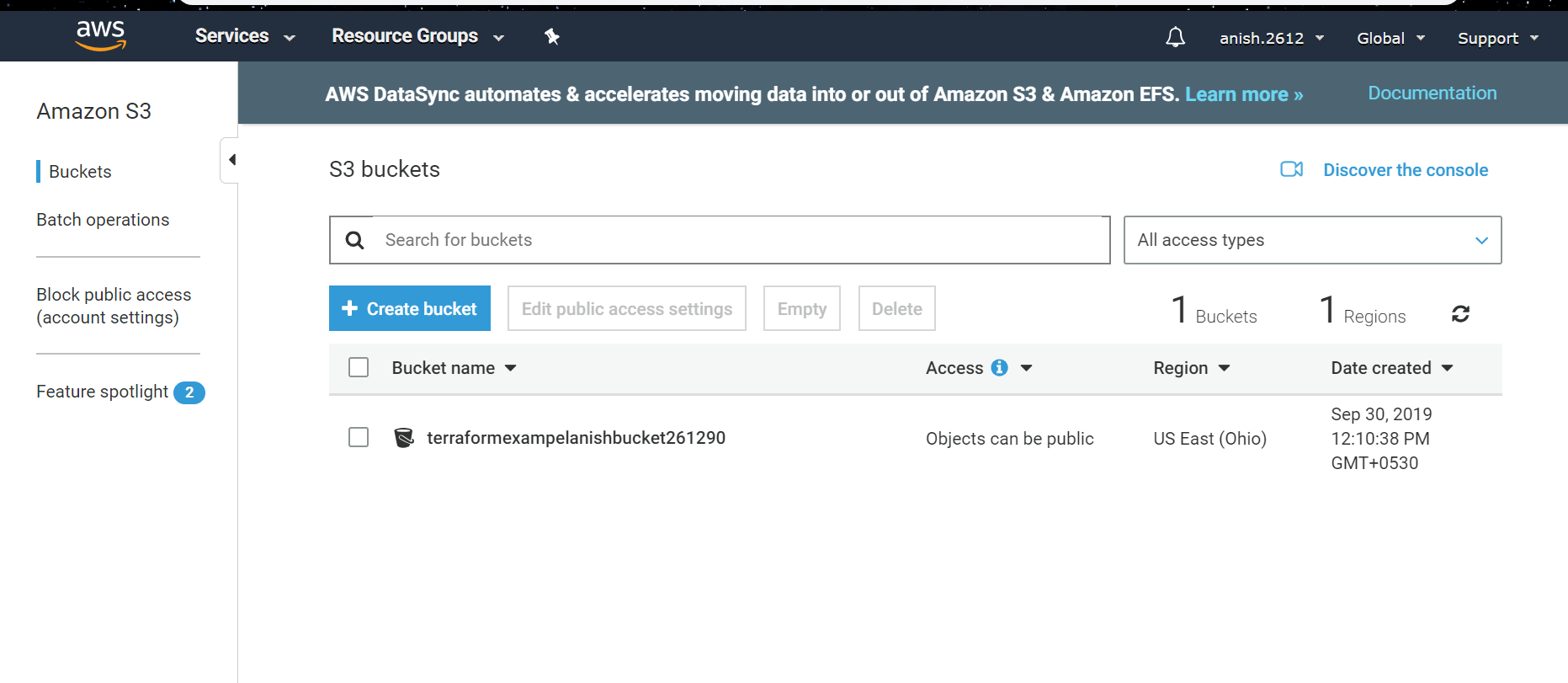
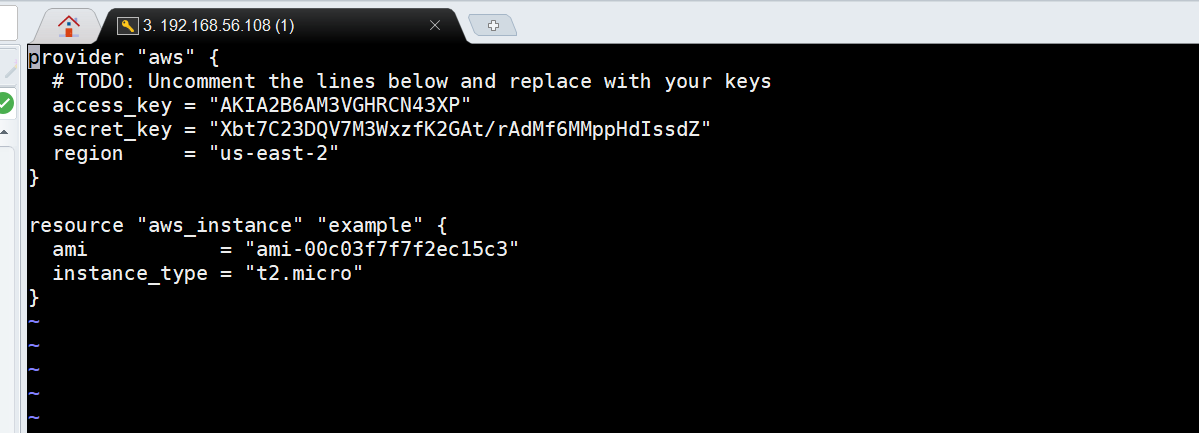
Terraform Remote State on Amazon S3

