Lab3:

1- Don’t work on the default Workspace Create a new workspace called dev

2- Using custom not public modules to implement the below diagram

3- remote bucket For statefile

4- Use remote provisioner to install apache or proxy in machines then use local-exec to print all the ips to a file called all-ips.txt with format

public-ip1 1.1.1.1

public-ip2 2.2.2.2

5- Use the datasource to get the image id for ec2

6- The first Loadbalancer is public , and the other one that will send thee traffic to the private machines will be private

7- Github URl with the below inn it:

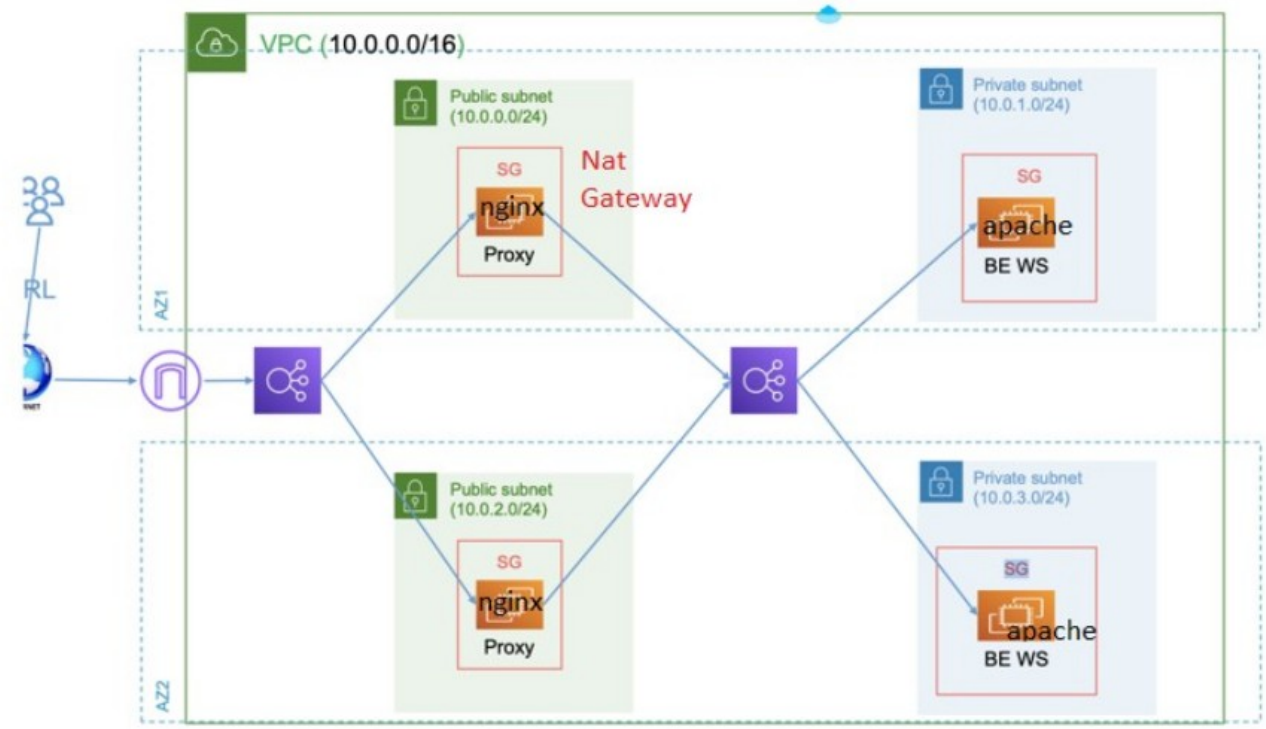
a. Code

b. Screenshot from creating and working on workspace dev

c. Screenshot from the configuration of the proxy

d. Screenshot from the public dns of the load balancer when you send a traffic to it from a browser and it returns the content of the private ec2s

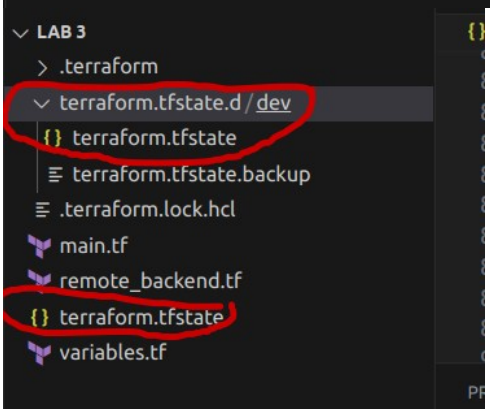
e. Screenshot from the s3 that contain the state file



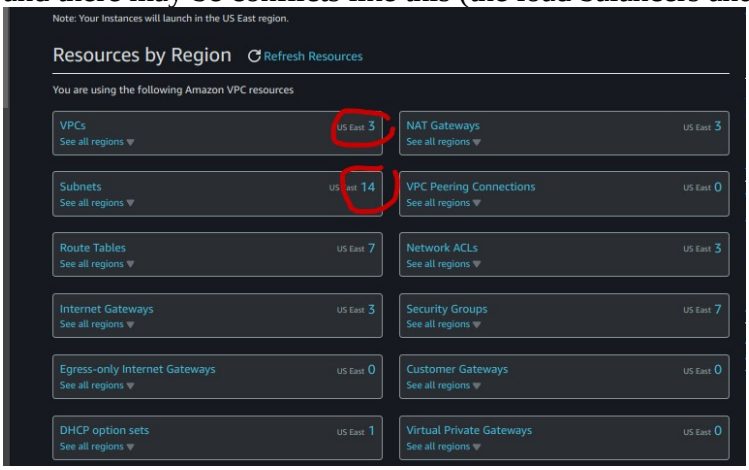
1.

```
spot@spot-pc:/media/spot/My-Data/Downloads/DevOps/20. terraform/day 3/lab 3$ terraform workspace list
* default

spot@spot-pc:/media/spot/My-Data/Downloads/DevOps/20. terraform/day 3/lab 3$ terraform workspace new dev
Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
```

```
spot@spot-pc:/media/spot/My-Data/Downloads/DevOps/20. terraform/day 3/lab 3$ terraform workspace select dev
Switched to workspace "dev".
spot@spot-pc:/media/spot/My-Data/Downloads/DevOps/20. terraform/day 3/lab 3$ terraform workspace list
  default
* dev

spot@spot-pc:/media/spot/My-Data/Downloads/DevOps/20. terraform/day 3/lab 3$
```

