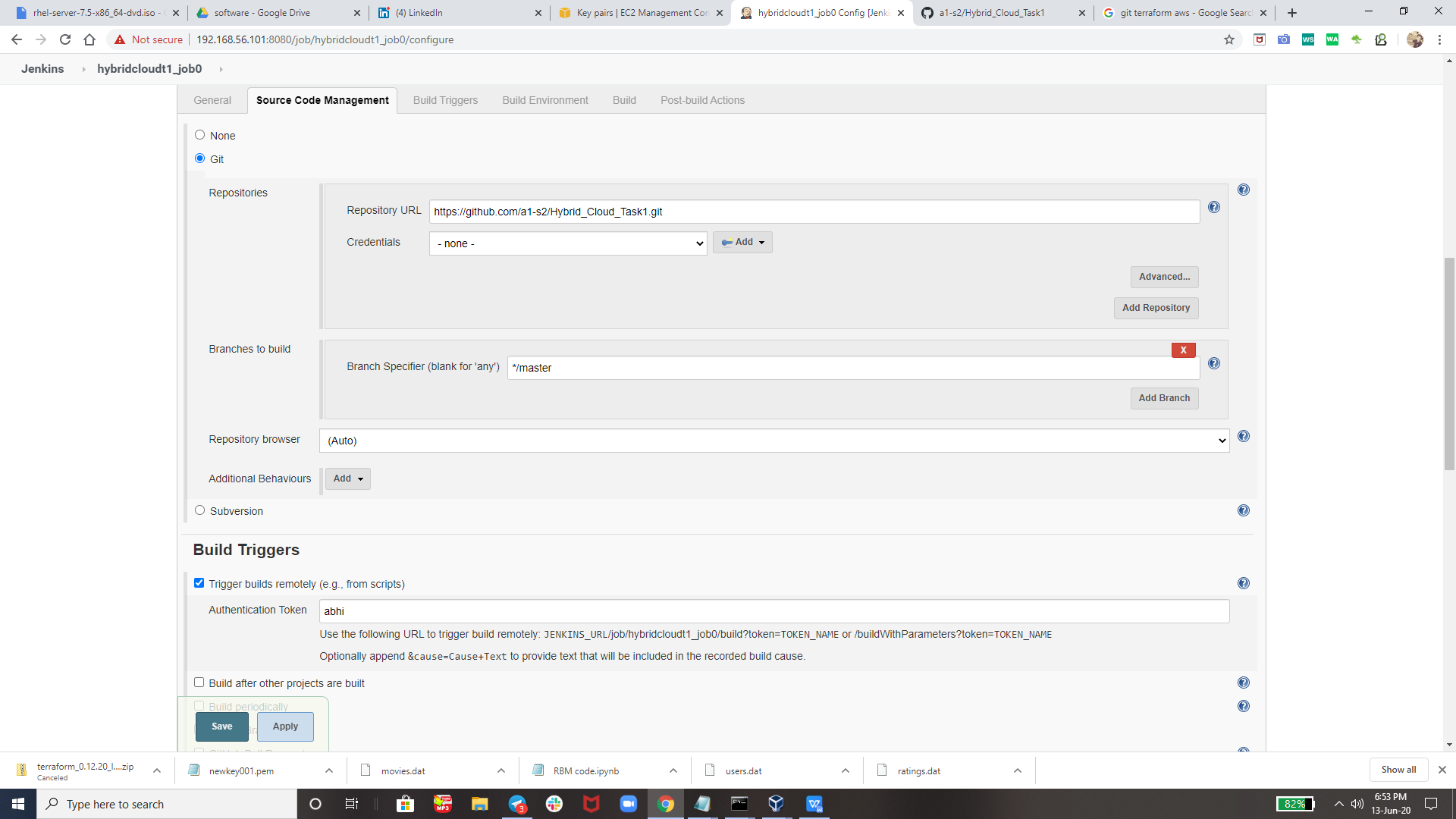
EveryThing that can be done from here is manually so I am using jenkins to automate this

Created One Job as hybridcloud\_job it will download all the data from my github repos.

