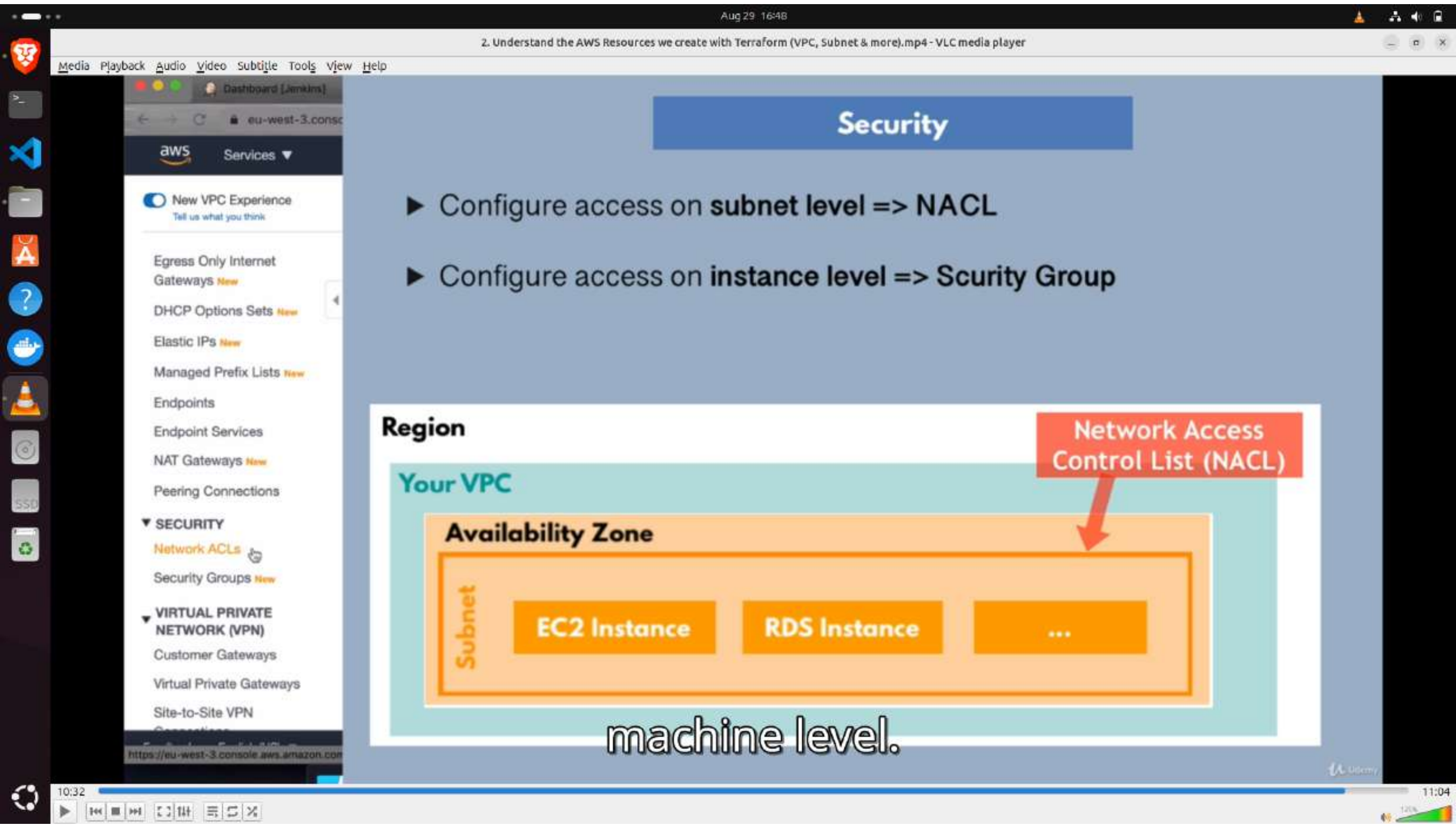
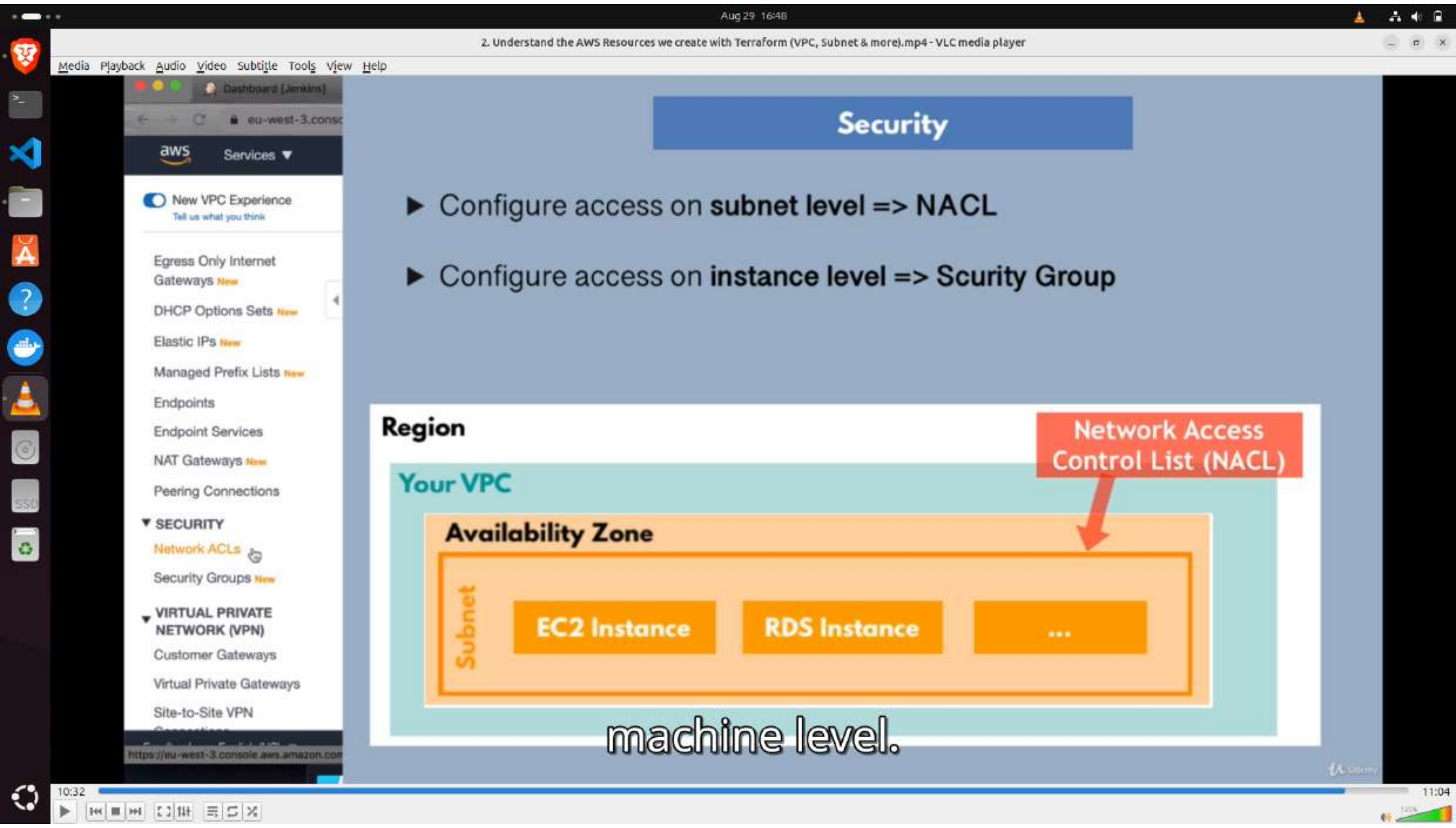


Security

- Configure access on **subnet level => NACL**
- Configure access on **instance level => Scurity Group**

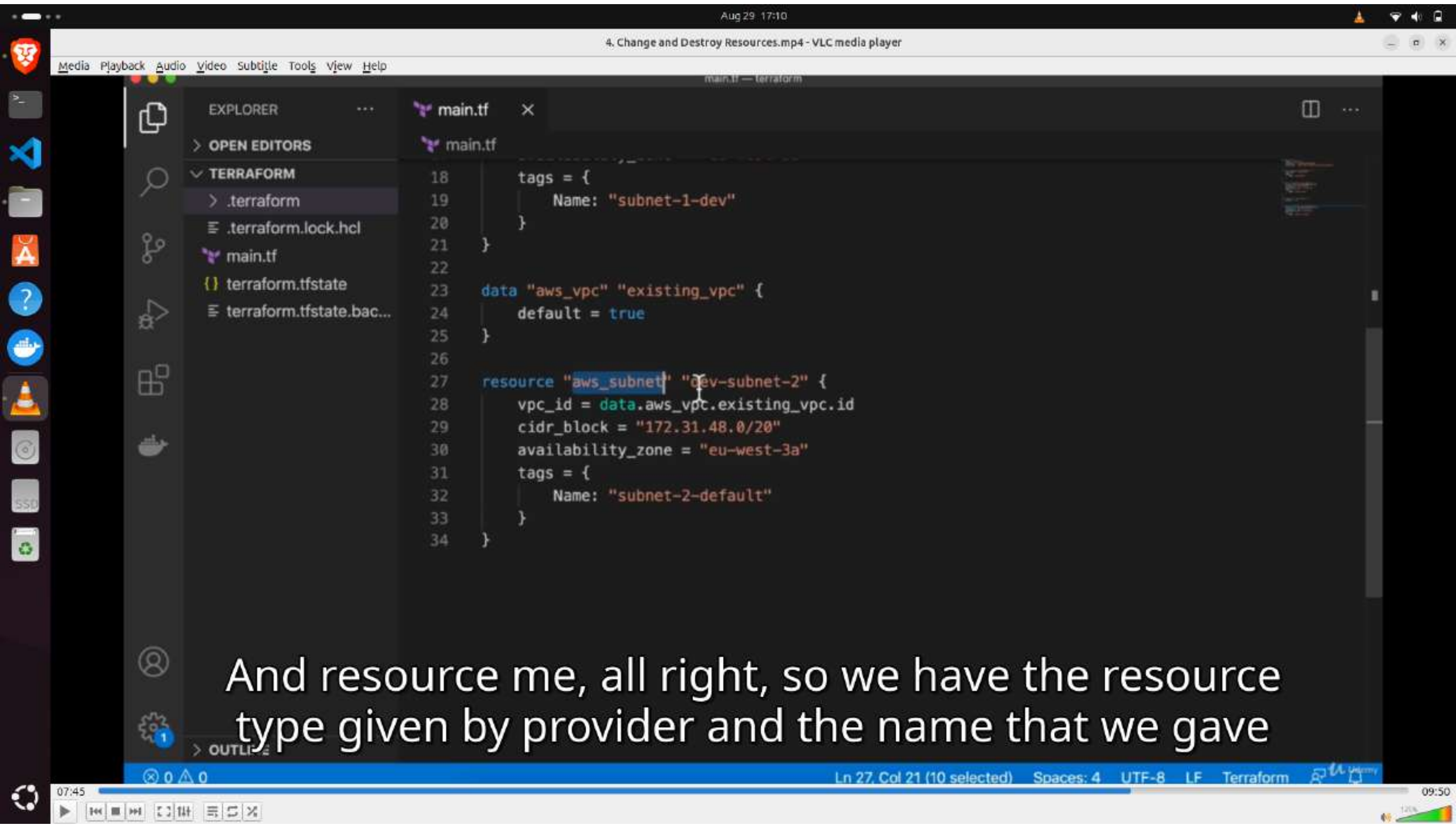
machine level.