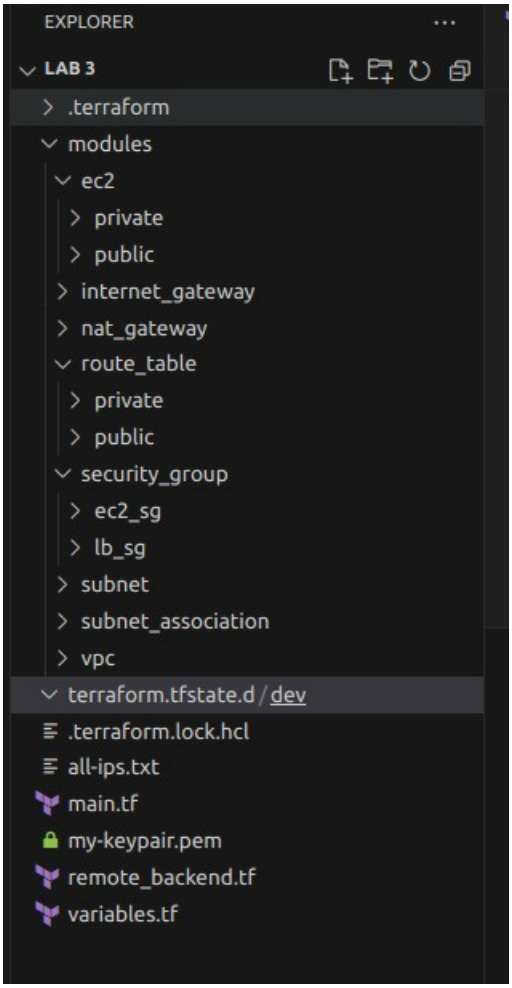


if run terraform apply again from the default workspace it will create all the infrastructure again with different ips only and there may be conflicts like this (the load balancers and the the target groups)



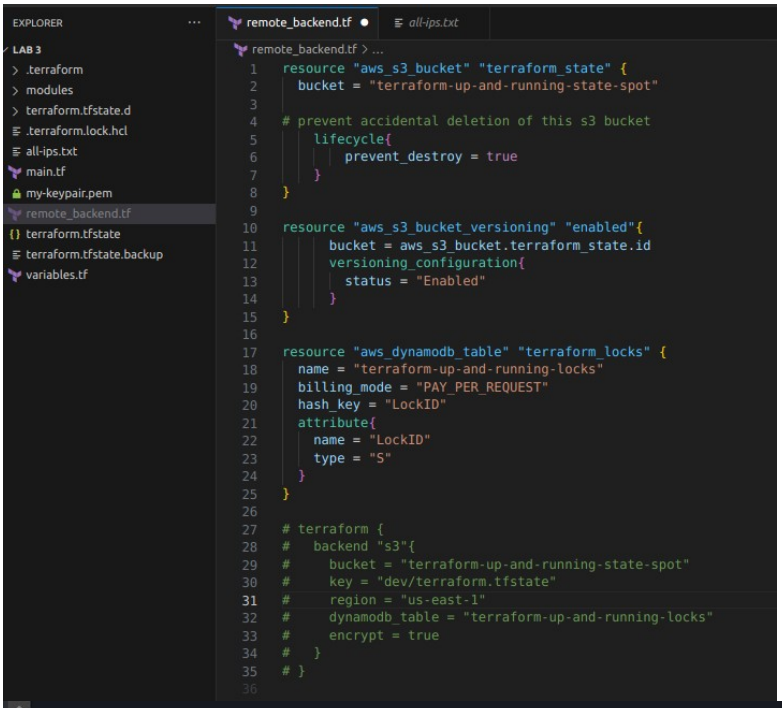

```
module.private_subnet1_association.aws_route_table_association.subnet_association: Creation complete after 1s [id=rtbasso
c-0a4500d5d4b41ea55]
module.private_subnet2_association.aws_route_table_association.subnet_association: Creation complete after 1s [id=rtbasso
c-0fe9c1127d93f17c0]

Error: creating ELBv2 application Load Balancer (public-lb): DuplicateLoadBalancerName: A load balancer with the same n
ame 'public-lb' exists, but with different settings
       status code: 400, request id: 3653f0c5-ab75-46b3-b32e-dd9f5d0c7e1a

  with aws_lb.public_lb,
  on main.tf line 257, in resource "aws_lb" "public_lb":
 257: resource "aws_lb" "public_lb" {


Error: ELBv2 Target Group (public-target-group) already exists

  with aws_lb_target_group.public_target_group,
  on main.tf line 265, in resource "aws_lb_target_group" "public_target_group":
 265: resource "aws_lb_target_group" "public_target_group" {


Error: creating ELBv2 application Load Balancer (private-lb): DuplicateLoadBalancerName: A load balancer with the same
```

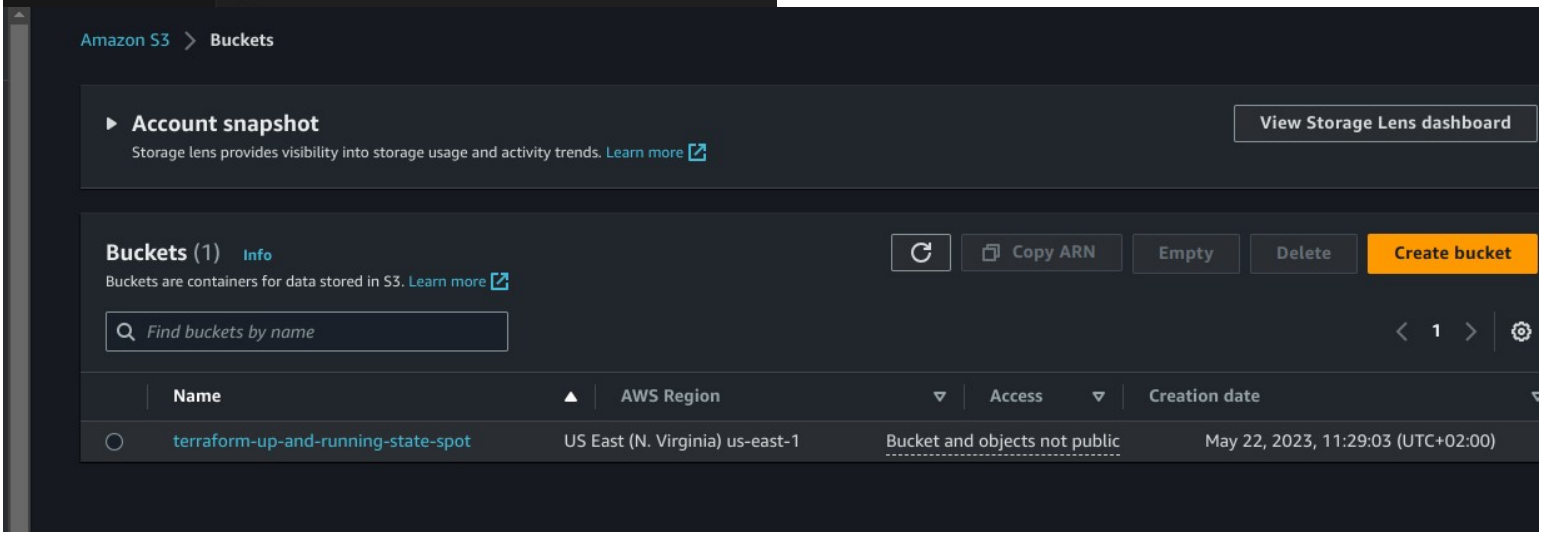
2.