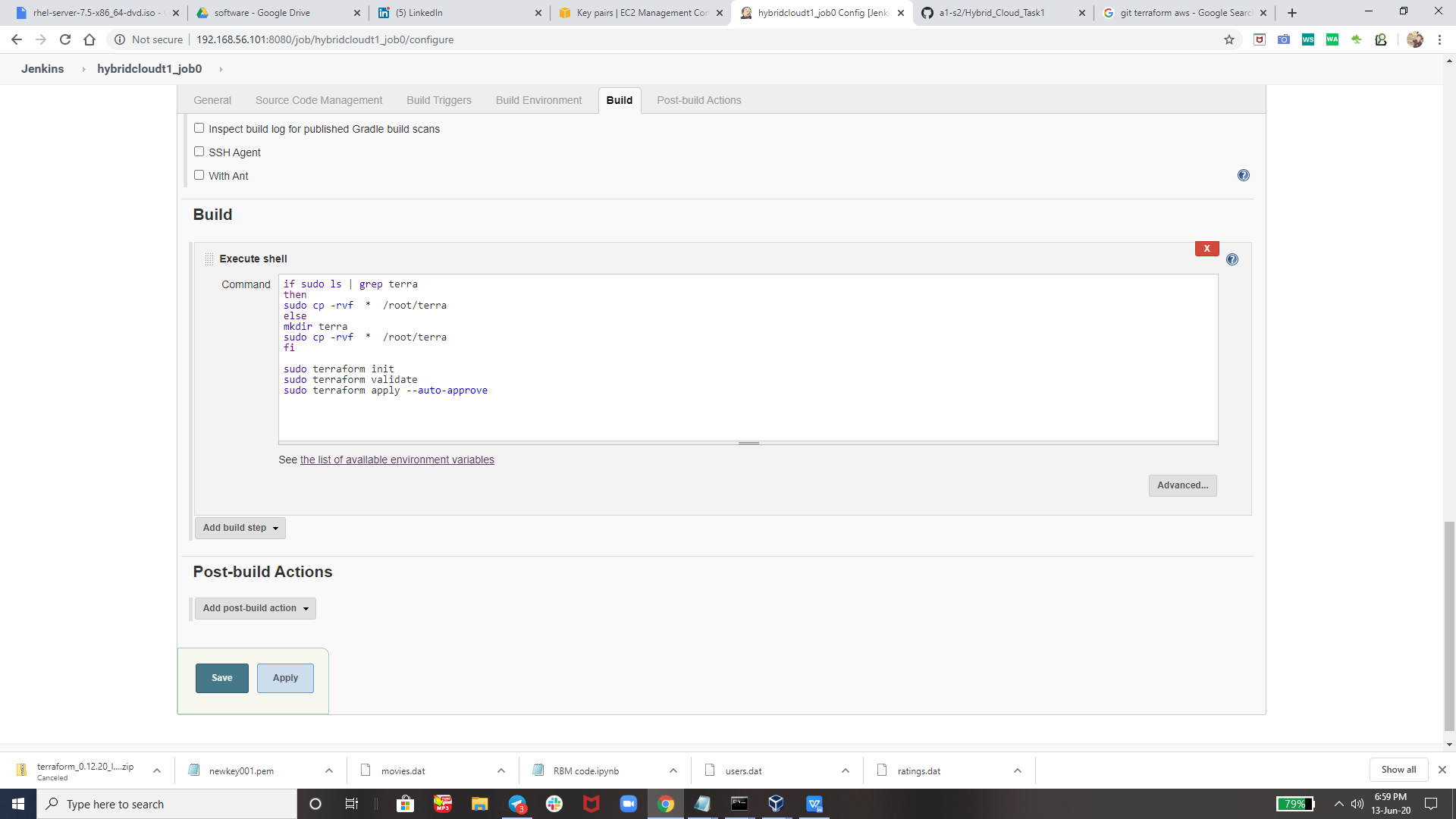
And will copy where we want then it will run all required commands that we have run above automatically and this job will automatically run whenever developer commit from git bash

Because I used git trigger concept here..