Created a bucket in aws console with the name arn: aws:s3:::terraformexampelanishbucket261290 for storing the terraform state.

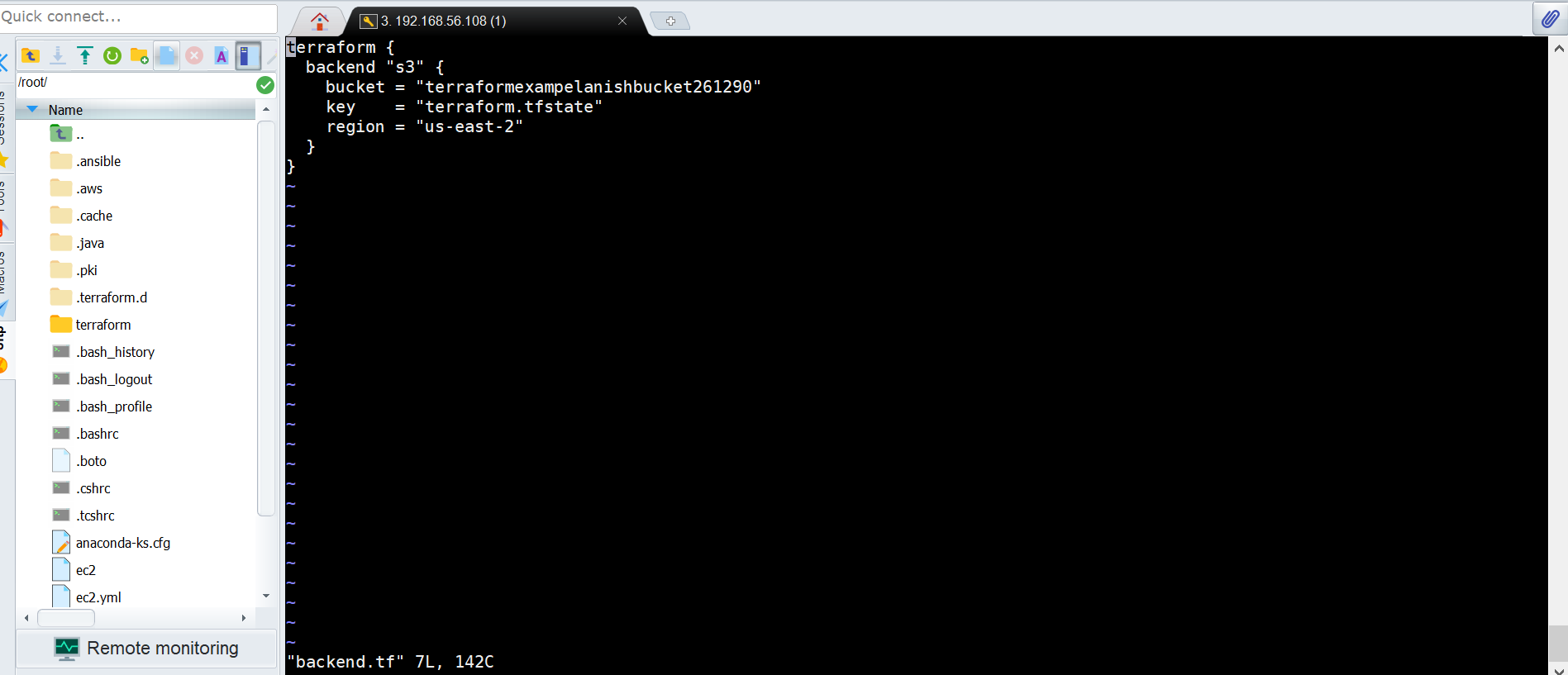


Created a sample terraform script(ec2.tf) with the below code.



The above code will create an ec2 instance in us-east-2 after execution.

After executing terraform apply the file with the name terraform. tfstate will be created in working directory. The above code will save this terraform. tfstate file in remote location (s3 bucket)



[root@localhost ec2a]# ls

backend.tf ec2.tf terraform.tfstate

[root@localhost ec2a]# rm terraform.tfstate

rm: remove regular file ‘terraform.tfstate’? yes

[root@localhost ec2a]# ls

backend.tf ec2.tf

[root@localhost ec2a]# terraform init

Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically

use this backend unless the backend configuration changes.

Initializing provider plugins...

The following providers do not have any version constraints in configuration,

so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking

changes, it is recommended to add version = "..." constraints to the

corresponding provider blocks in configuration, with the constraint strings

suggested below.

\* provider.aws: version = "~> 2.30"

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.

[root@localhost ec2a]# terraform destroy

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.

There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

Destroy complete! Resources: 0 destroyed.

[root@localhost ec2a]# ls

backend.tf ec2.tf

[root@localhost ec2a]# terraform apply

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

# aws\_instance.example will be created

+ resource "aws\_instance" "example" {