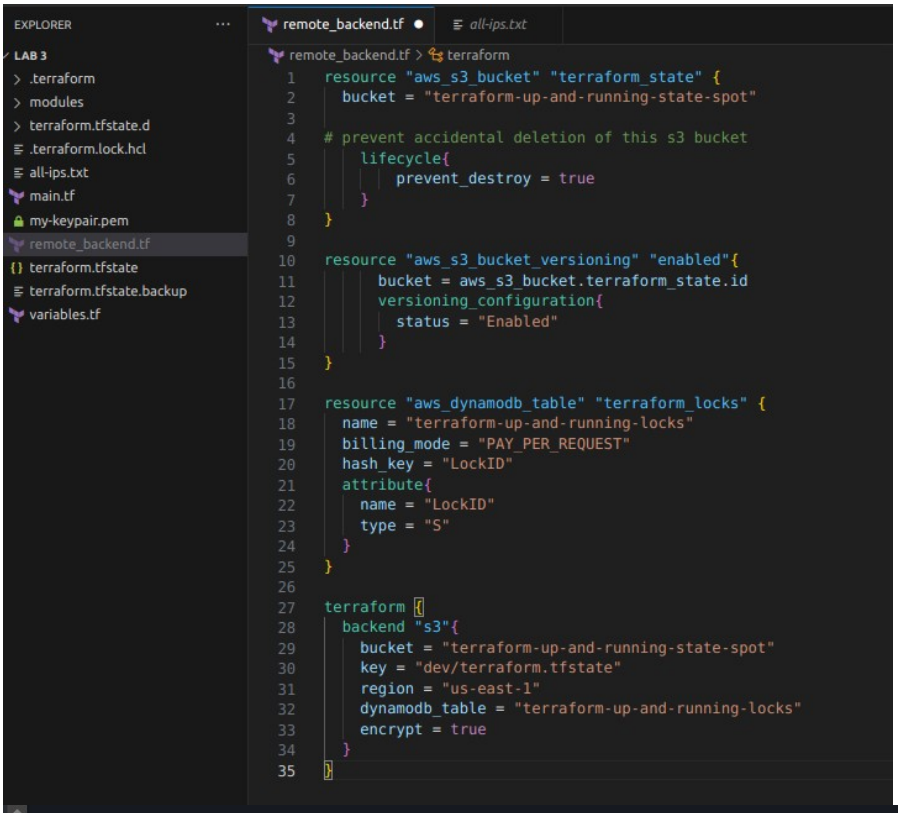

EXPLORER
LAB 3
.terraform
modules
ec2
private
public
internet_gateway
nat_gateway
route_table
private
public
security_group
ec2_sg
lb_sg
subnet
subnet_association
vpc
terraform.tfstate.d / dev
.terraform.lock.hcl
all-ips.txt
main.tf
my-keypair.pem
remote_backend.tf
variables.tf

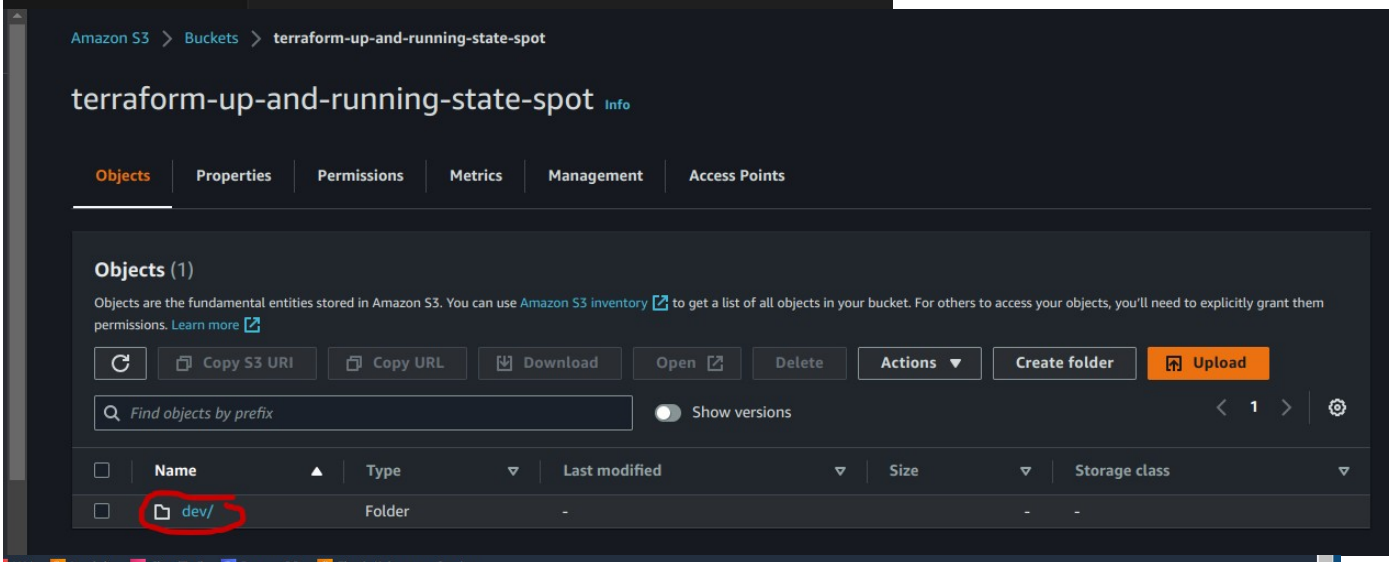
3.