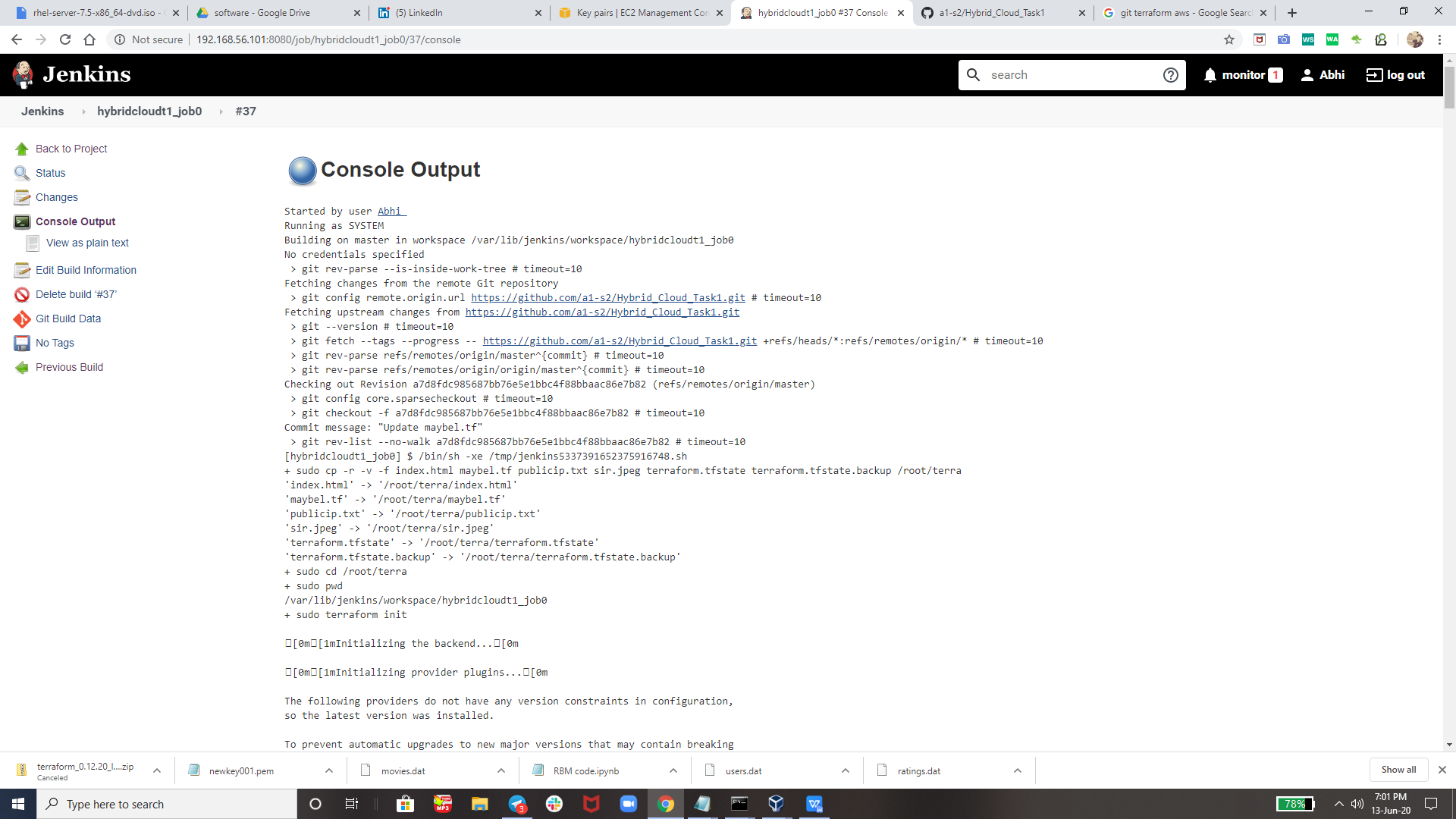


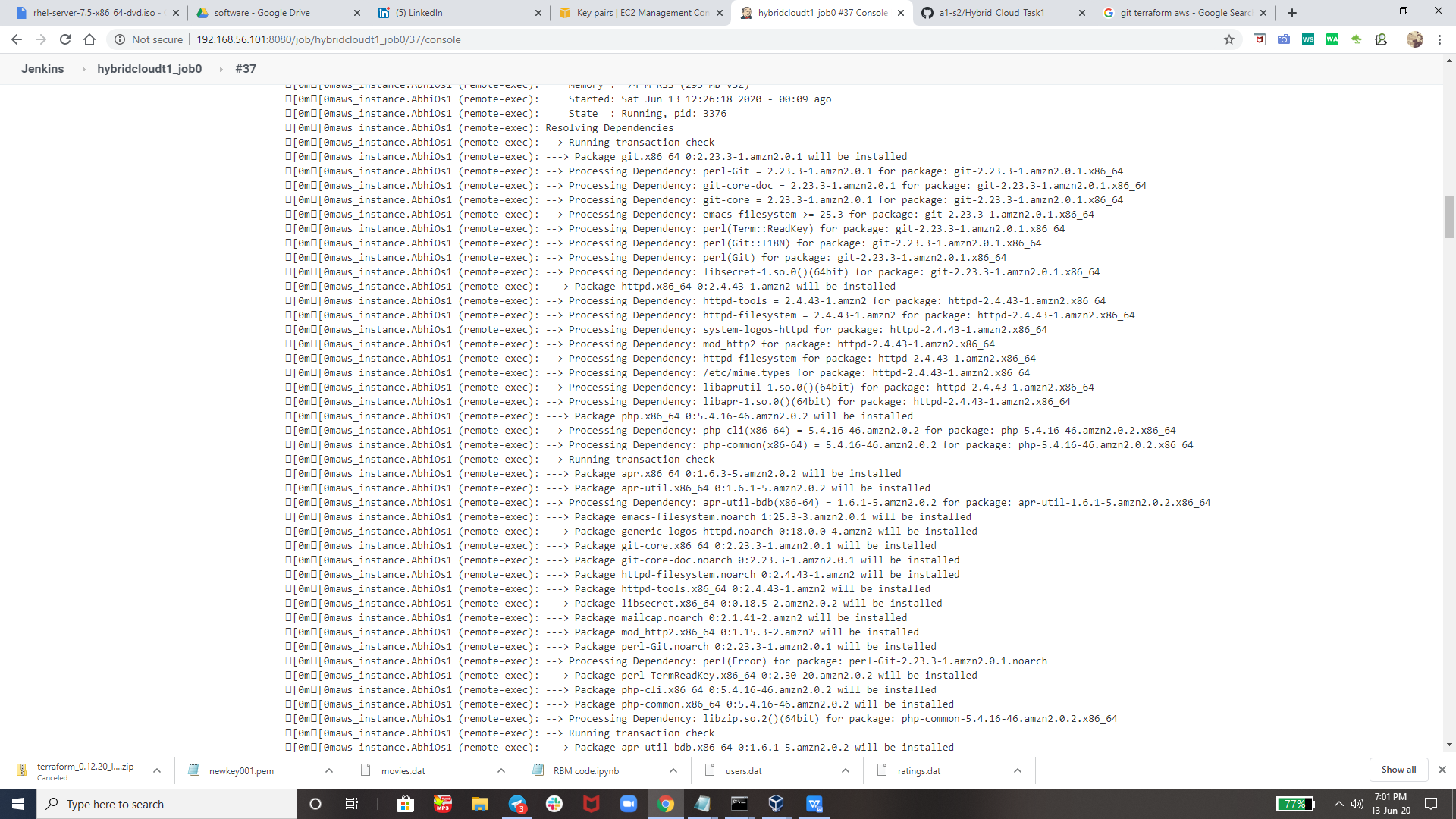


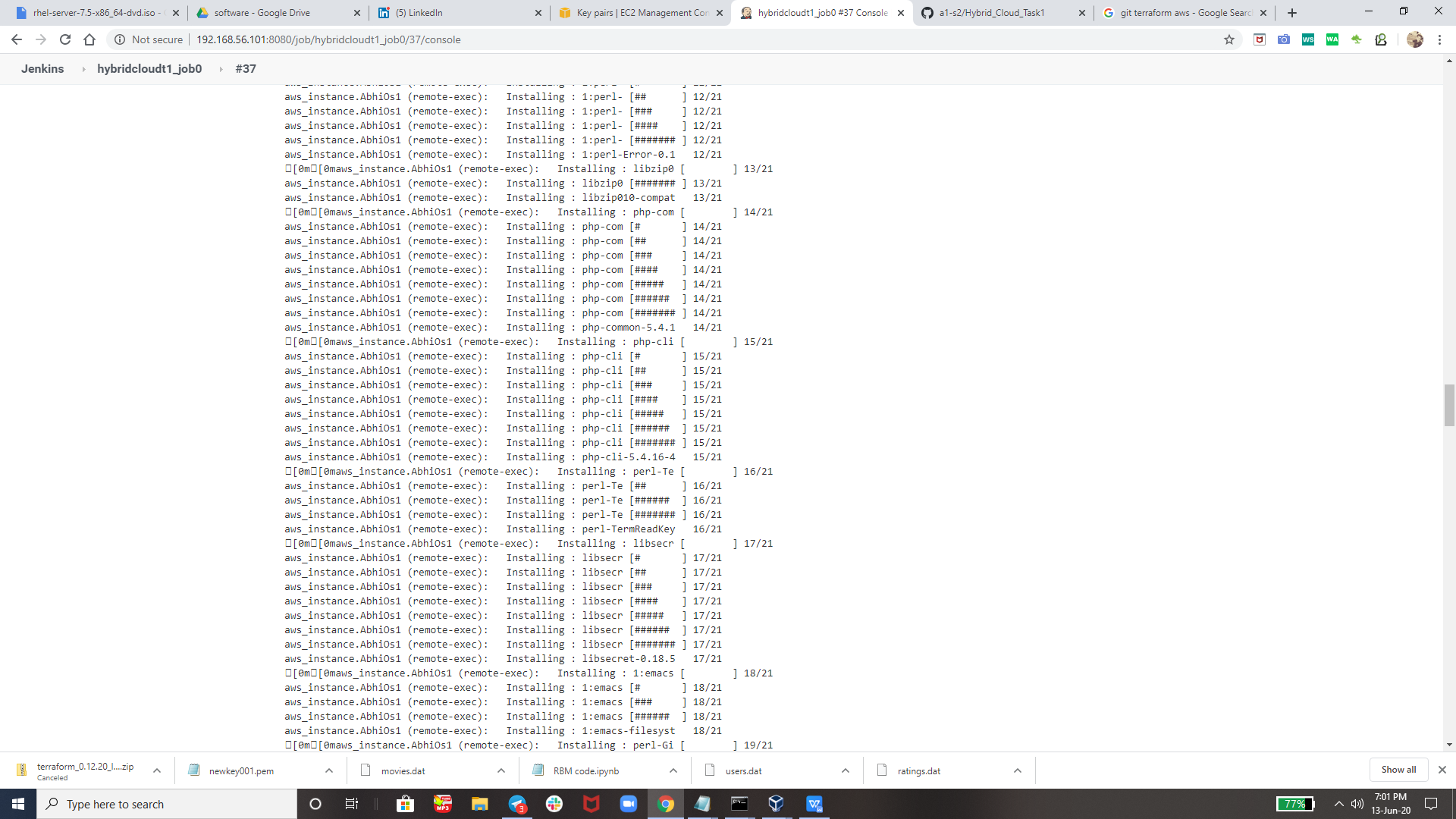
This code will search for the folder we want if that not found then it will created new one and then then copy all the data their & do terraform apply also as u can see in below image .