Security

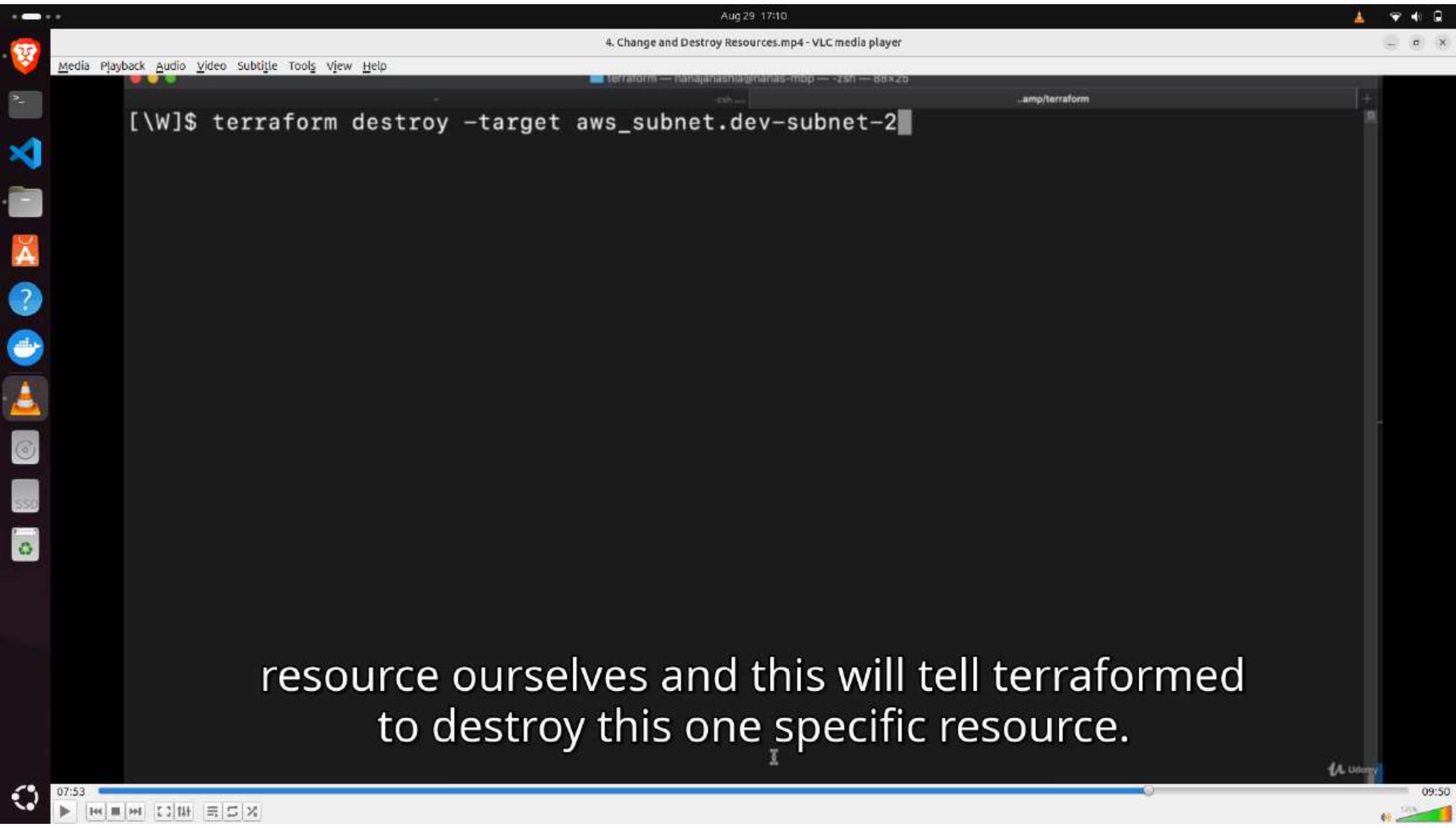
- Configure access on **subnet level => NACL**
- Configure access on **instance level => Scurity Group**

machine level.



```
    tags = {
        Name: "subnet-1-dev"
    }
}

data "aws_vpc" "existing_vpc" {
    default = true
}

resource "aws_subnet" "dev-subnet-2" {
    vpc_id = data.aws_vpc.existing_vpc.id
    cidr_block = "172.31.48.0/20"
    availability_zone = "eu-west-3a"
    tags = {
        Name: "subnet-2-default"
    }
}
```

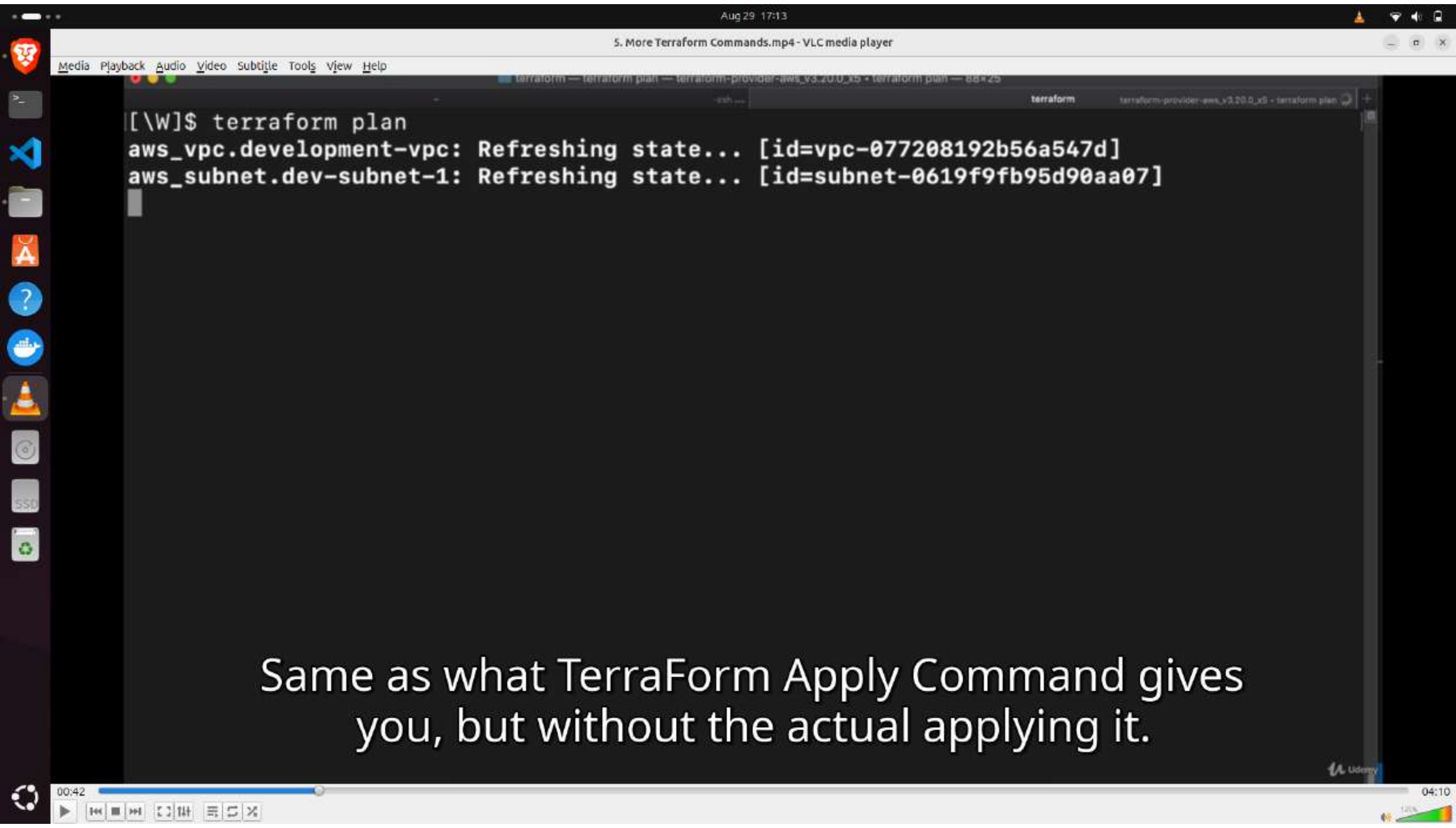
And resource me, all right, so we have the resource type given by provider and the name that we gave



Aug 29 17:10
4. Change and Destroy Resources.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
..amp/terraform
07:53
09:50

```
[\W]$ terraform destroy -target aws_subnet.dev-subnet-2
```

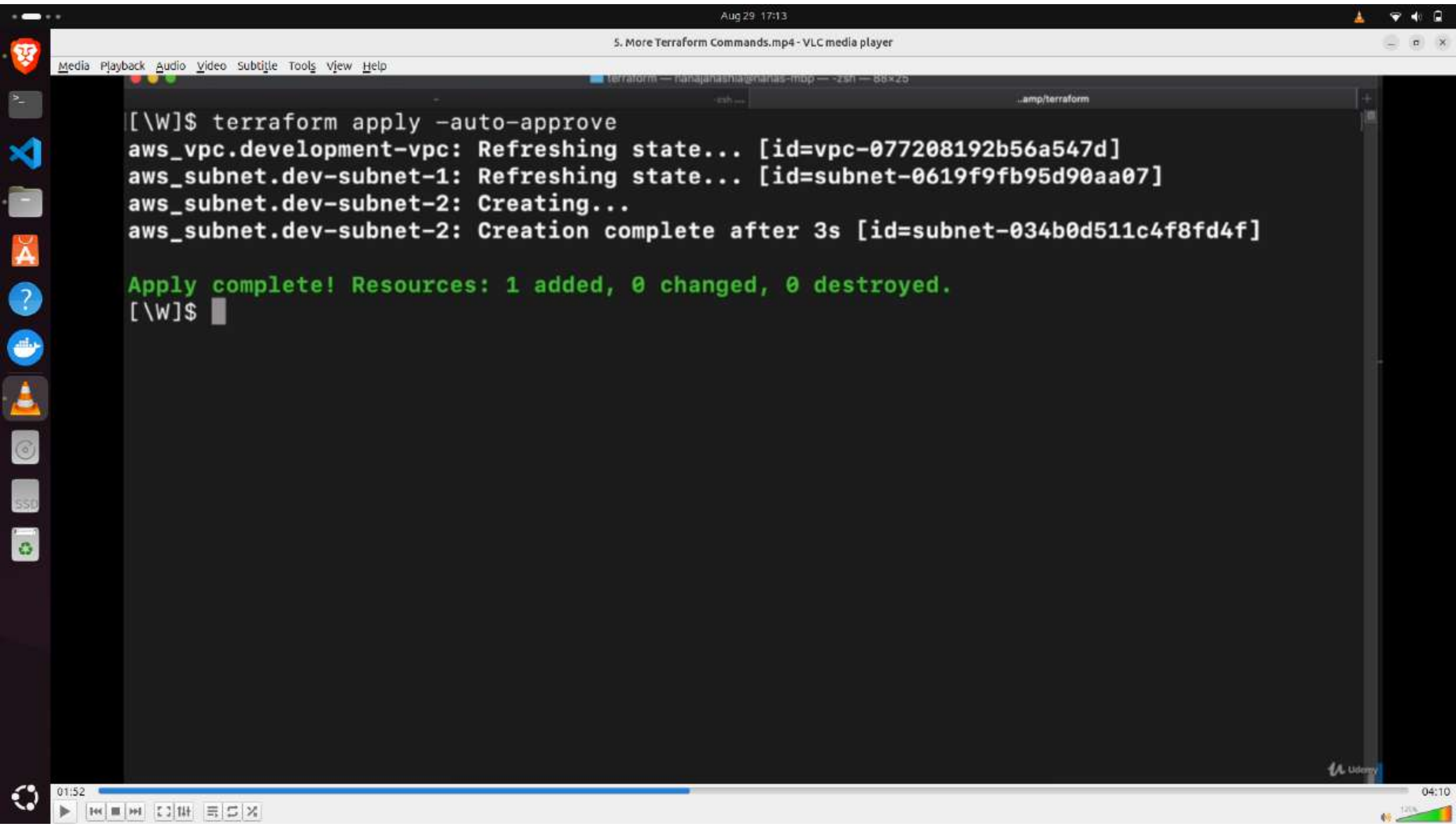
resource ourselves and this will tell terraformed
to destroy this one specific resource.



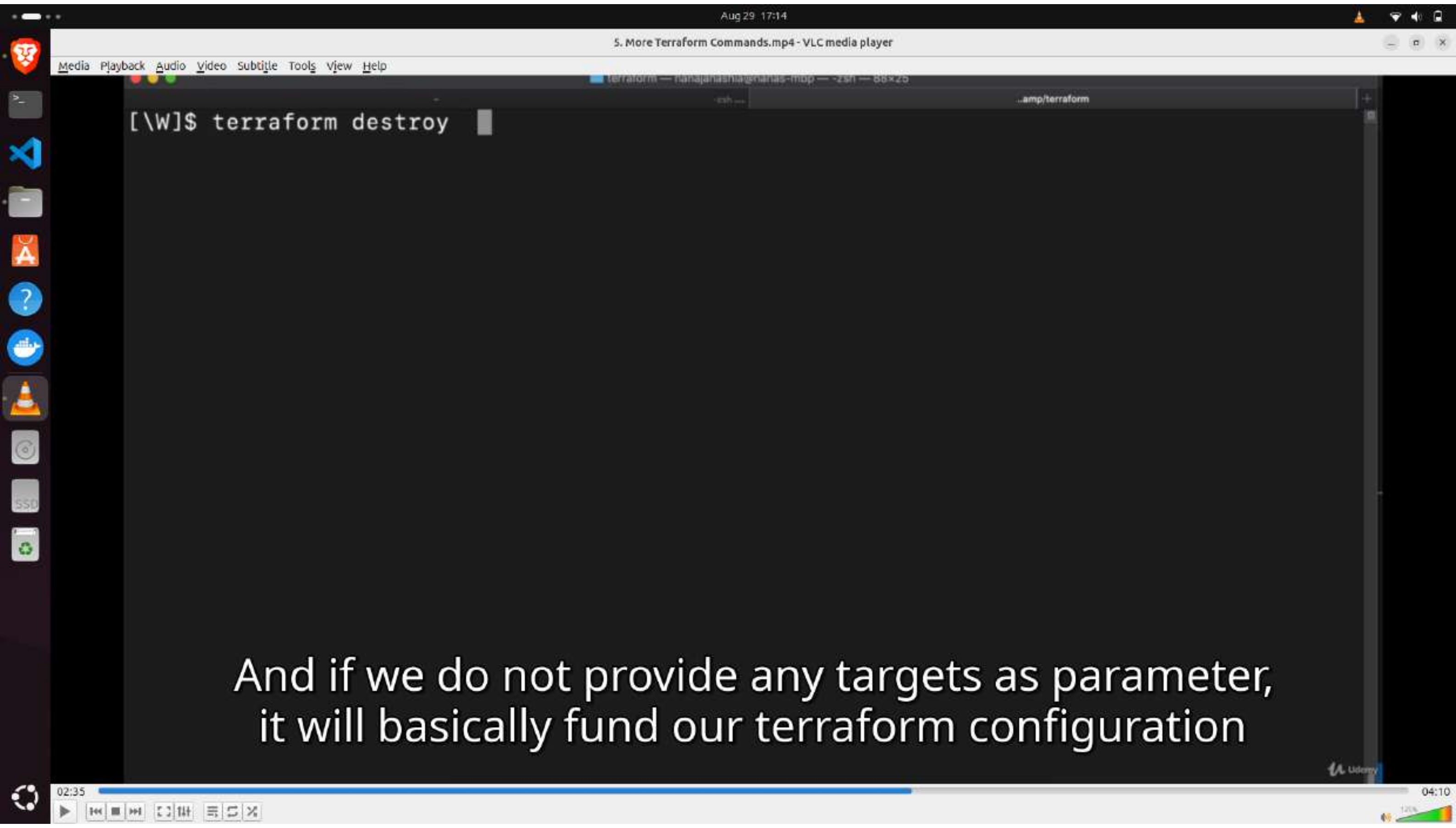
Aug 29 17:13
5. More Terraform Commands.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
terraform
00:42
04:10

```
[\W]$ terraform plan
aws_vpc.development-vpc: Refreshing state... [id=vpc-077208192b56a547d]
aws_subnet.dev-subnet-1: Refreshing state... [id=subnet-0619f9fb95d90aa07]
```

Same as what TerraForm Apply Command gives you, but without the actual applying it.