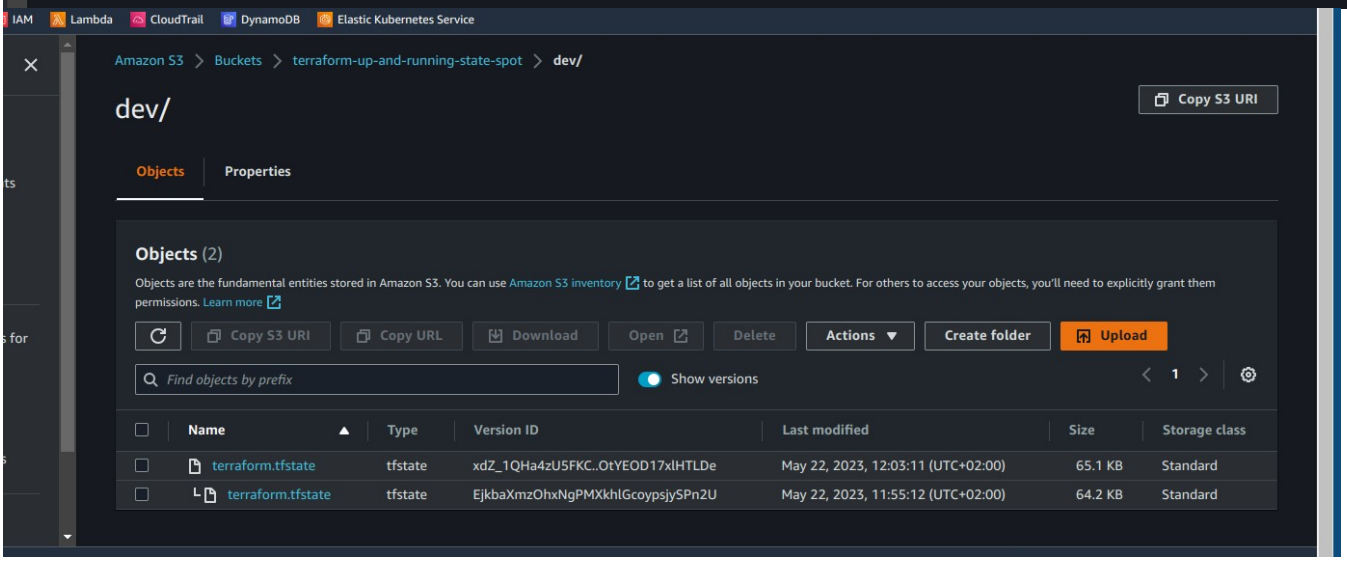
the s3 bucket must be created first (note: everthing in the backend file should be hard coded)

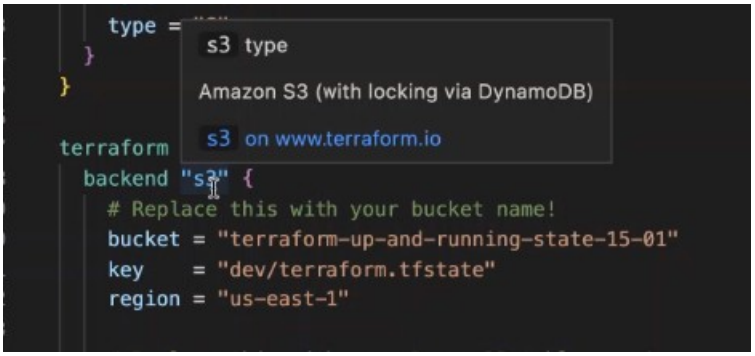




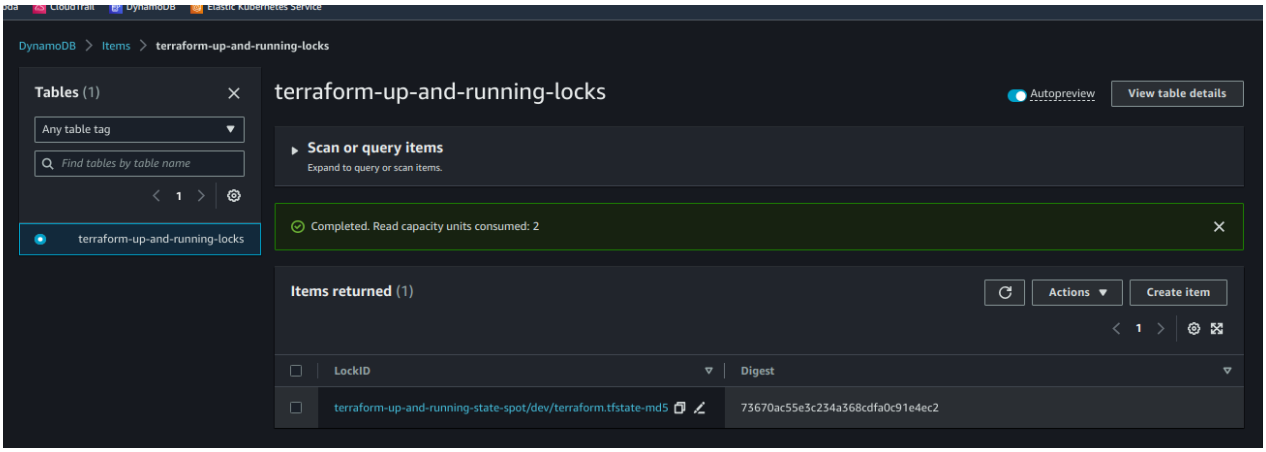

EXPLORER
remote_backend.tf
all-ips.txt
LAB 3
.terraform
modules
terraform.tfstate.d
.terraform.lock.hcl
all-ips.txt
main.tf
my-keypair.pem
remote_backend.tf
terraform.tfstate
terraform.tfstate.backup
variables.tf
remote_backend.tf > terraform
resource "aws_s3_bucket" "terraform_state" {
bucket = "terraform-up-and-running-state-spot"
# prevent accidental deletion of this s3 bucket
lifecycle{
prevent_destroy = true
}
}
resource "aws_s3_bucket_versioning" "enabled"{
bucket = aws_s3_bucket.terraform_state.id
versioning_configuration{
status = "Enabled"
}
}
resource "aws_dynamodb_table" "terraform_locks" {
name = "terraform-up-and-running-locks"
billing_mode = "PAY_PER_REQUEST"
hash_key = "LockID"
attribute{
name = "LockID"
type = "S"
}
}
terraform {
backend "s3"{
bucket = "terraform-up-and-running-state-spot"
key = "dev/terraform.tfstate"
region = "us-east-1"
dynamodb_table = "terraform-up-and-running-locks"
encrypt = true
}
}


Amazon S3 > Buckets > terraform-up-and-running-state-spot
terraform-up-and-running-state-spot Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
dev/
Folder
-