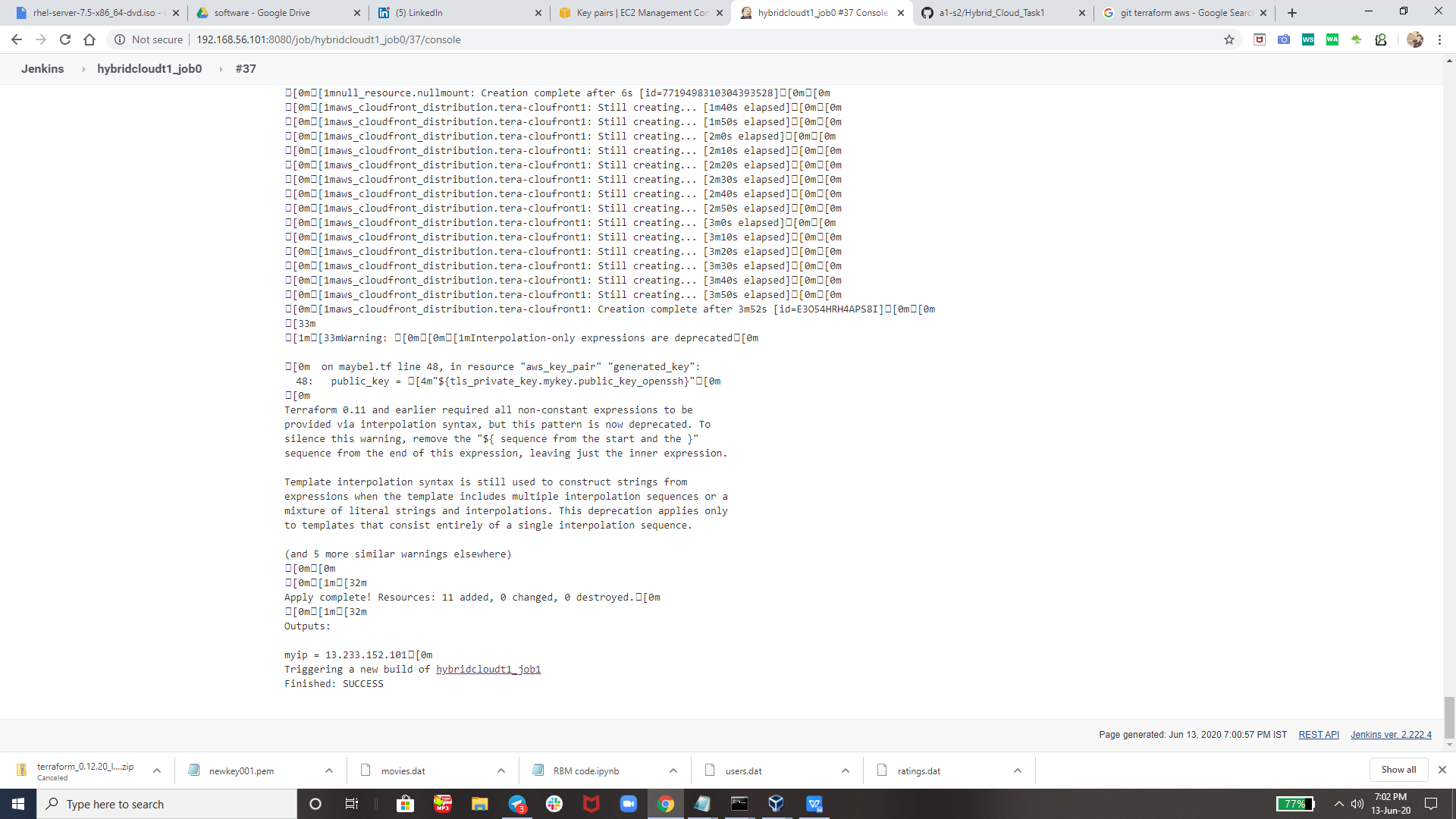


Now Output of Job is Here ..