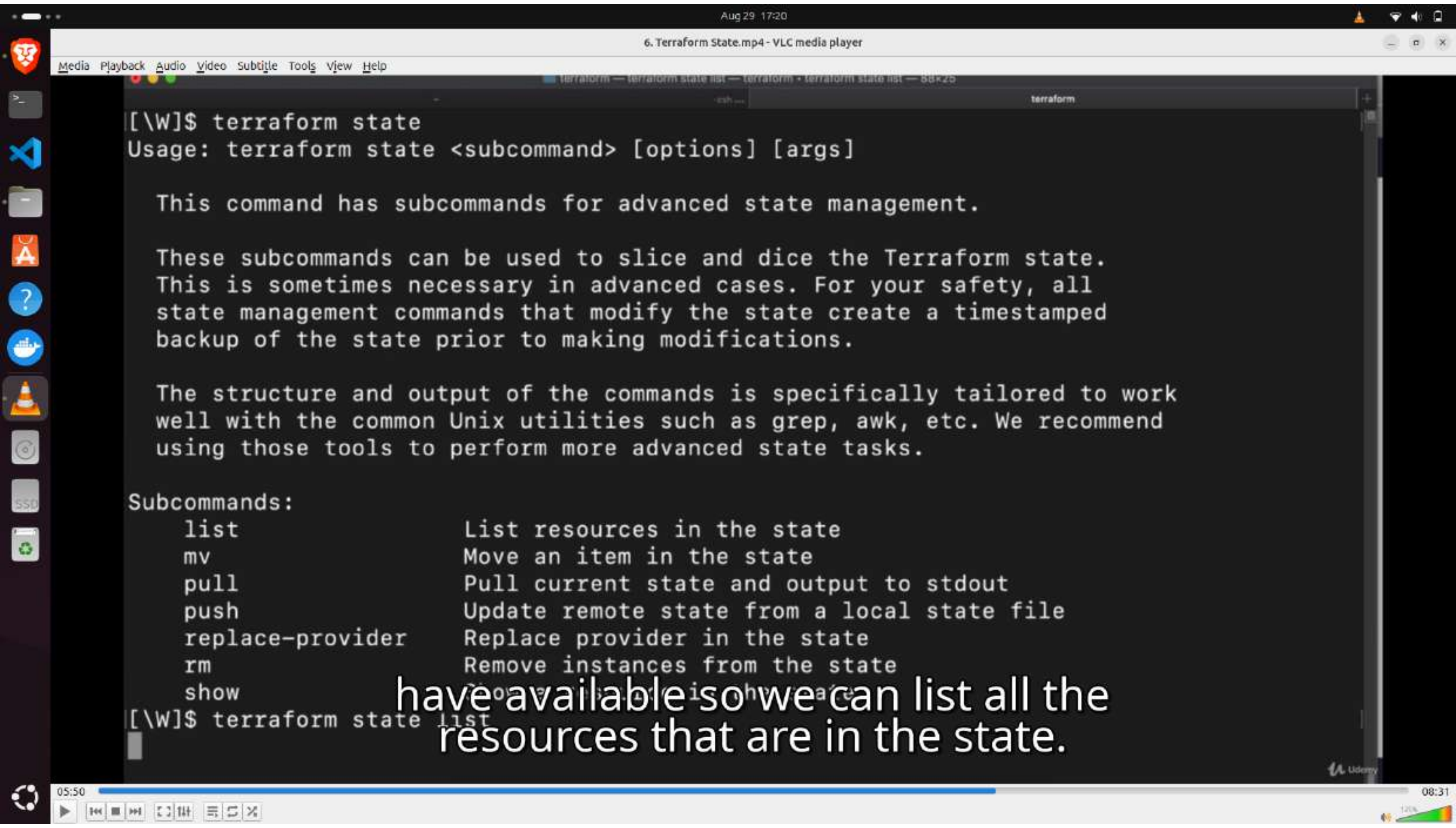


```
[\W]$ terraform apply -auto-approve
aws_vpc.development-vpc: Refreshing state... [id=vpc-077208192b56a547d]
aws_subnet.dev-subnet-1: Refreshing state... [id=subnet-0619f9fb95d90aa07]
aws_subnet.dev-subnet-2: Creating...
aws_subnet.dev-subnet-2: Creation complete after 3s [id=subnet-034b0d511c4f8fd4f]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[\W]$ 
```



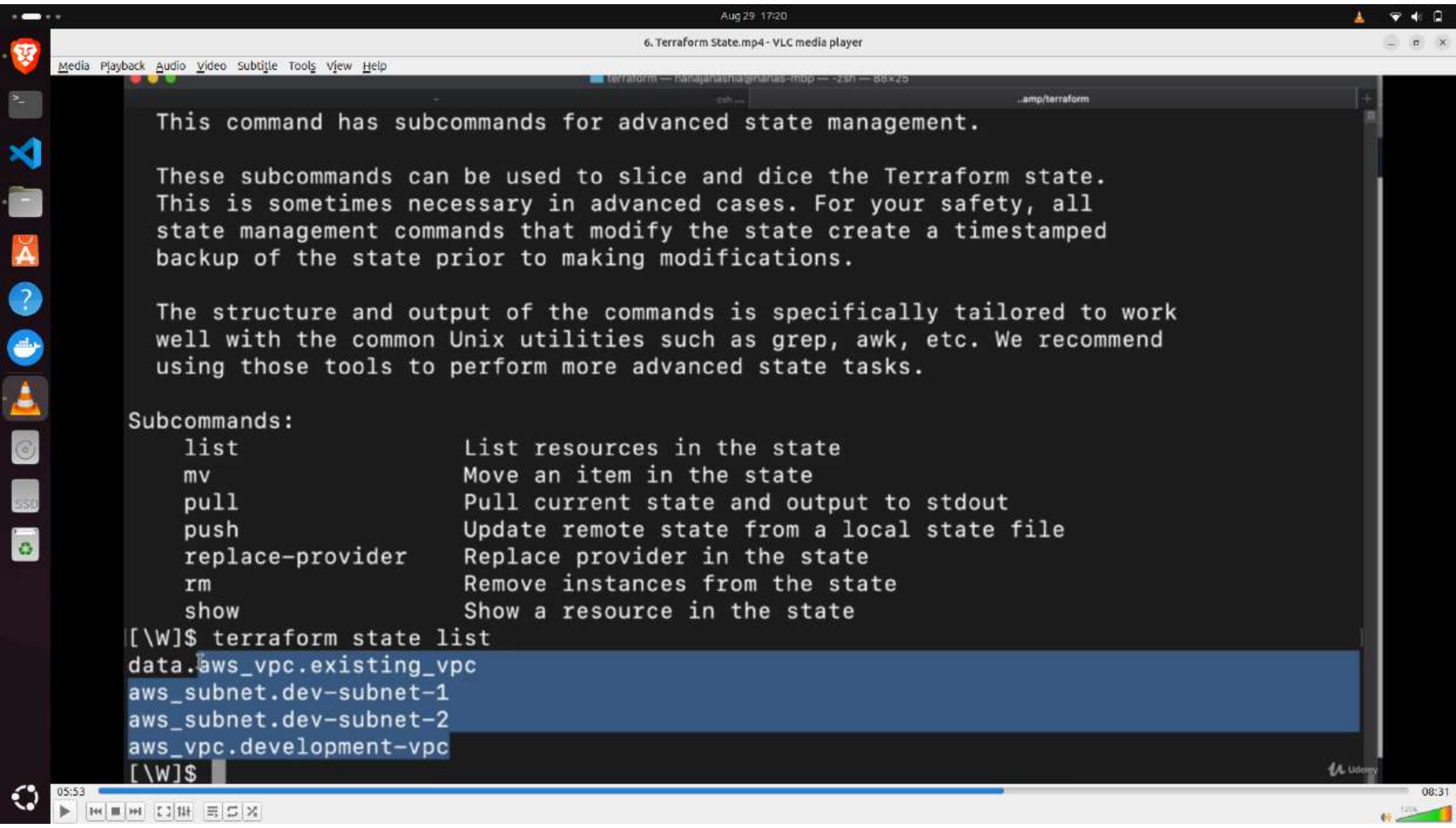
```
[\W]$ terraform destroy
```

And if we do not provide any targets as parameter,
it will basically fund our terraform configuration

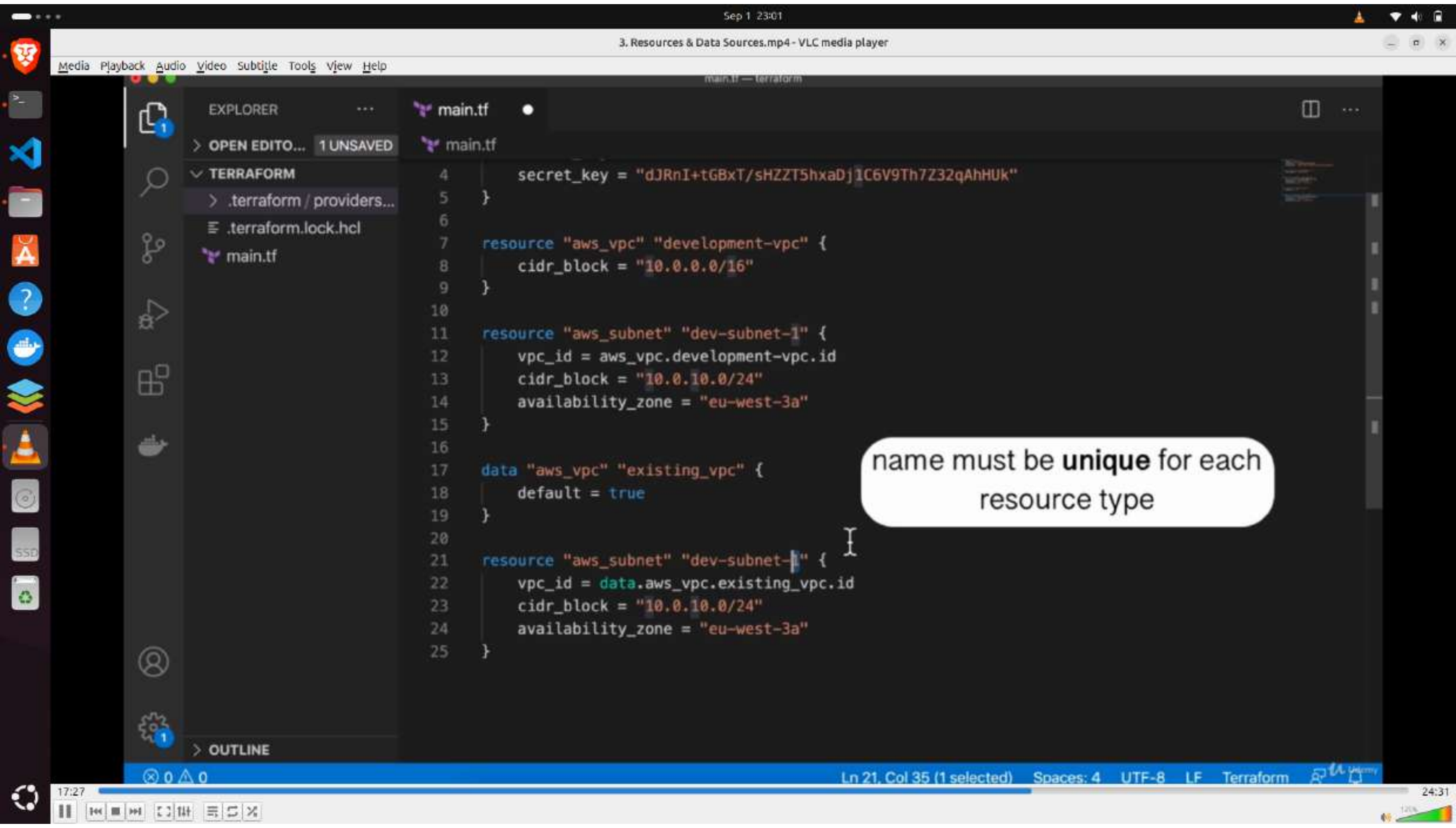

Aug 29 17:20
6. Terraform State.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
terraform
05:50
08:31

```
[\W]$ terraform state
Usage: terraform state <subcommand> [options] [args]

  This command has subcommands for advanced state management.

  These subcommands can be used to slice and dice the Terraform state.
  This is sometimes necessary in advanced cases. For your safety, all
  state management commands that modify the state create a timestamped
  backup of the state prior to making modifications.

  The structure and output of the commands is specifically tailored to work
  well with the common Unix utilities such as grep, awk, etc. We recommend
  using those tools to perform more advanced state tasks.

Subcommands:
    list                List resources in the state
    mv                  Move an item in the state
    pull                Pull current state and output to stdout
    push                Update remote state from a local state file
    replace-provider    Replace provider in the state
    rm                  Remove instances from the state
    show
[\W]$ terraform state list
```

have available so we can list all the
resources that are in the state.