+ ami = "ami-00c03f7f7f2ec15c3"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ id = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.micro"

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = (known after apply)

+ network\_interface\_id = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ primary\_network\_interface\_id = (known after apply)

+ private\_dns = (known after apply)

+ private\_ip = (known after apply)

+ public\_dns = (known after apply)

+ public\_ip = (known after apply)

+ security\_groups = (known after apply)

+ source\_dest\_check = true

+ subnet\_id = (known after apply)

+ tenancy = (known after apply)

+ volume\_tags = (known after apply)

+ vpc\_security\_group\_ids = (known after apply)

+ ebs\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ snapshot\_id = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

+ ephemeral\_block\_device {

+ device\_name = (known after apply)

+ no\_device = (known after apply)

+ virtual\_name = (known after apply)

}

+ network\_interface {

+ delete\_on\_termination = (known after apply)

+ device\_index = (known after apply)

+ network\_interface\_id = (known after apply)

}

+ root\_block\_device {

+ delete\_on\_termination = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

}

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

aws\_instance.example: Creating...

aws\_instance.example: Still creating... [10s elapsed]

aws\_instance.example: Still creating... [20s elapsed]

aws\_instance.example: Still creating... [30s elapsed]

aws\_instance.example: Creation complete after 38s [id=i-0bd78c578f55370e5]

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

[root@localhost ec2a]# ls

backend.tf ec2.tf

[root@localhost ec2a]# vi ec2.tf

[root@localhost ec2a]# vi backend.tf

[root@localhost ec2a]#