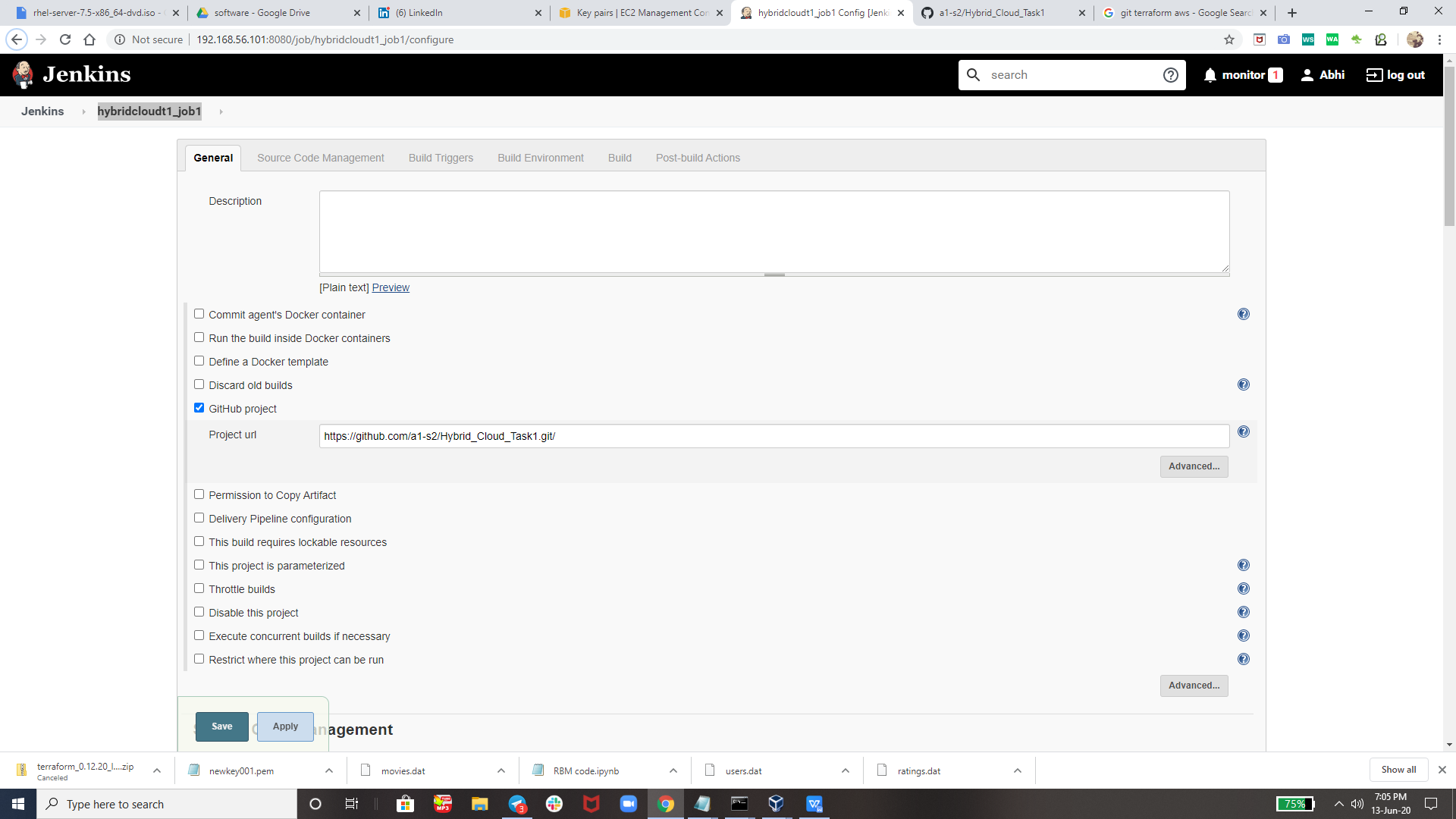


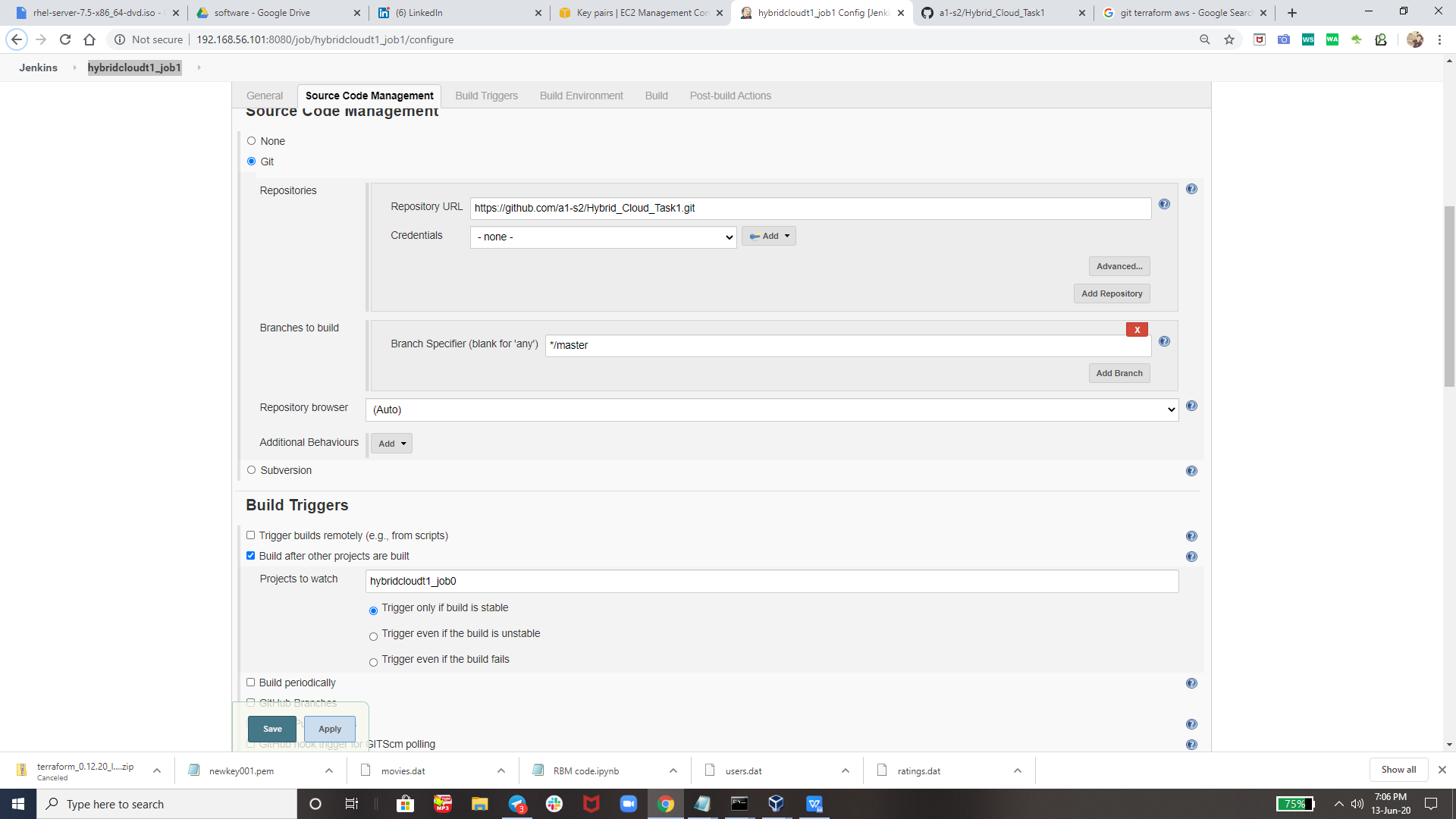
And then After Succesfully Running Job1 it will trigger job 2 so