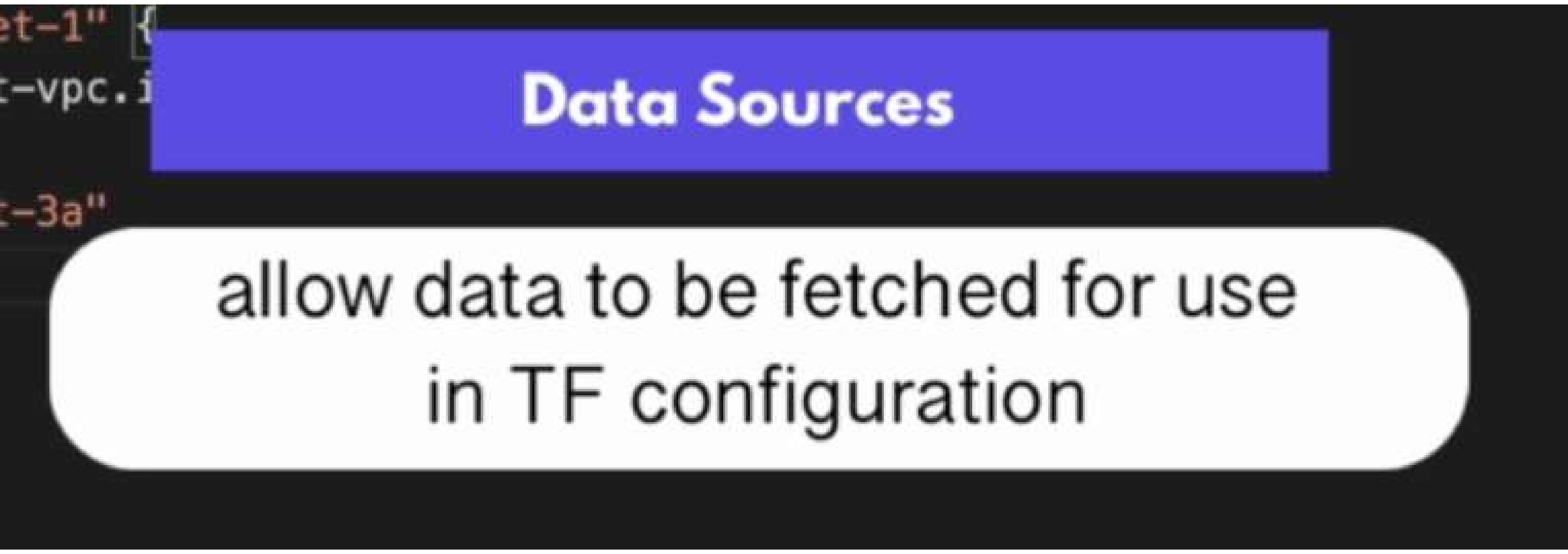


```
  This command has subcommands for advanced state management.

  These subcommands can be used to slice and dice the Terraform state.
  This is sometimes necessary in advanced cases. For your safety, all
  state management commands that modify the state create a timestamped
  backup of the state prior to making modifications.

  The structure and output of the commands is specifically tailored to work
  well with the common Unix utilities such as grep, awk, etc. We recommend
  using those tools to perform more advanced state tasks.

Subcommands:
    list                List resources in the state
    mv                  Move an item in the state
    pull                Pull current state and output to stdout
    push                Update remote state from a local state file
    replace-provider    Replace provider in the state
    rm                  Remove instances from the state
    show                Show a resource in the state
[\W]$ terraform state list
data.aws_vpc.existing_vpc
aws_subnet.dev-subnet-1
aws_subnet.dev-subnet-2
aws_vpc.development-vpc
[\W]$
```


Sep 1 23:01
3. Resources & Data Sources.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
main.tf — terraform
EXPLORER
OPEN EDITO... 1 UNSAVED
TERRAFORM
.terraform / providers...
.terraform.lock.hcl
main.tf
OUTLINE
⊗ 0 ⚠ 0
Ln 21, Col 35 (1 selected) Spaces: 4 UTF-8 LF Terraform
17:27
24:31

```
    secret_key = "dJRnI+tGBxT/sHZZT5hxaDj1C6V9Th7Z32qAhHUk"
}

resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = "10.0.10.0/24"
    availability_zone = "eu-west-3a"
}

data "aws_vpc" "existing_vpc" {
    default = true
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = data.aws_vpc.existing_vpc.id
    cidr_block = "10.0.10.0/24"
    availability_zone = "eu-west-3a"
}
```

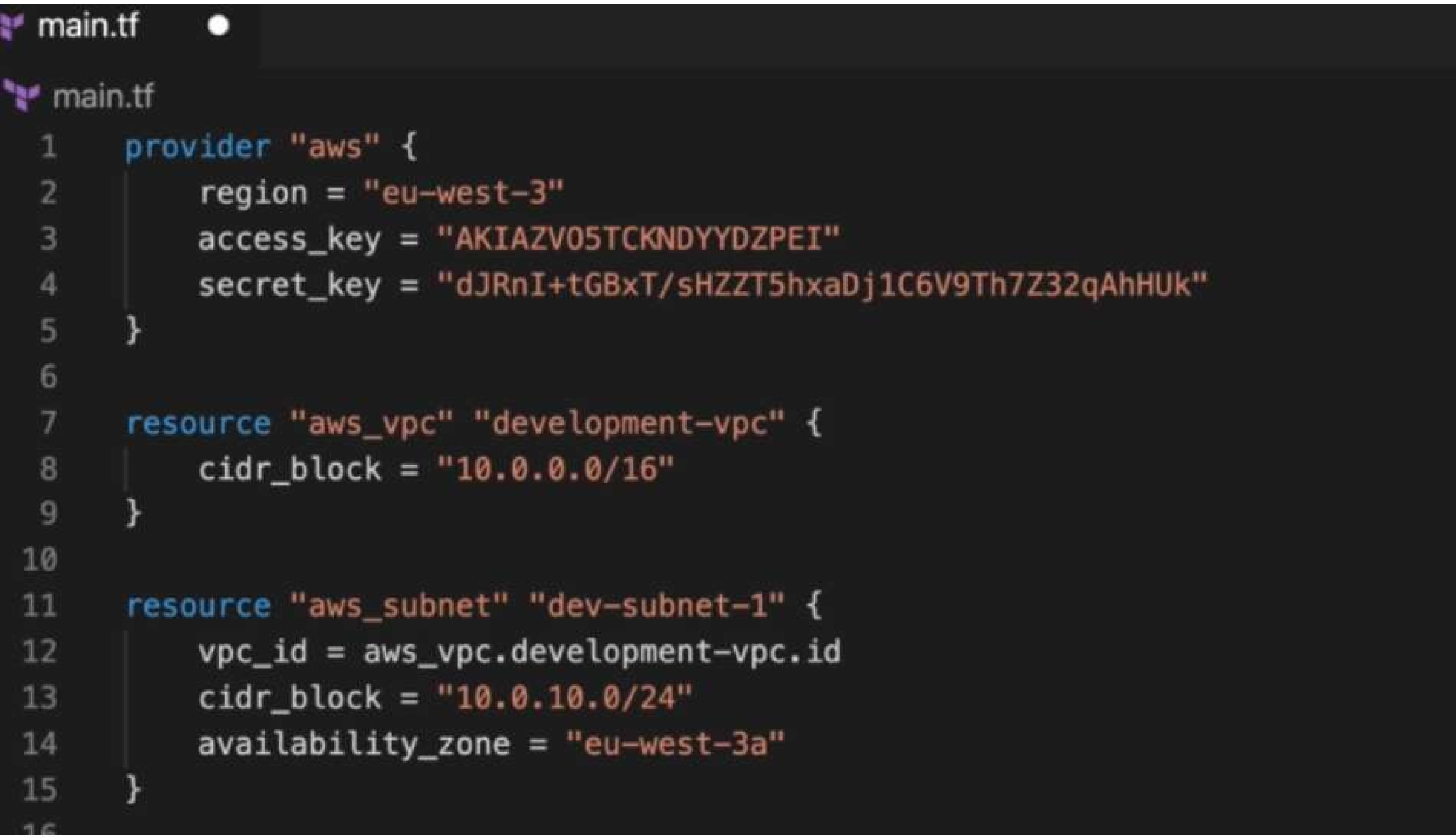
name must be **unique** for each resource type