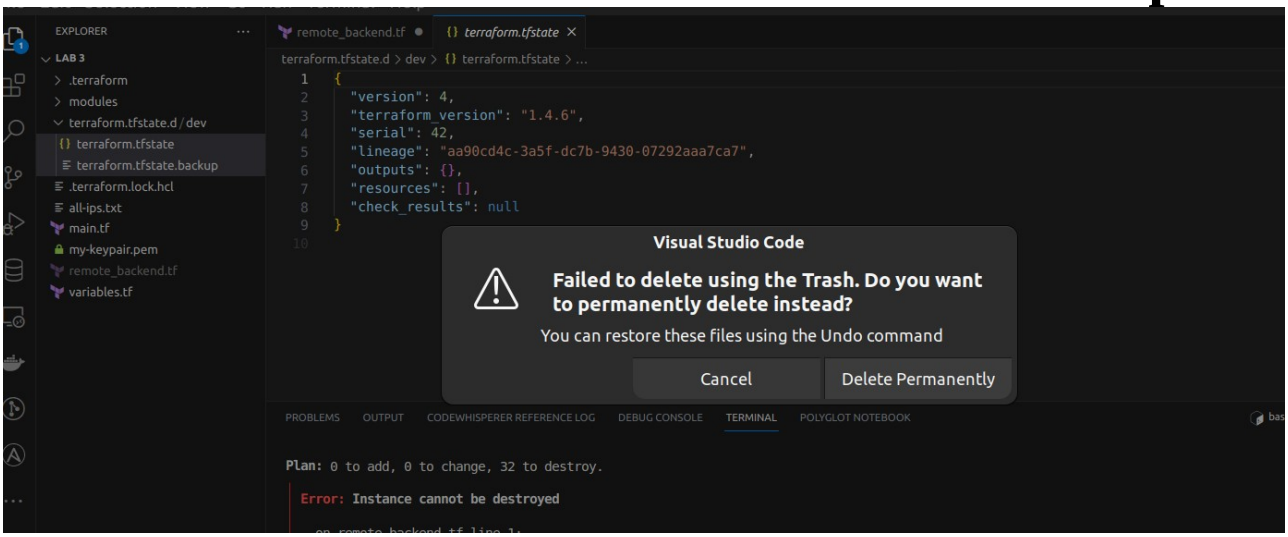

IAM
Lambda
CloudTrail
DynamoDB
Elastic Kubernetes Service
Amazon S3 > Buckets > terraform-up-and-running-state-spot > dev/
dev/
Copy S3 URI
Objects
Properties
Objects (2)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Show versions
Name
Type
Version ID
Last modified
Size
Storage class
terraform.tfstate
tfstate
xdZ_1QHa4zU5FKC..OtYEOD17xlHTLDe
May 22, 2023, 12:03:11 (UTC+02:00)
65.1 KB
Standard
terraform.tfstate
tfstate
EjkbaXmzOhxNgPMXkhlGcoypsjySPn2U
May 22, 2023, 11:55:12 (UTC+02:00)
64.2 KB
Standard



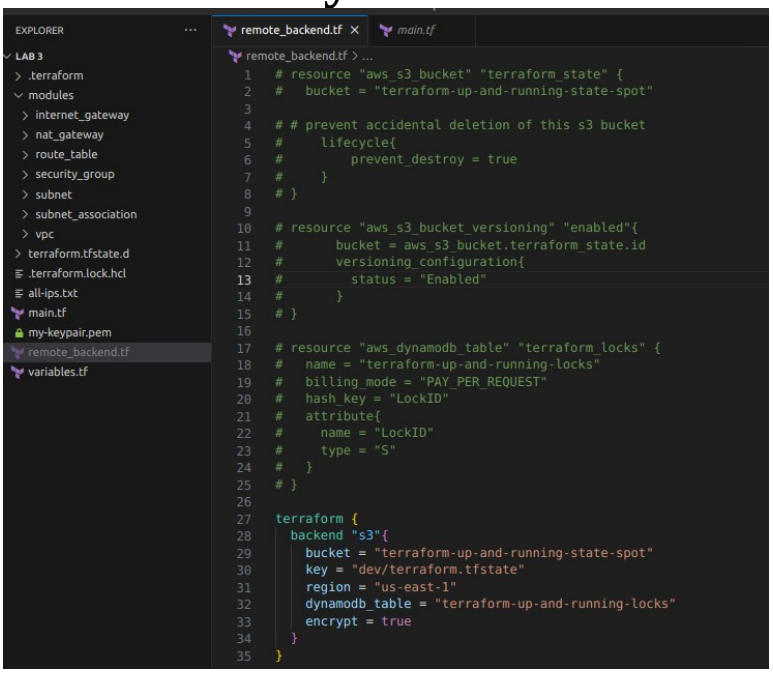
the lock record to prevent anyone from making changes on the same state file



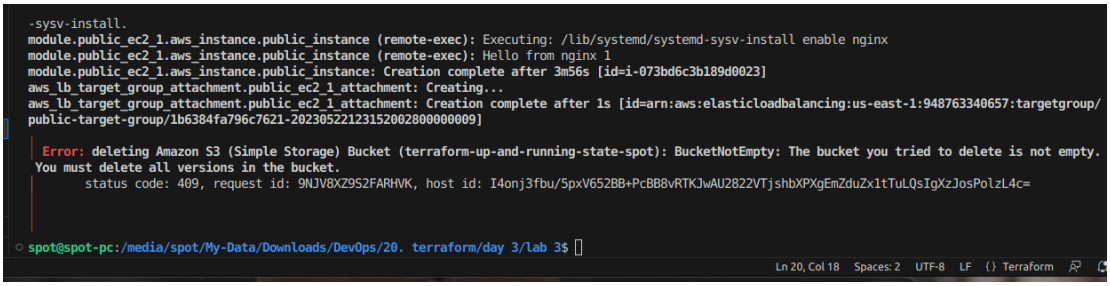
now removing all the terraform.tfstate local files (we don’t need them anymore and in every new apply he will fetch the latest state file from the buckets and works depends on it