I have created one more job as hybridcloudt1\_job1 it will run only if job is run succesfully without any

And because our job1 is run succesfully without any error so it will run job2 I.e. hybridcloudt1\_job1

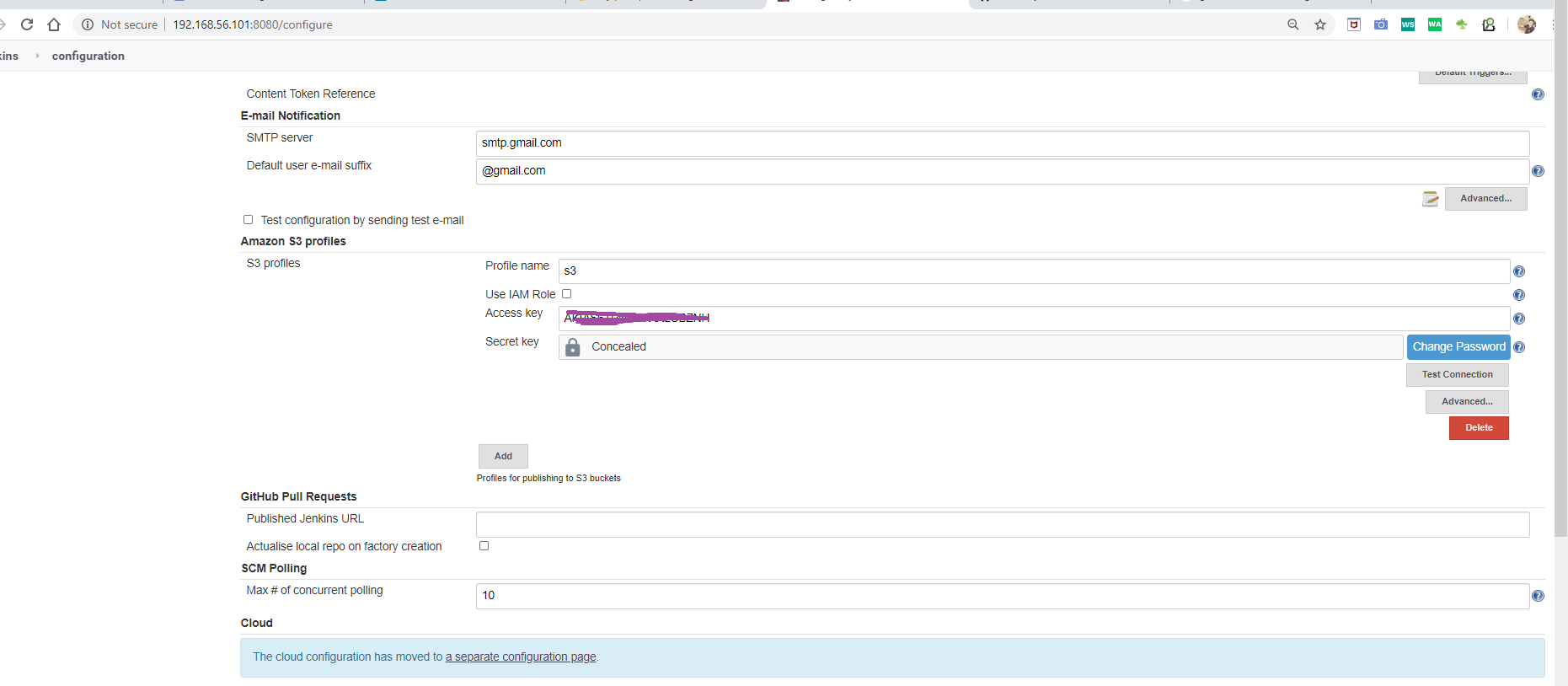
Here is my Job to Configuration



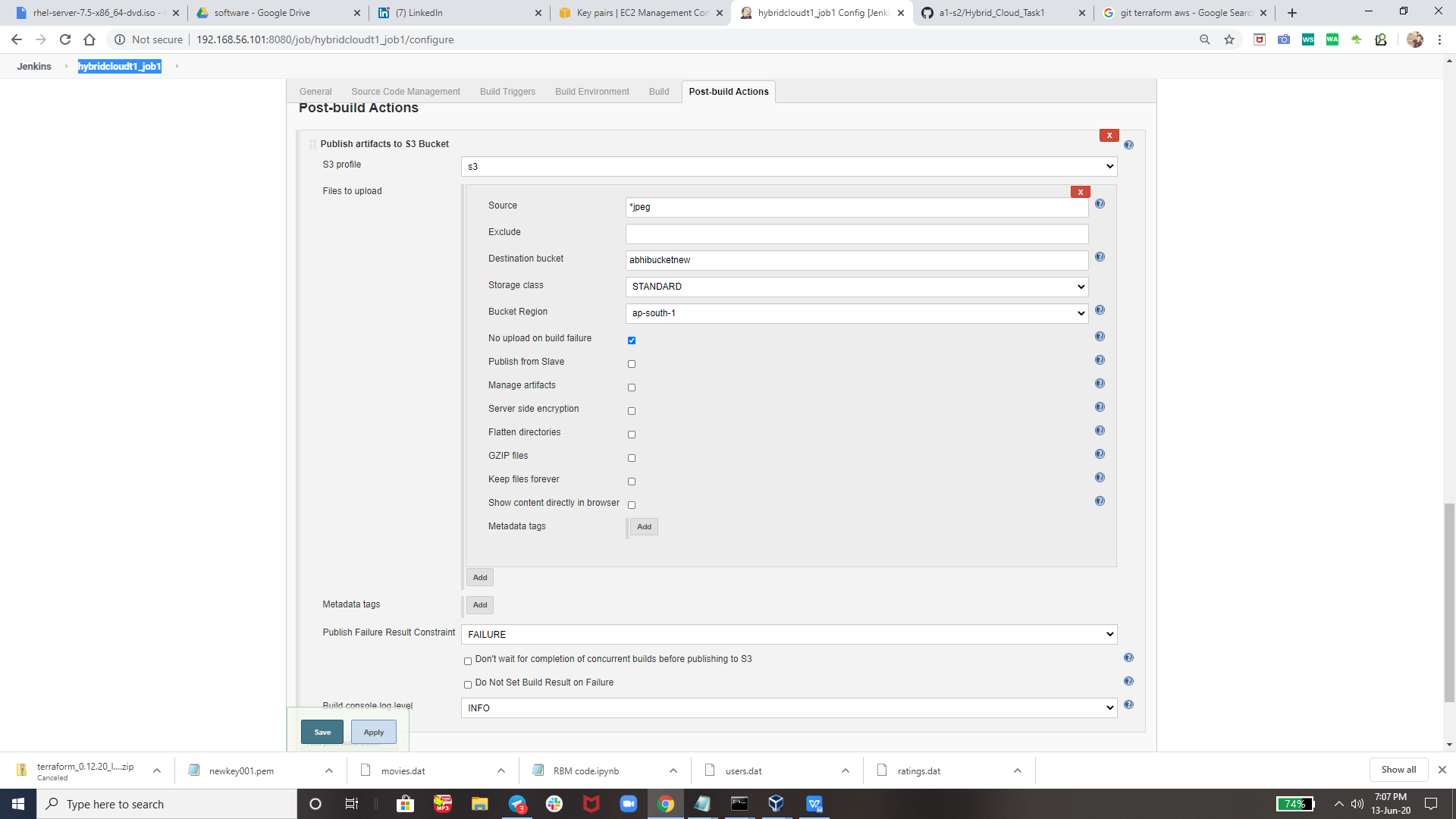


BuT Before NexT Steps we need tO Configurer SomeThing I.e.