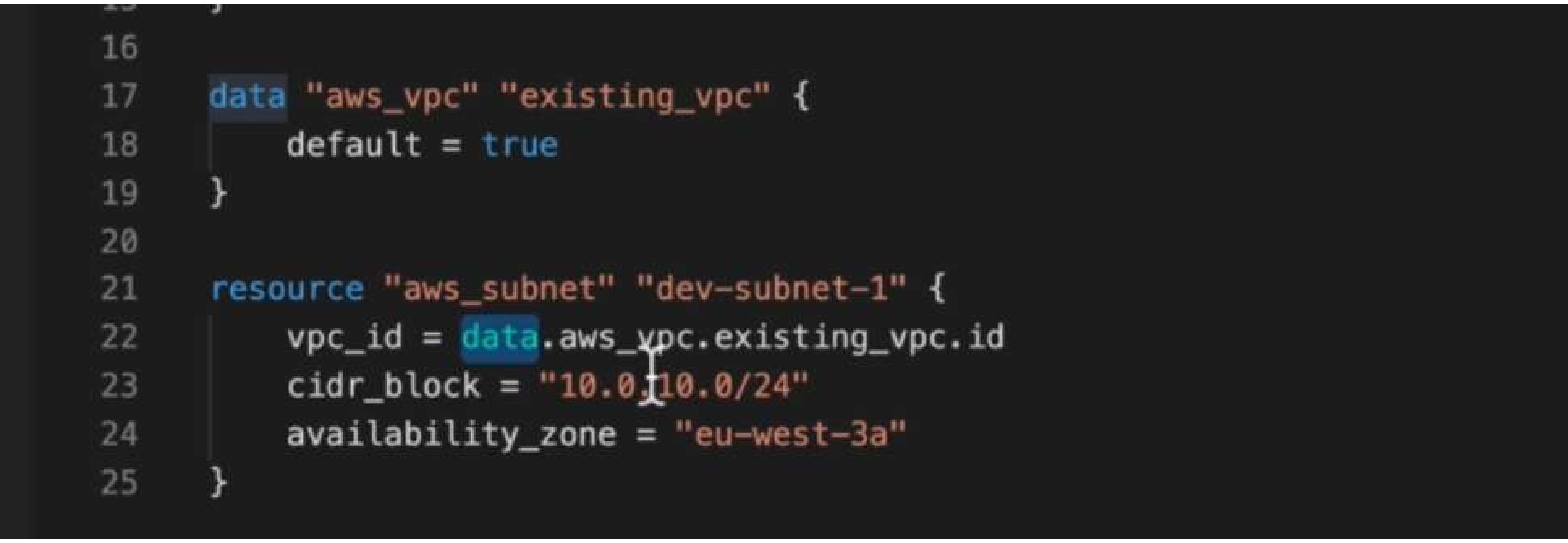
```
et-1" {
t-vpc.i

t-3a"
```

Data Sources

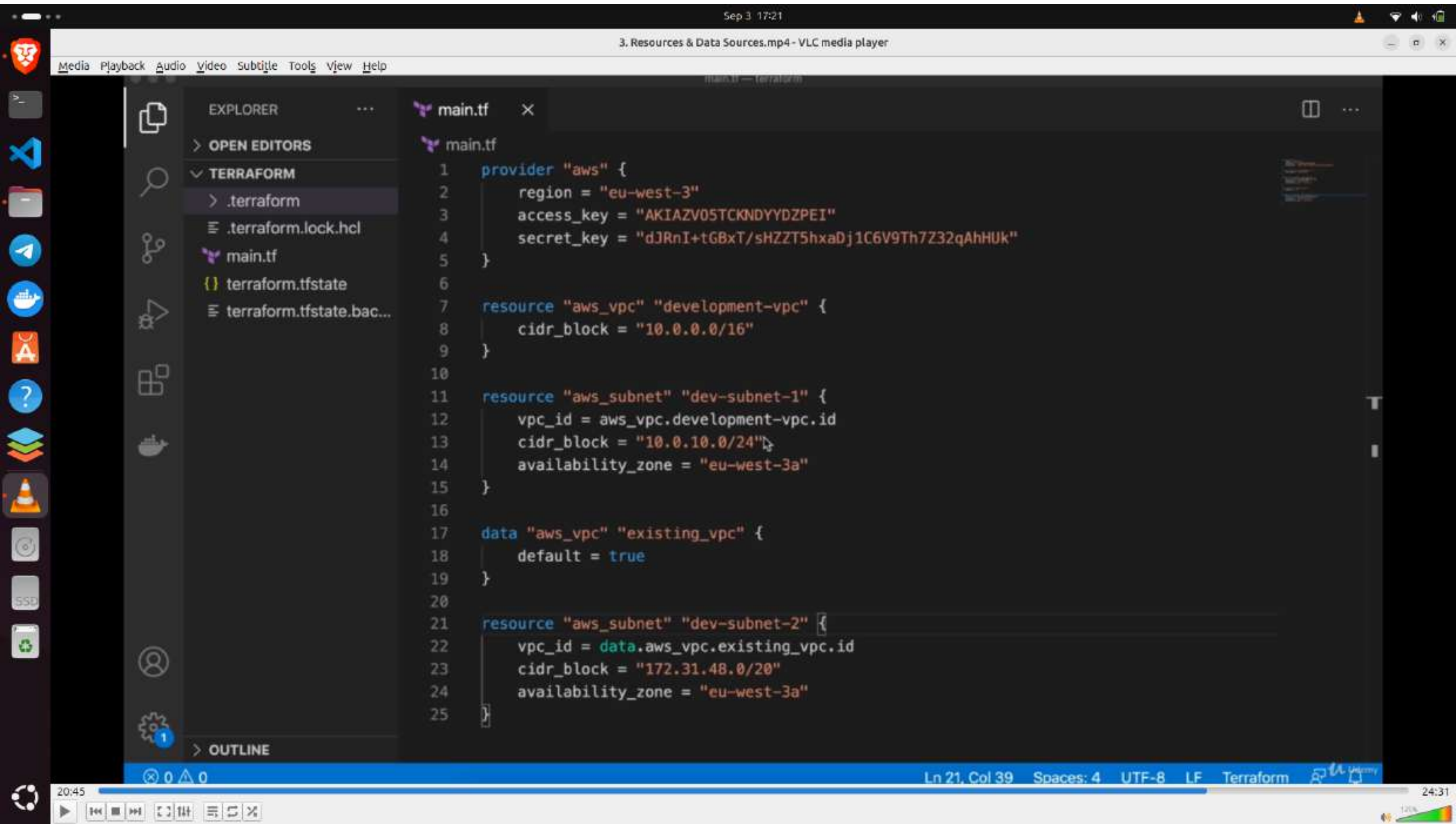
allow data to be fetched for use
in TF configuration



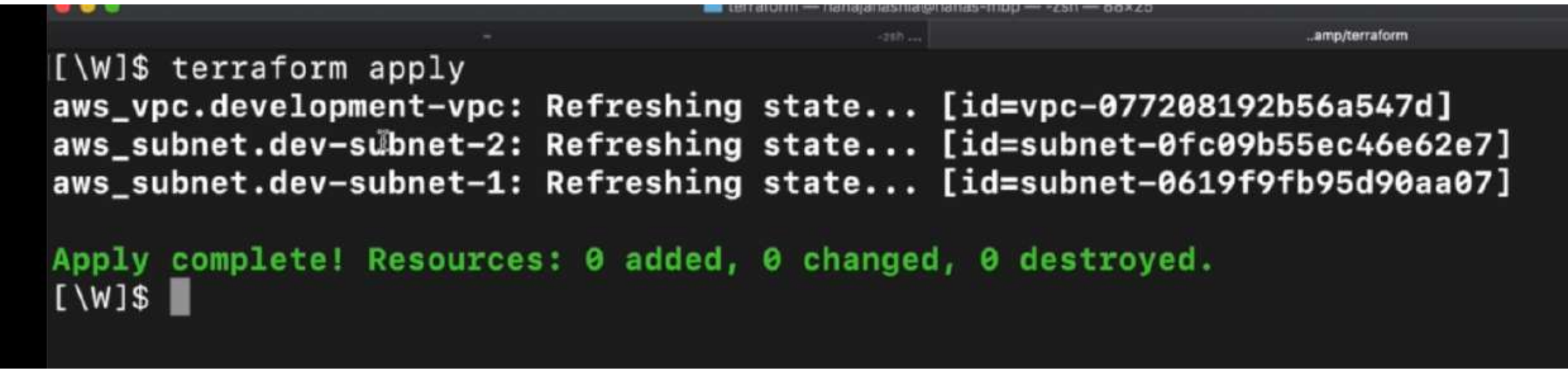
```
provider "aws" {
    region = "eu-west-3"
    access_key = "AKIAZVO5TCKNDYYDZPEI"
    secret_key = "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"
}

resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = "10.0.10.0/24"
    availability_zone = "eu-west-3a"
}
```



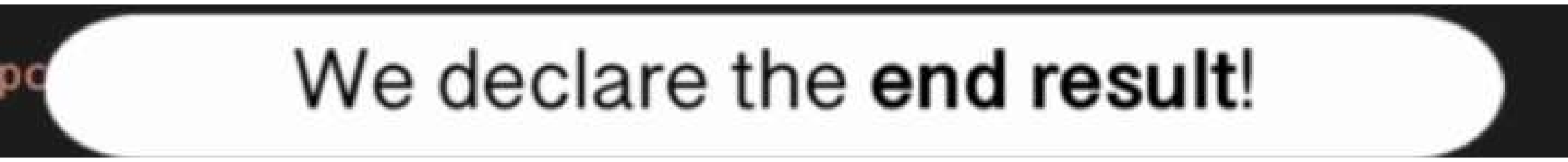
```
}

data "aws_vpc" "existing_vpc" {
    default = true
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = data.aws_vpc.existing_vpc.id
    cidr_block = "10.0.10.0/24"
    availability_zone = "eu-west-3a"
}
```



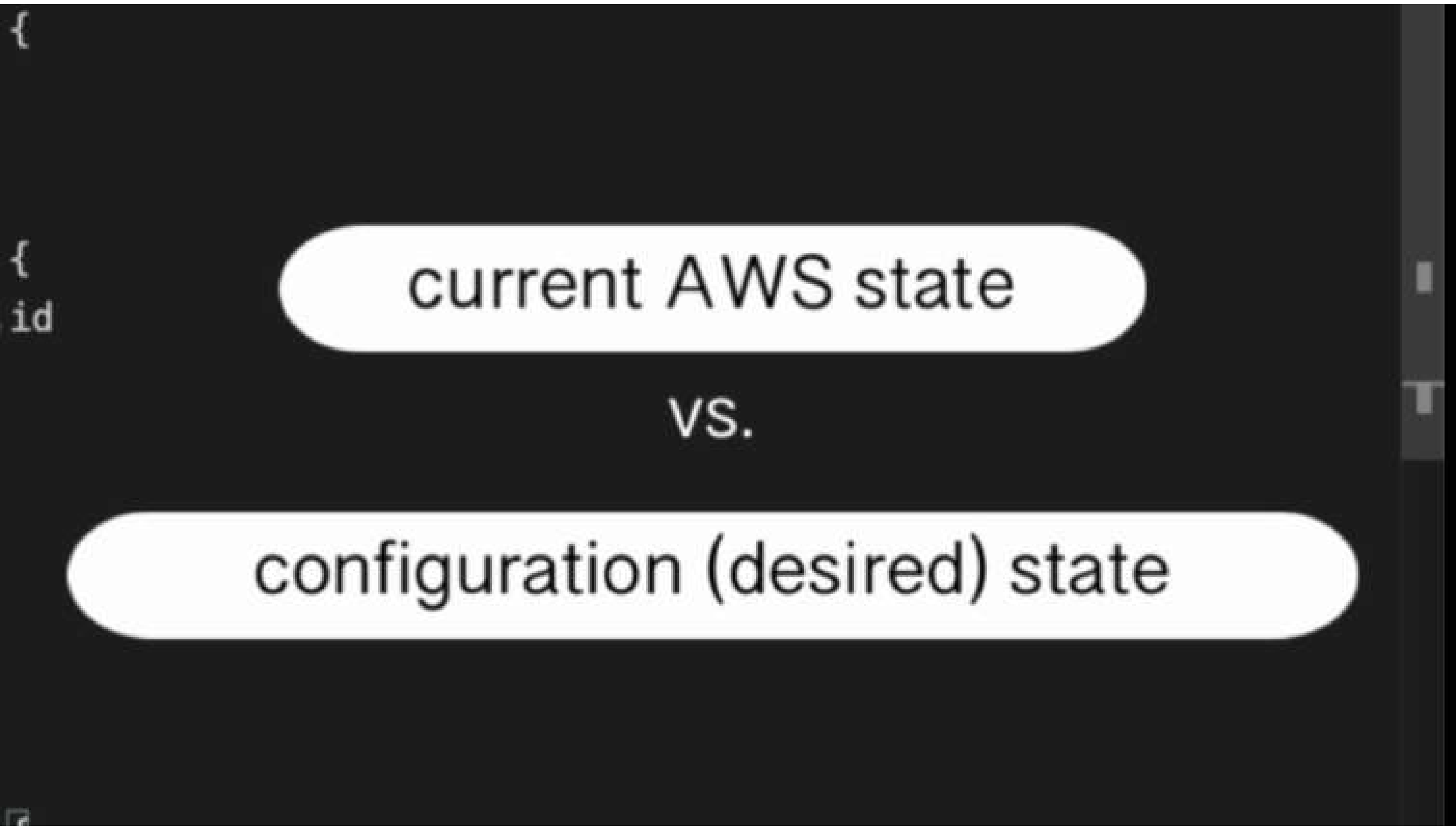
```
provider "aws" {
    region = "eu-west-3"
    access_key = "AKIAZVO5TCKNDYYDZPEI"
    secret_key = "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"
}

resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = "10.0.10.0/24"
    availability_zone = "eu-west-3a"
}

data "aws_vpc" "existing_vpc" {
    default = true
}

resource "aws_subnet" "dev-subnet-2" {
    vpc_id = data.aws_vpc.existing_vpc.id
    cidr_block = "172.31.48.0/20"
    availability_zone = "eu-west-3a"
}
```



```
[\W]$ terraform apply
aws_vpc.development-vpc: Refreshing state... [id=vpc-077208192b56a547d]
aws_subnet.dev-subnet-2: Refreshing state... [id=subnet-0fc09b55ec46e62e7]
aws_subnet.dev-subnet-1: Refreshing state... [id=subnet-0619f9fb95d90aa07]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
[\W]$
```



We declare the **end result**!