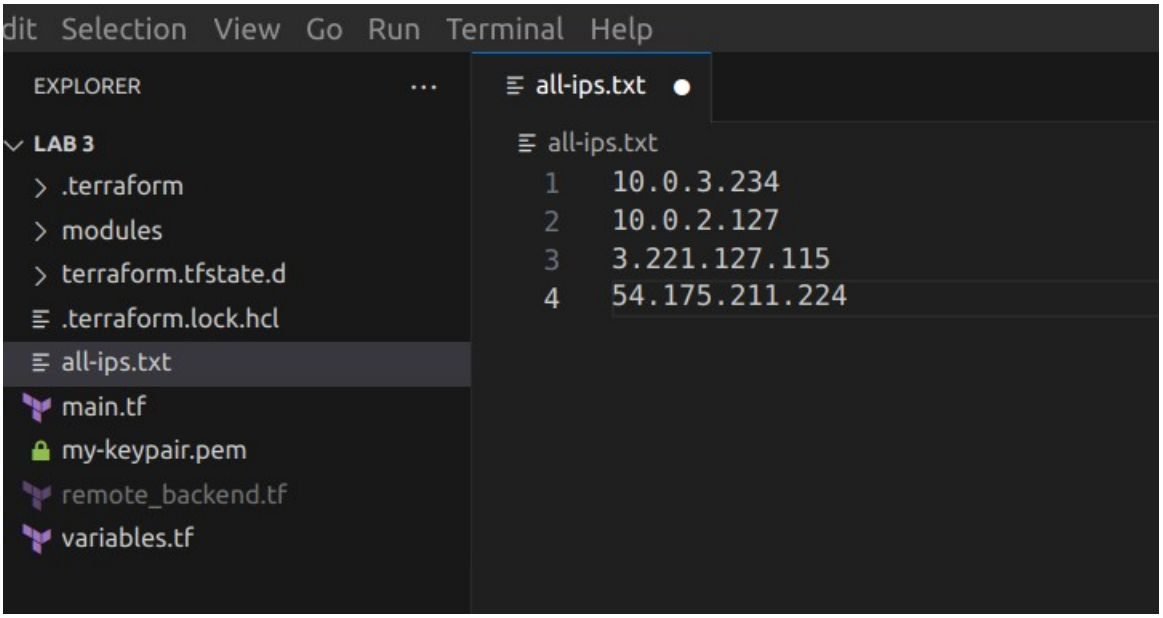
and to destroy all resources without getting error of you can’t delete the s3 you should comment the s3 block



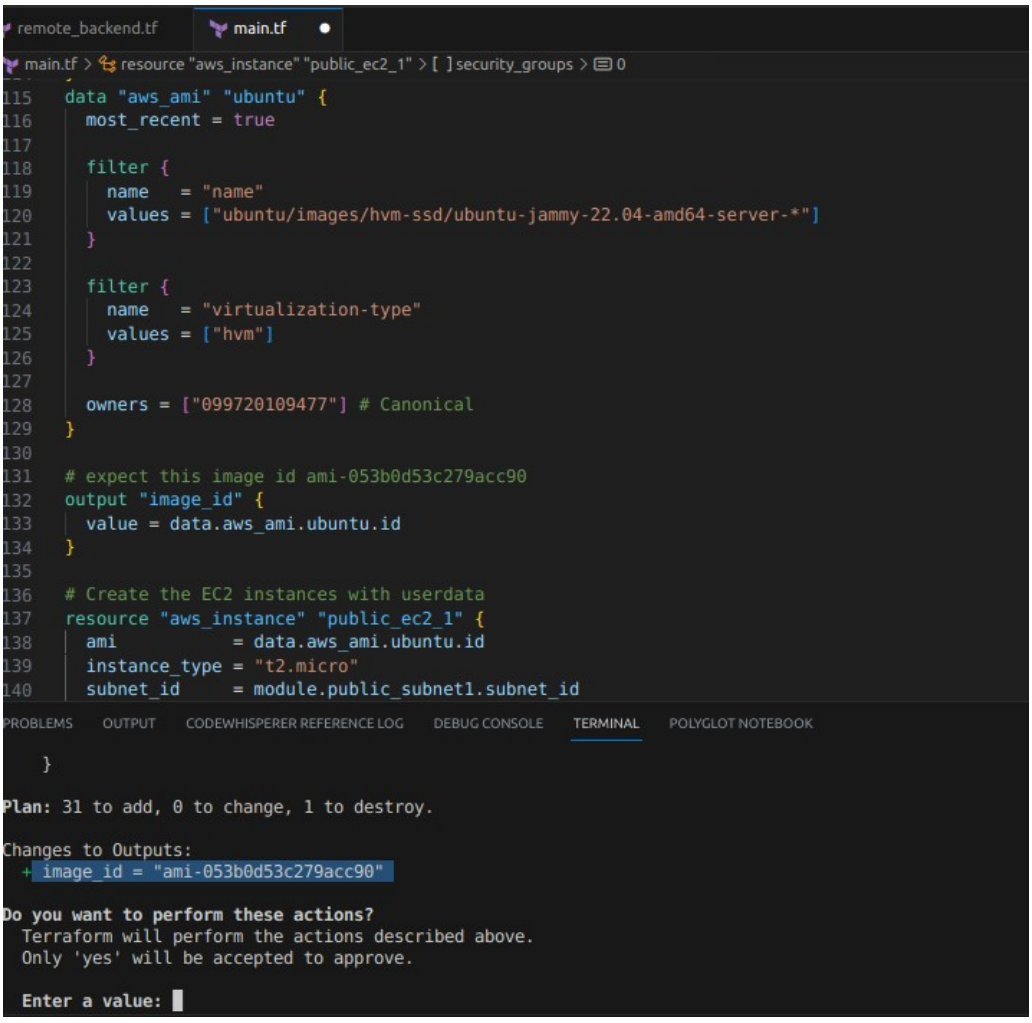
when I try to apply again everything is created and the s3 still not affected and this is what I want



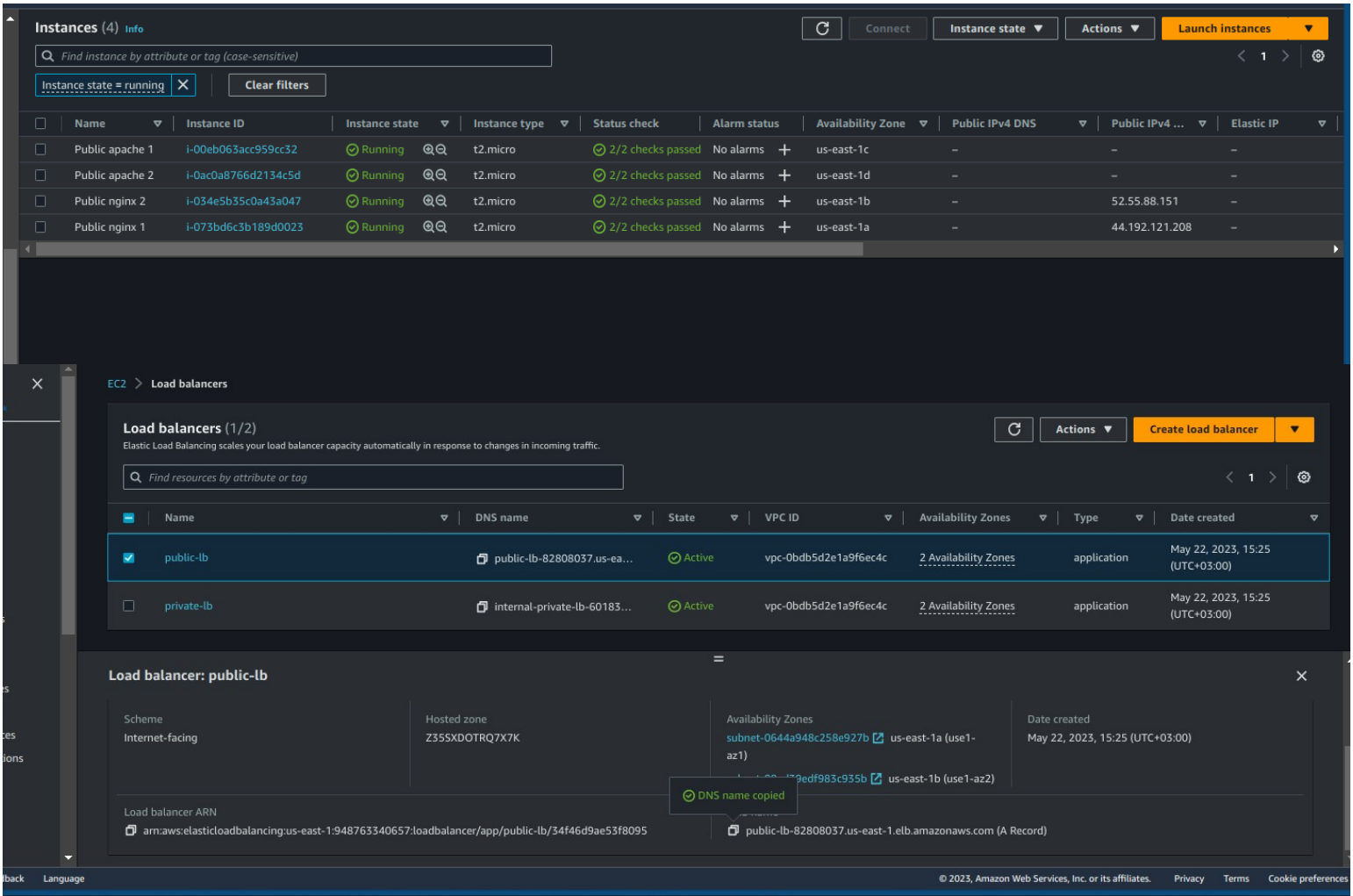
4.