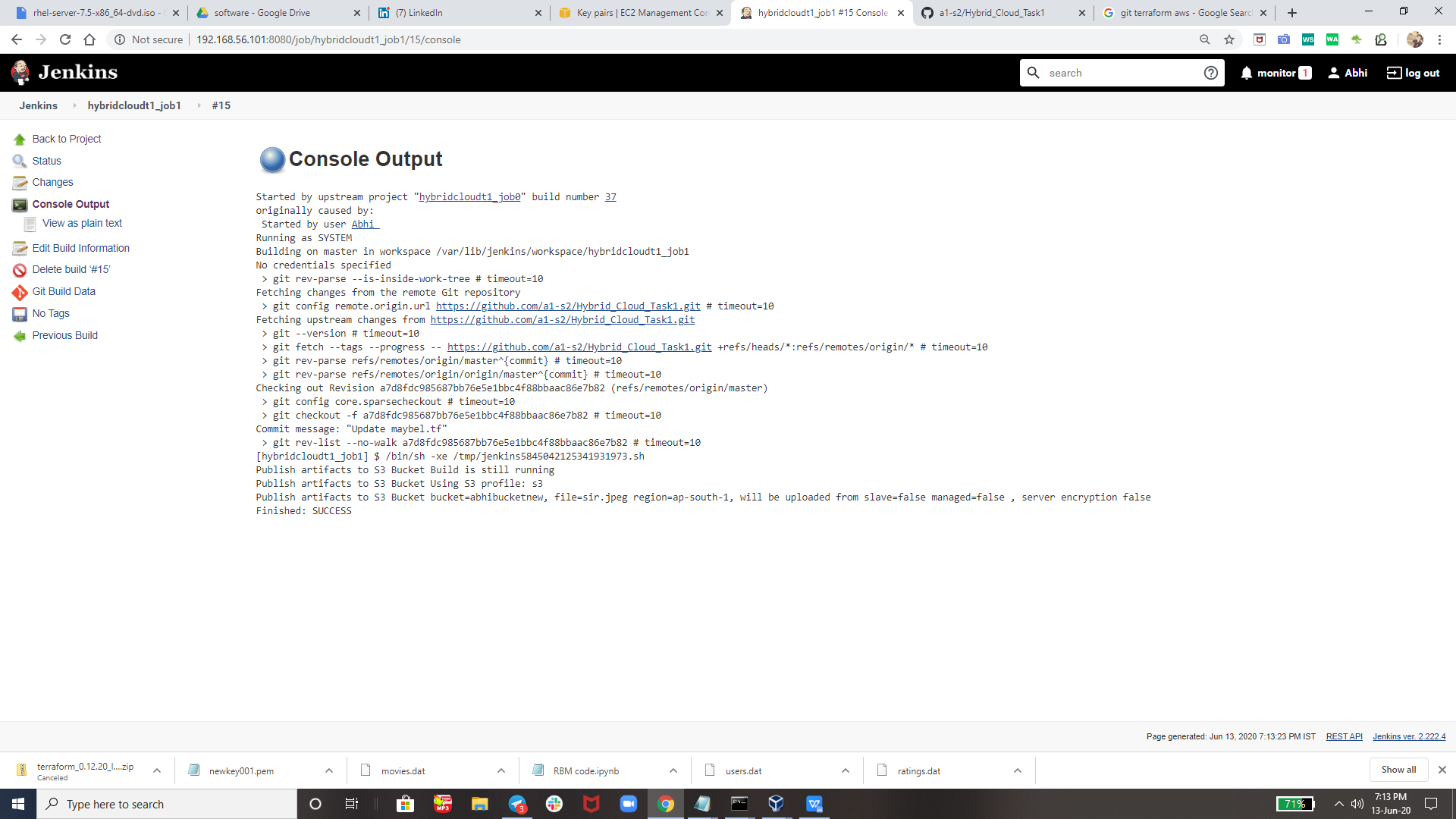




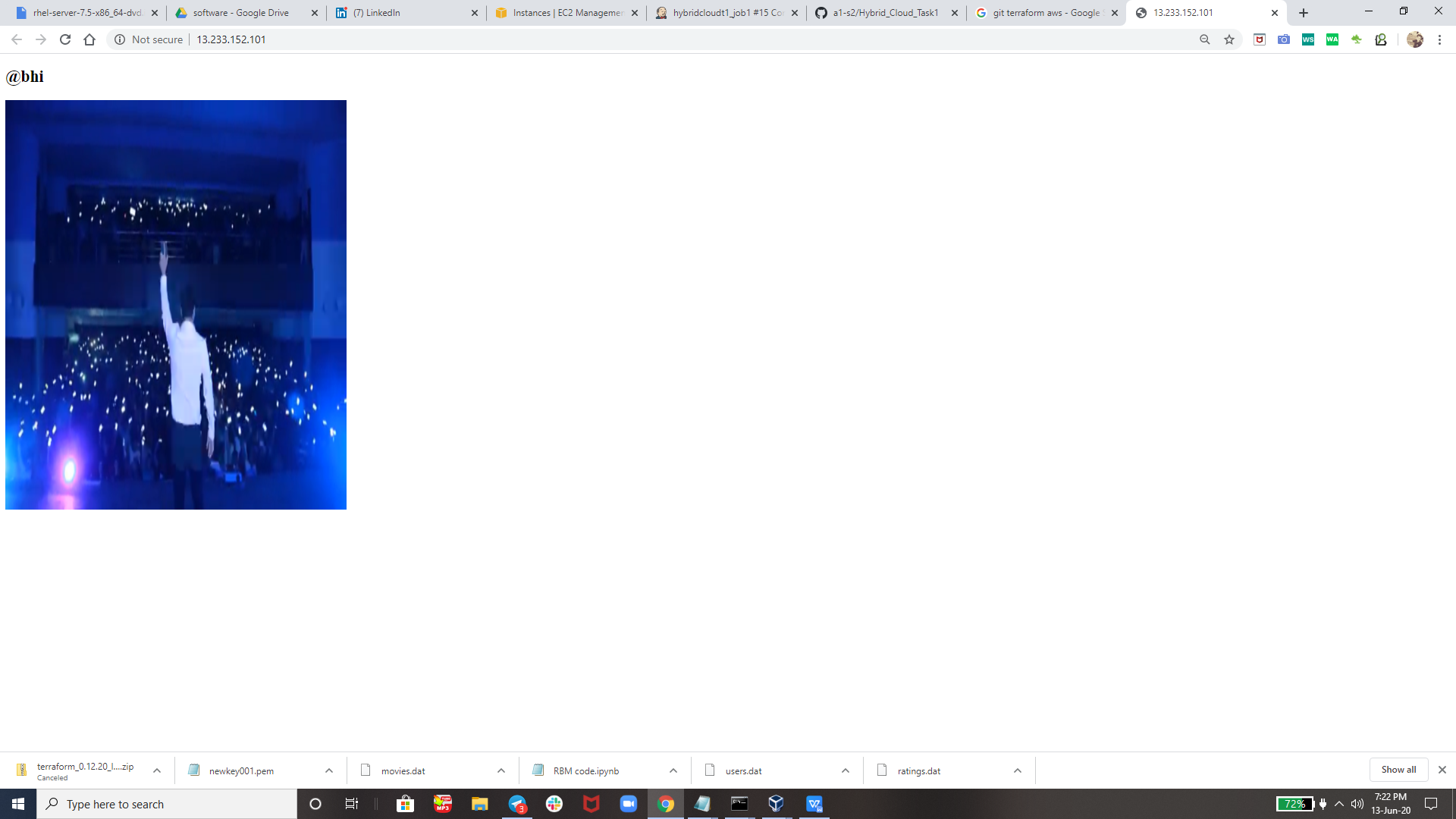
And this is for sending our image to created bucket



Here is OuTPuT of Job2 I.e. hybridcloudt1\_job1



Now I am going to show u from their output on Aws ..



And After This I had Destroyed everything in one click using command that is **terraform destsroy --auto-approve &** remember its not completely destroyed whenever developer commit from git bash in future this process will happen again and again whenever required ..

This Kind of ProjecT can be used in industry to Automate the Things.

All The Code, Files, Images I am uploading on GitHub

**All Thanks tO my Mentor The World Record Holder Mr. Vimal Daga Sir**

**Thank you vimal sir for teaching us such an great tools technology like RHEL 8, Aws, Terraform , git ,Jenkins and many more. So that today i am capable of creating such an AuTomaTeD ProJecT.**

**Thank you EveryOne For Reading !**