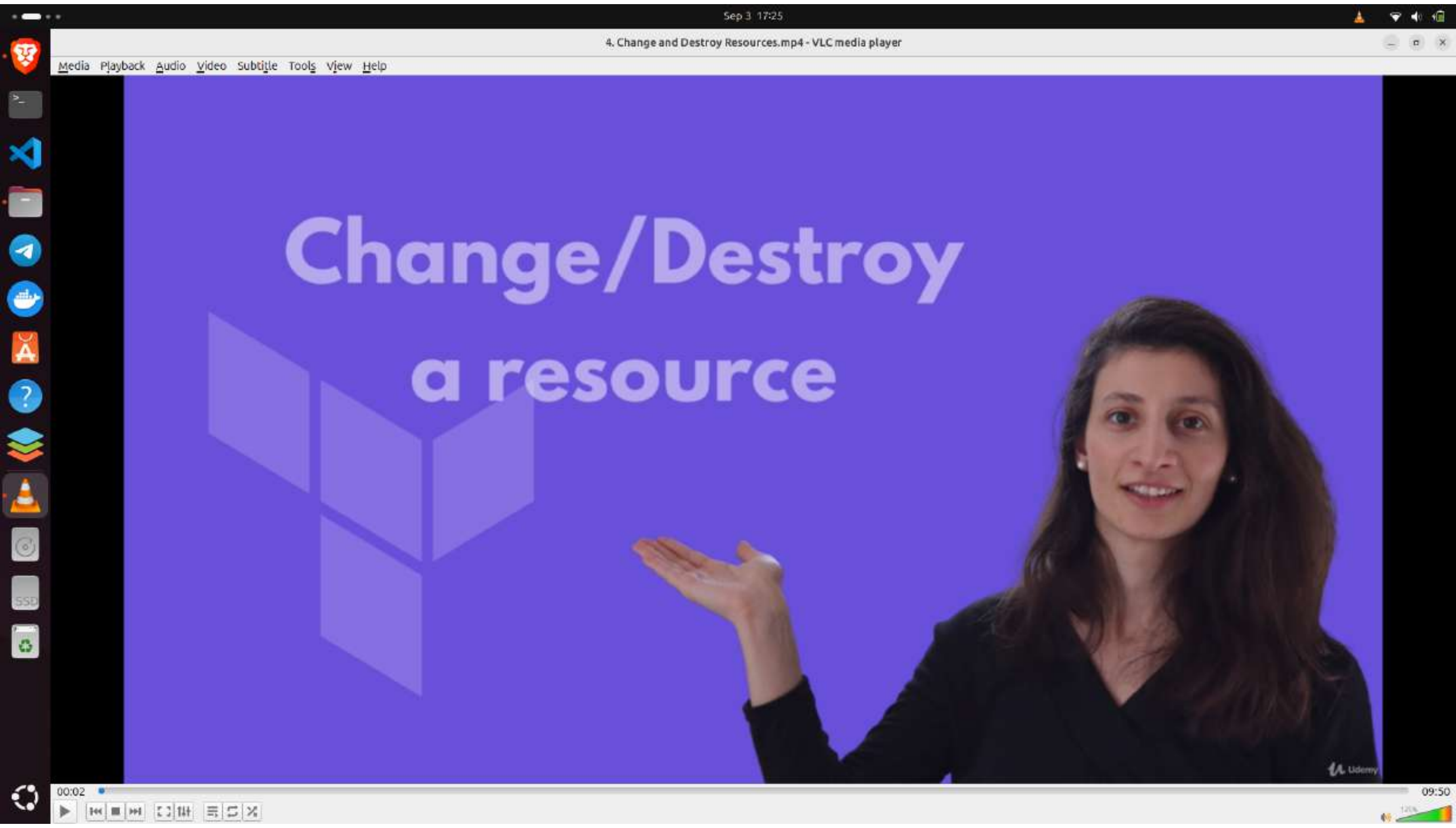


```
{

{
id
```

current AWS state

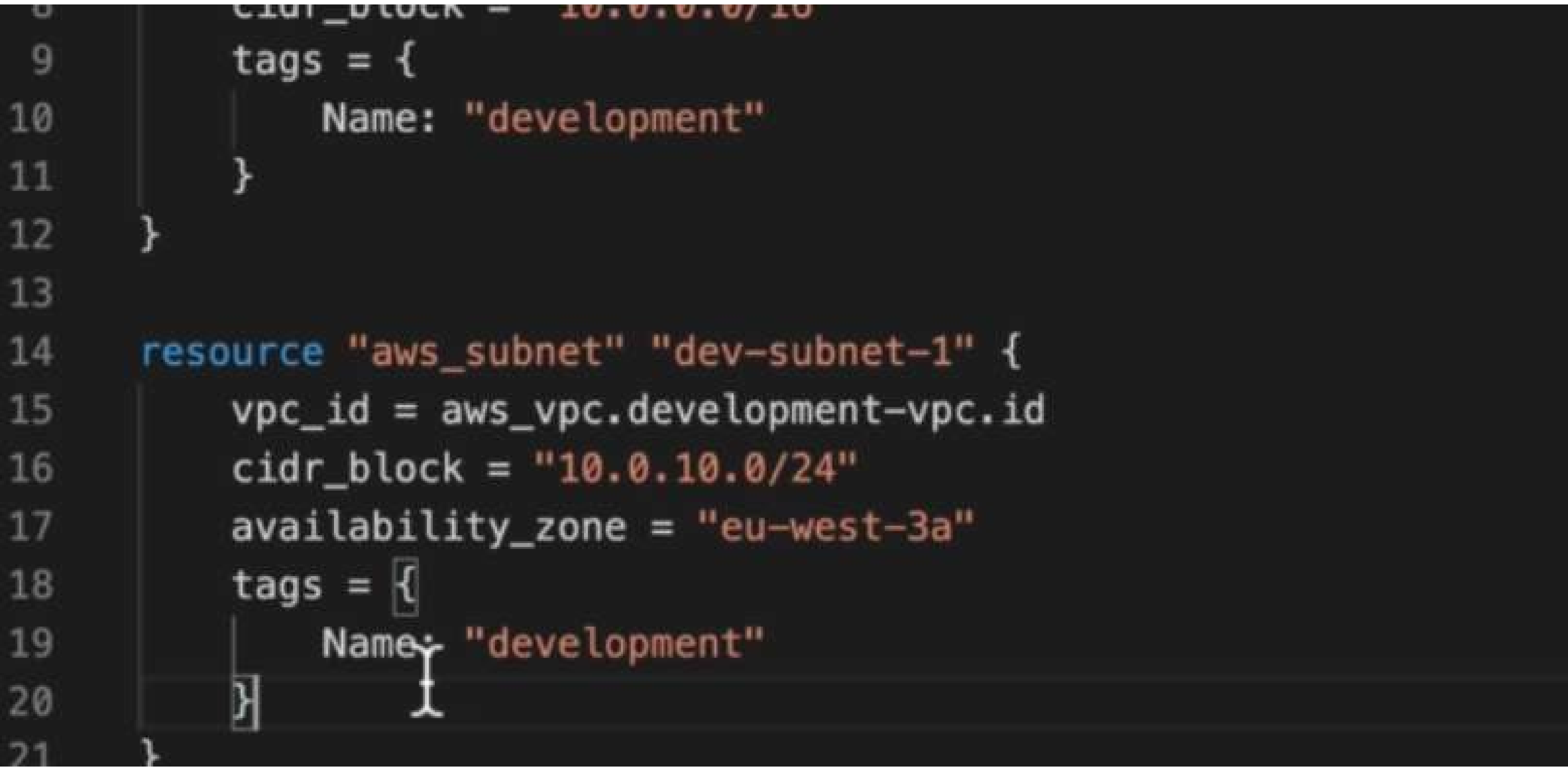
vs.

configuration (desired) state

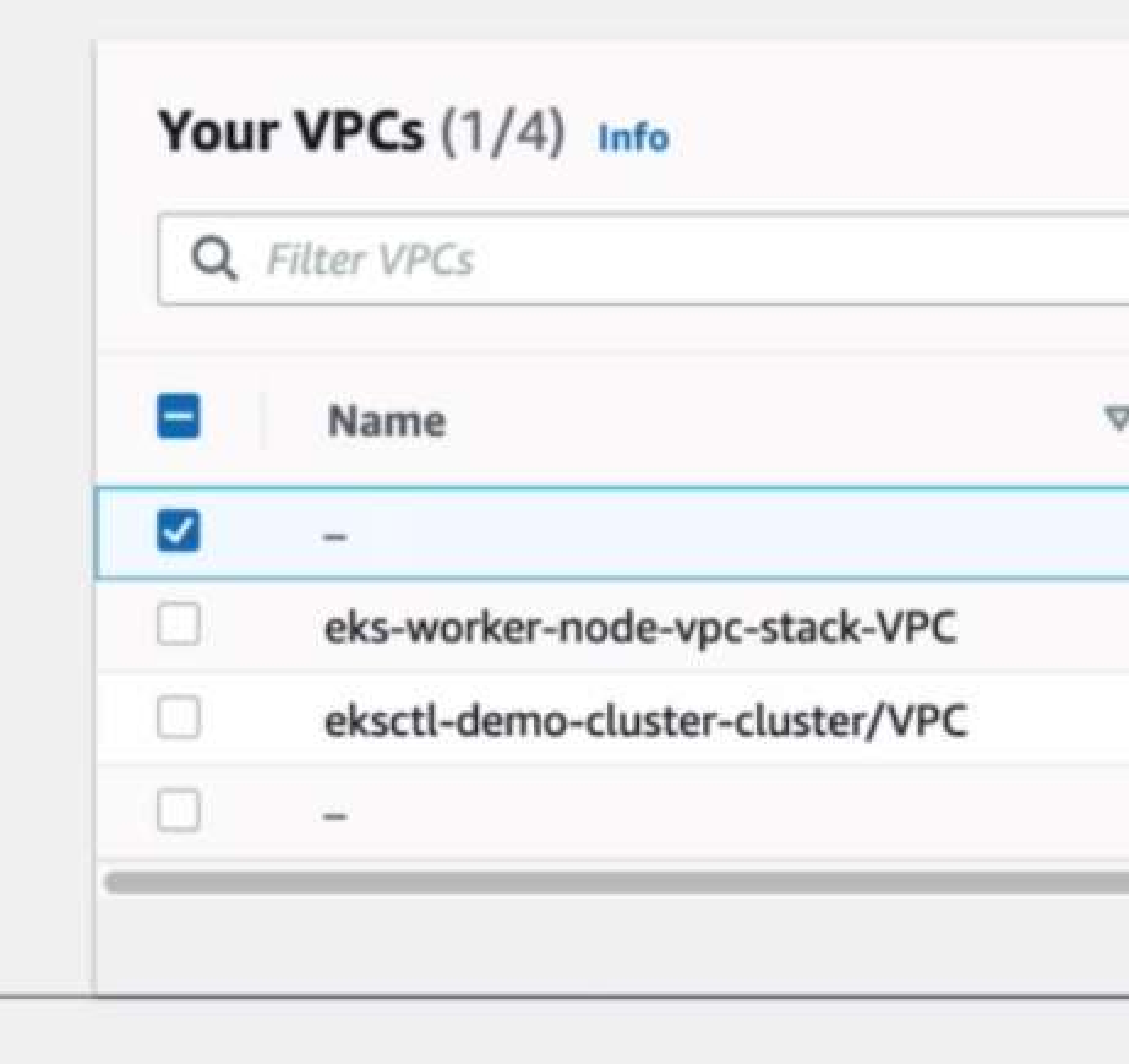


4. Change and Destroy Resources.mp4 - VLC media player

Change/Destroy
a resource



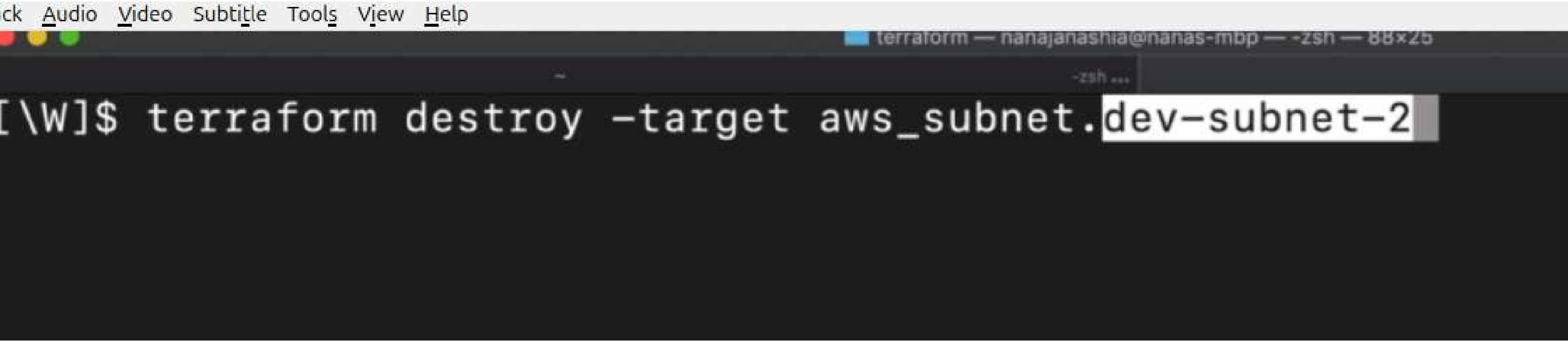
```
    cidr_block = "10.0.0.0/16"
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = "10.0.10.0/24"
    availability_zone = "eu-west-3a"
    tags = {
        Name: "development"
    }
}
```



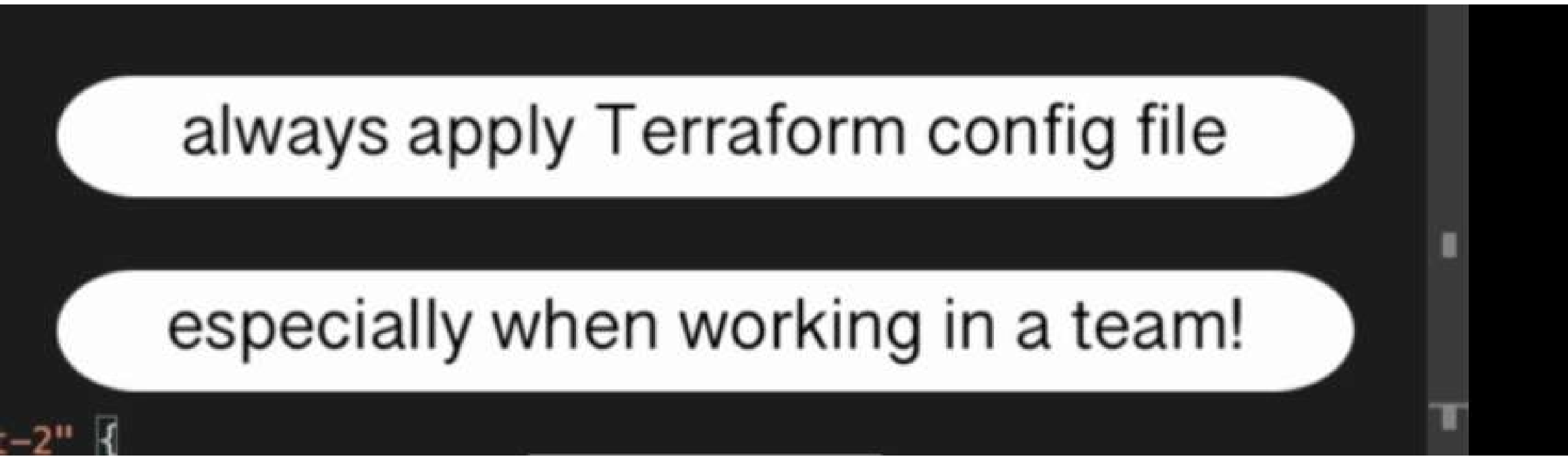
Your VPCs (1/4) Info

Filter VPCs

▣	Name
☑	–
□	eks-worker-node-vpc-stack-VPC
□	eksctl-demo-cluster-cluster/VPC
□	–


ck Audio Video Subtitle Tools View Help
terraform — nanajanashia@nanas-mbp — -zsh — 88×25
~
-zsh ...