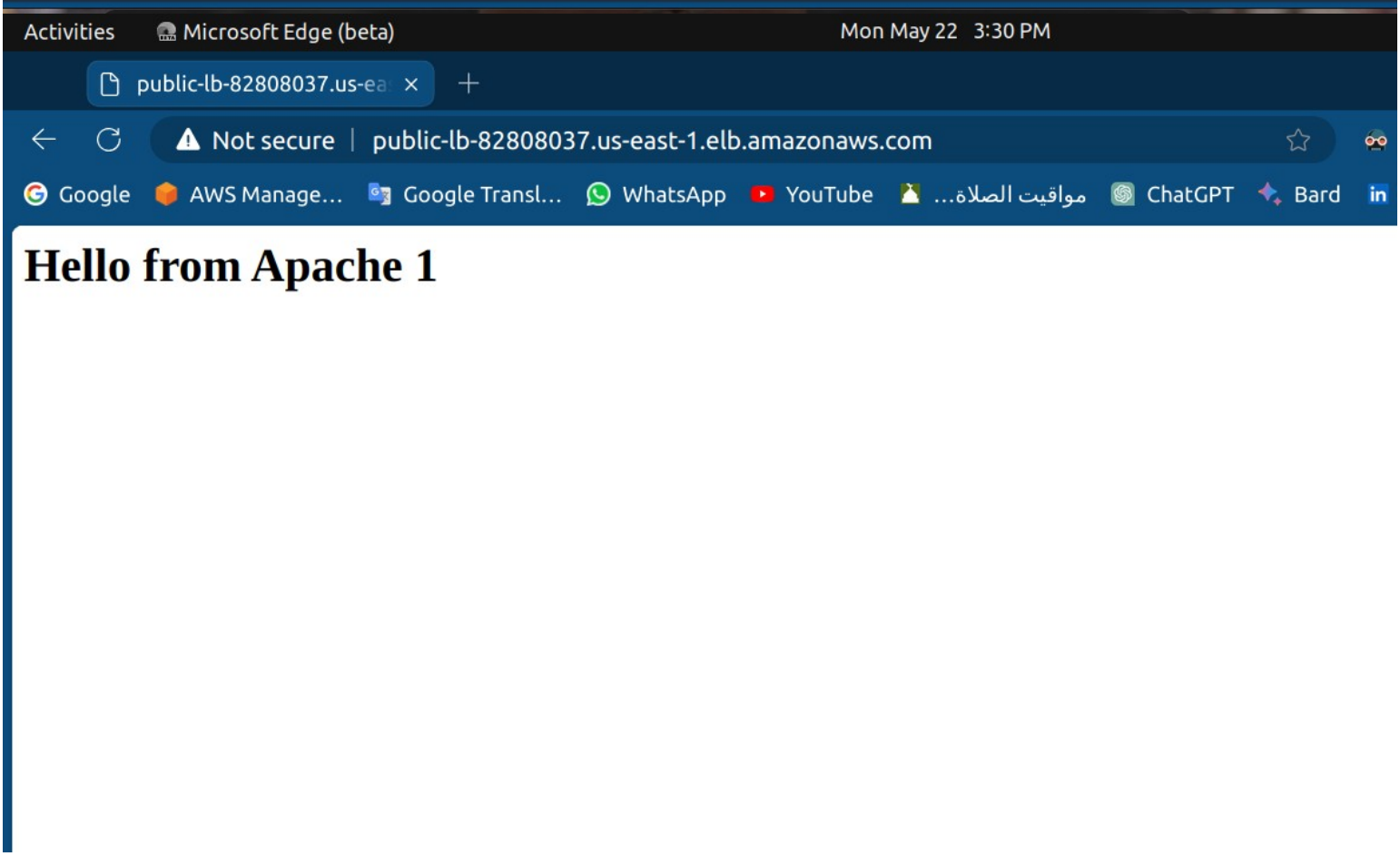

dit Selection View Go Run Terminal Help
EXPLORER
LAB 3
.terraform
modules
terraform.tfstate.d
.terraform.lock.hcl
all-ips.txt
main.tf
my-keypair.pem
remote_backend.tf
variables.tf
all-ips.txt
all-ips.txt
1 10.0.3.234
2 10.0.2.127
3 3.221.127.115
4 54.175.211.224

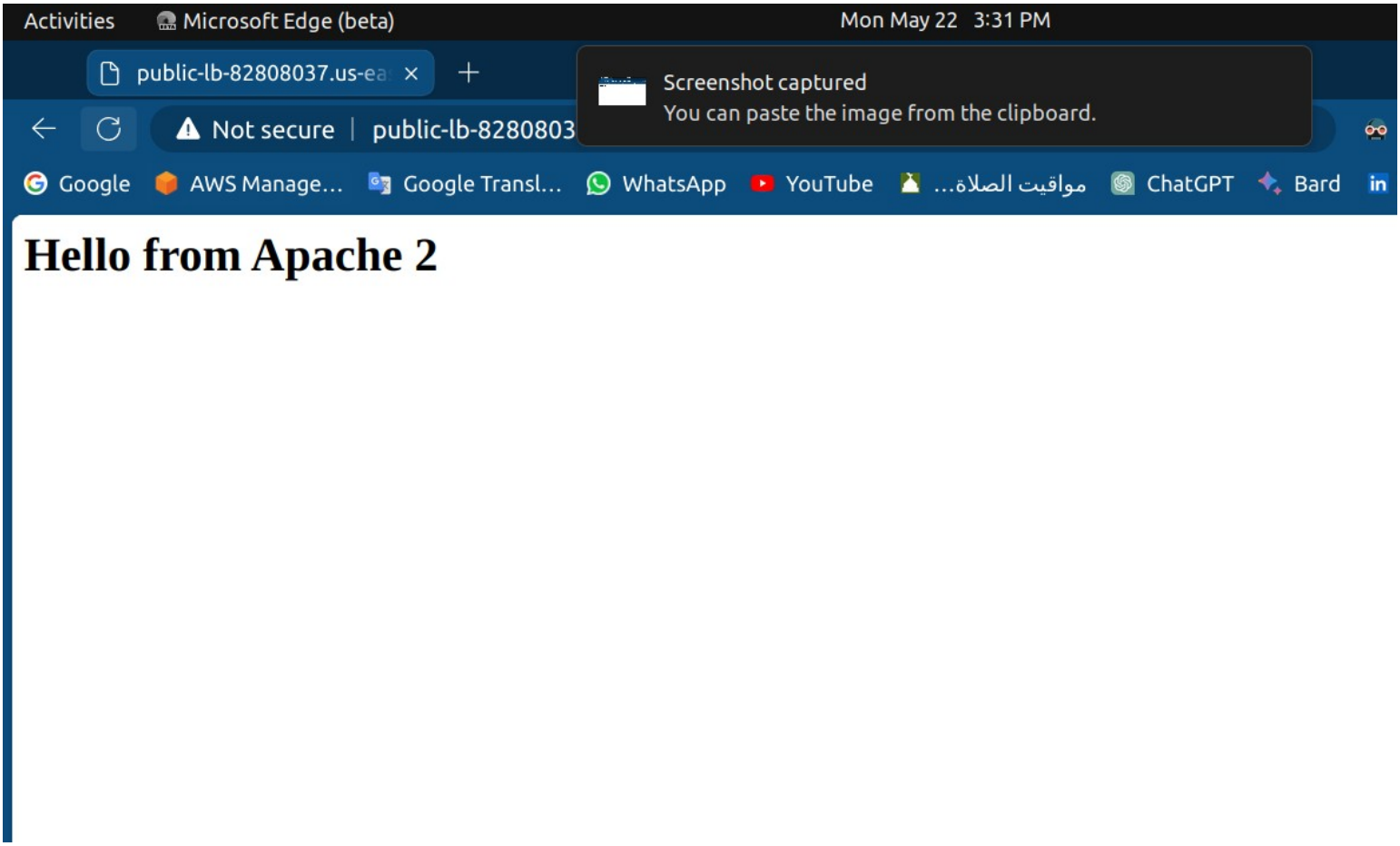
5.


remote_backend.tf
main.tf
main.tf > resource "aws_instance" "public_ec2_1" > [ ] security_groups > 0
115 data "aws_ami" "ubuntu" {
116 most_recent = true
117
118 filter {
119 name = "name"
120 values = ["ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-*"]
121 }
122
123 filter {
124 name = "virtualization-type"
125 values = ["hvm"]
126 }
127
128 owners = ["099720109477"] # Canonical
129 }
130
131 # expect this image id ami-053b0d53c279acc90
132 output "image_id" {
133 value = data.aws_ami.ubuntu.id
134 }
135
136 # Create the EC2 instances with userdata
137 resource "aws_instance" "public_ec2_1" {
138 ami = data.aws_ami.ubuntu.id
139 instance_type = "t2.micro"
140 subnet_id = module.public_subnet1.subnet_id
PROBLEMS OUTPUT CODEWHISPERER REFERENCE LOG DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK
}
Plan: 31 to add, 0 to change, 1 to destroy.
Changes to Outputs:
+ image_id = "ami-053b0d53c279acc90"
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value:

6.


Instances (4) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
Public apache 1 i-00eb063acc959cc32 Running t2.micro 2/2 checks passed No alarms us-east-1c
Public apache 2 i-0ac0a8766d2134c5d Running t2.micro 2/2 checks passed No alarms us-east-1d
Public nginx 2 i-034e5b35c0a43a047 Running t2.micro 2/2 checks passed No alarms us-east-1b 52.55.88.151
Public nginx 1 i-073bd6c3b189d0023 Running t2.micro 2/2 checks passed No alarms us-east-1a 44.192.121.208
EC2 > Load balancers
Load balancers (1/2)
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Actions
Create load balancer
Find resources by attribute or tag
Name
DNS name
State
VPC ID
Availability Zones
Type
Date created
public-lb public-lb-82808037.us-ea... Active vpc-0bdb5d2e1a9f6ec4c 2 Availability Zones application May 22, 2023, 15:25 (UTC+03:00)
private-lb internal-private-lb-60183... Active vpc-0bdb5d2e1a9f6ec4c 2 Availability Zones application May 22, 2023, 15:25 (UTC+03:00)
Load balancer: public-lb
Scheme
Internet-facing
Hosted zone
Z35SXDOTRQ7X7K
Availability Zones
subnet-0644a948c258e927b us-east-1a (use1-az1)
us-east-1b (use1-az2)
Date created
May 22, 2023, 15:25 (UTC+03:00)
DNS name copied
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:948763340657:loadbalancer/app/public-lb/34f46d9ae53f8095
public-lb-82808037.us-east-1.elb.amazonaws.com (A Record)
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Activities
Microsoft Edge (beta)
Mon May 22 3:30 PM
public-lb-82808037.us-ea
Not secure | public-lb-82808037.us-east-1.elb.amazonaws.com
Google
AWS Manage...
Google Transl...
WhatsApp
YouTube
مواقيت الصلاة...
ChatGPT
Bard
Hello from Apache 1


Activities
Microsoft Edge (beta)
Mon May 22 3:31 PM
public-lb-82808037.us-ea
Screenshot captured
You can paste the image from the clipboard.
Not secure | public-lb-8280803
Google
AWS Manage...
Google Transl...
WhatsApp
YouTube
مواقيت الصلاة...
ChatGPT
Bard
Hello from Apache 2