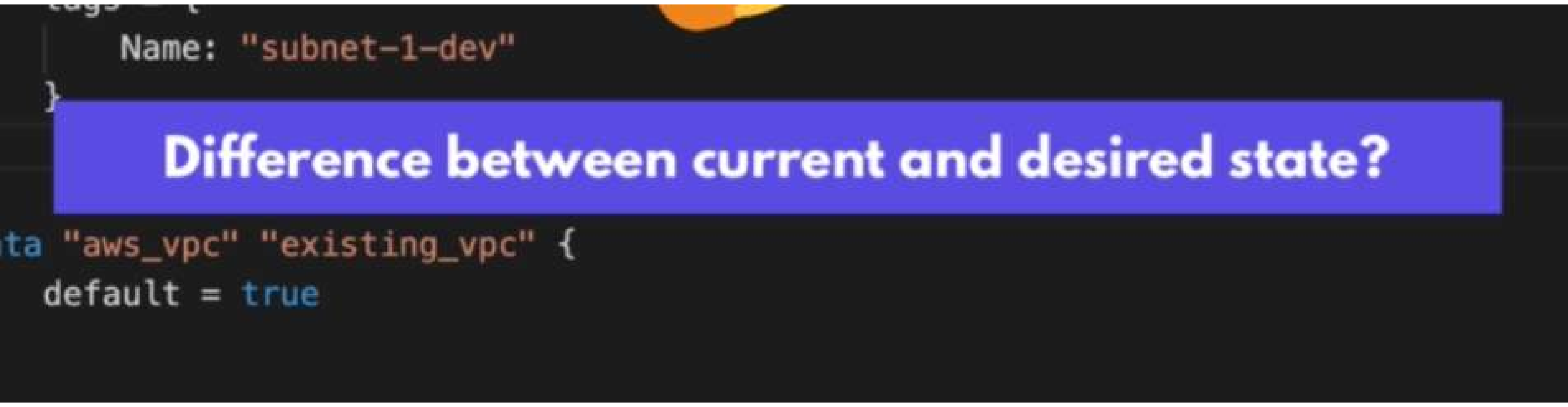
```
[\W]$ terraform destroy -target aws_subnet.dev-subnet-2
```



always apply Terraform config file

especially when working in a team!

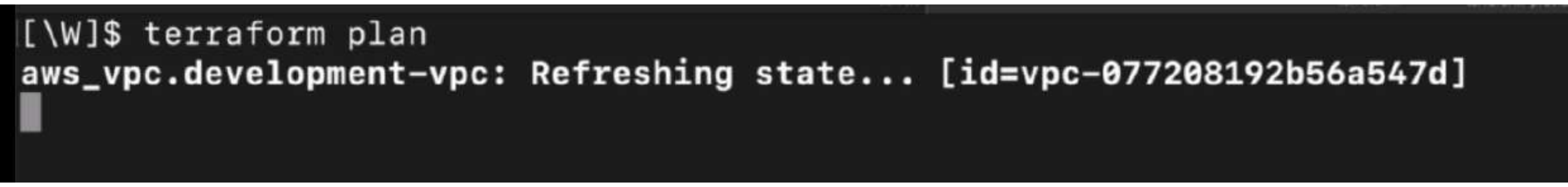
```
-2" {
```



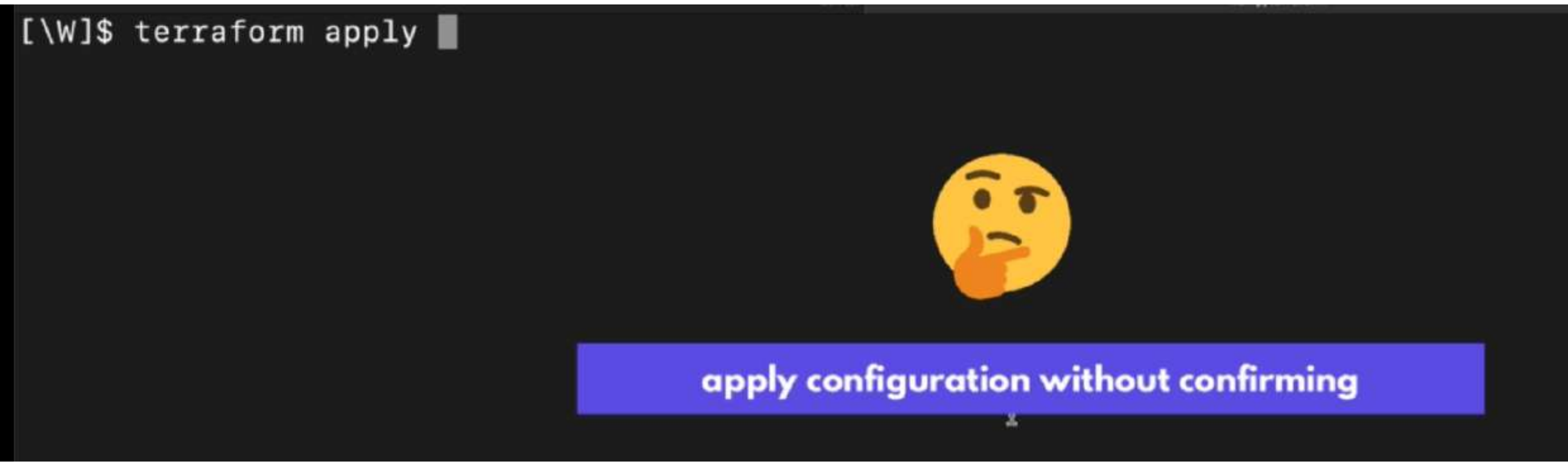
```
  tags = {
    Name: "subnet-1-dev"
  }
```

Difference between current and desired state?

```
ta "aws_vpc" "existing_vpc" {
  default = true
```

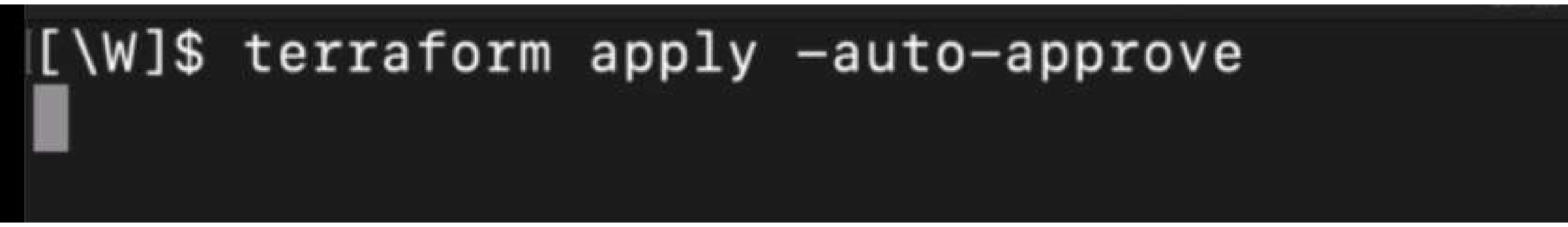


```
[\W]$ terraform plan
aws_vpc.development-vpc: Refreshing state... [id=vpc-077208192b56a547d]
```

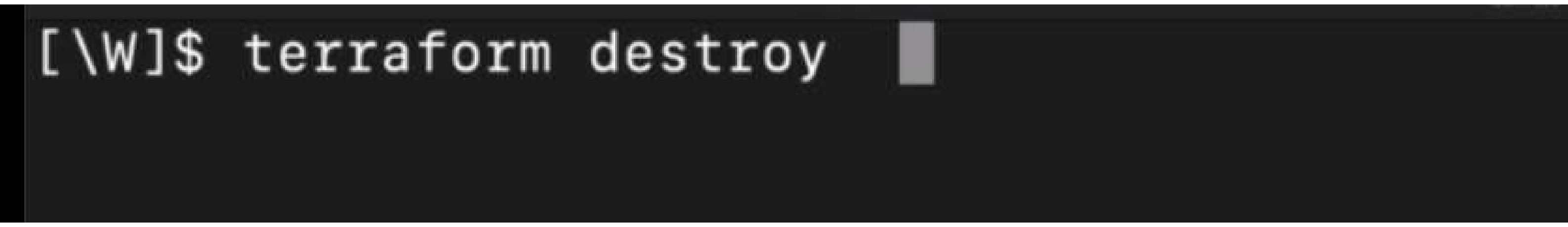


```
[\W]$ terraform apply
```

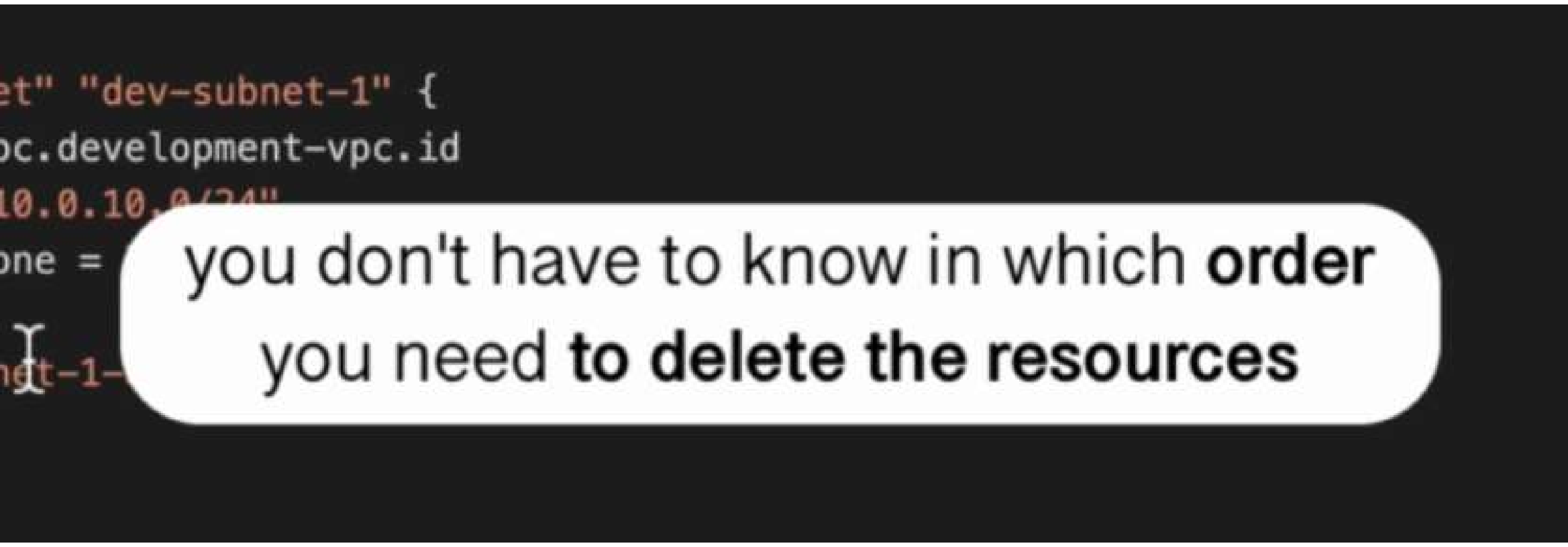
apply configuration without confirming



```
[\W]$ terraform apply -auto-approve
```

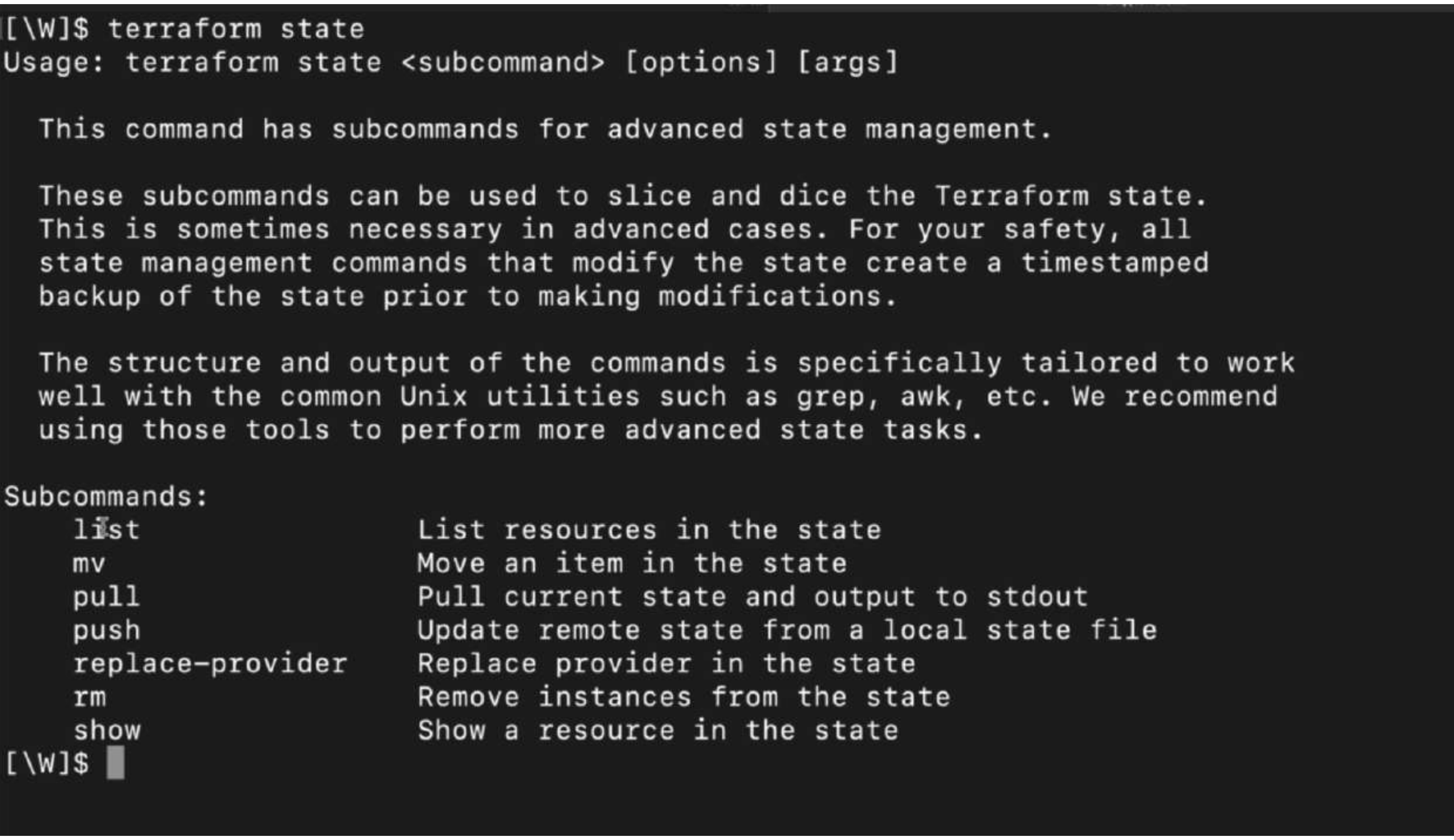


```
[\W]$ terraform destroy
```

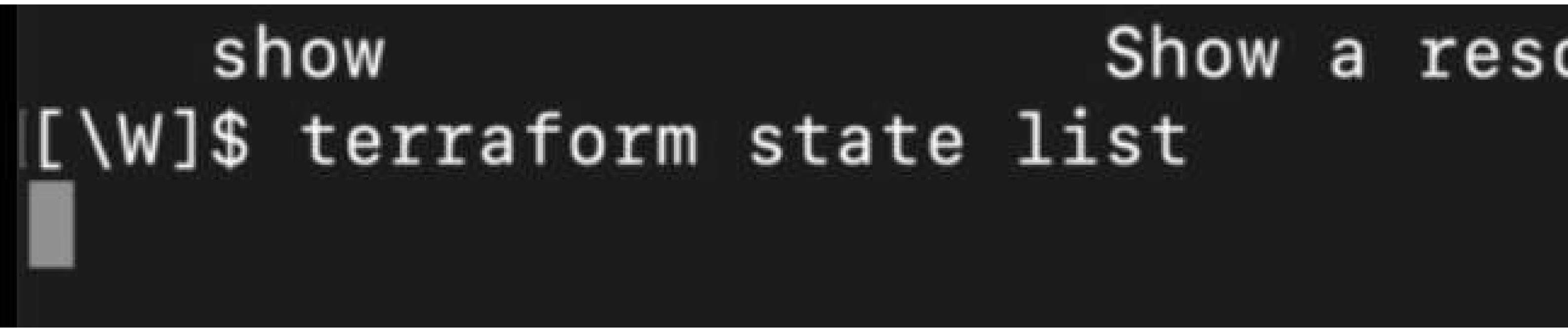


```
et" "dev-subnet-1" {
pc.development-vpc.id
10.0.10.0/24"
one =

net-1-
```

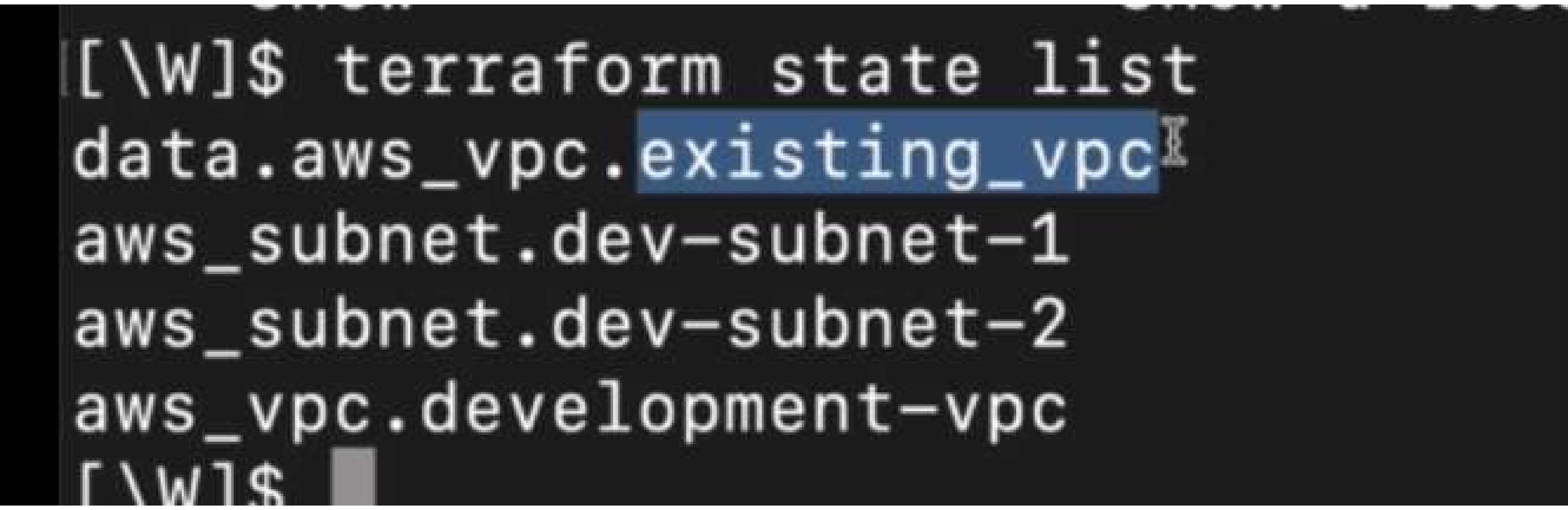
you don't have to know in which **order**
you need **to delete the resources**



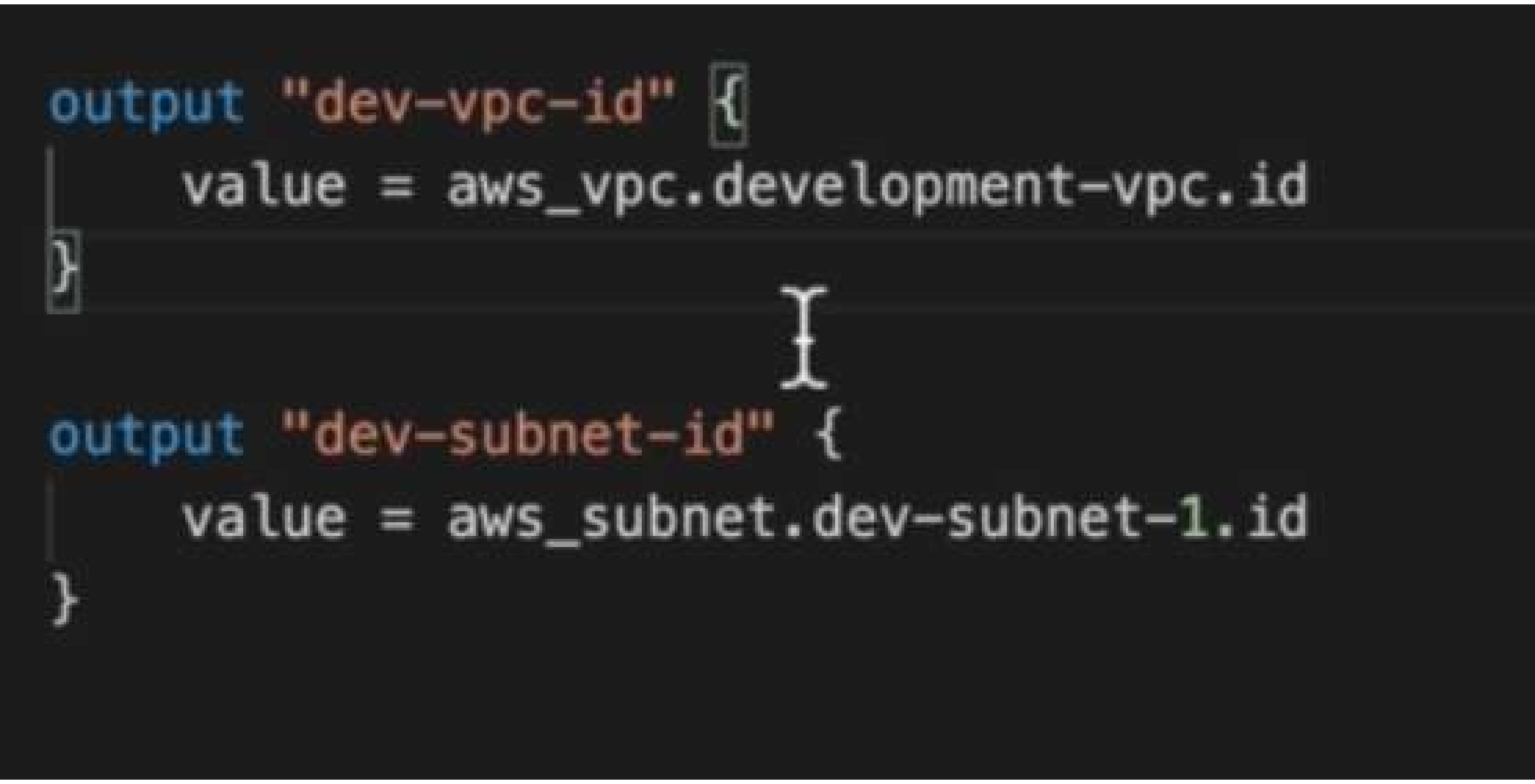
```
[\W]$ terraform state
Usage: terraform state <subcommand> [options] [args]

  This command has subcommands for advanced state management.

  These subcommands can be used to slice and dice the Terraform state.
  This is sometimes necessary in advanced cases. For your safety, all
  state management commands that modify the state create a timestamped
  backup of the state prior to making modifications.

  The structure and output of the commands is specifically tailored to work
  well with the common Unix utilities such as grep, awk, etc. We recommend
  using those tools to perform more advanced state tasks.

Subcommands:
    list                List resources in the state
    mv                  Move an item in the state
    pull                Pull current state and output to stdout
    push                Update remote state from a local state file
    replace-provider    Replace provider in the state
    rm                  Remove instances from the state
    show                Show a resource in the state
[\W]$ 
```



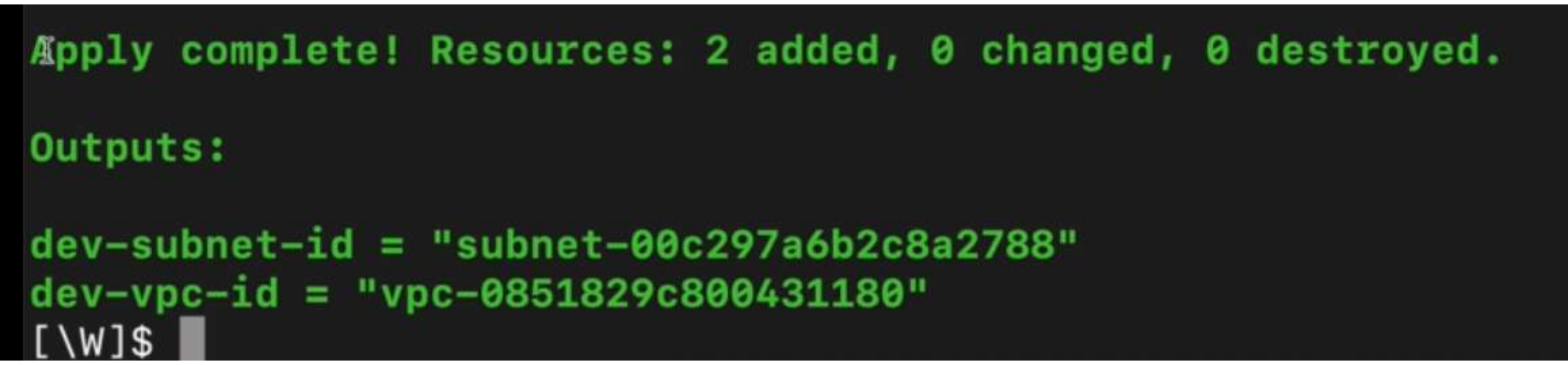
```
    show                    Show a reso
[\W]$ terraform state list
```



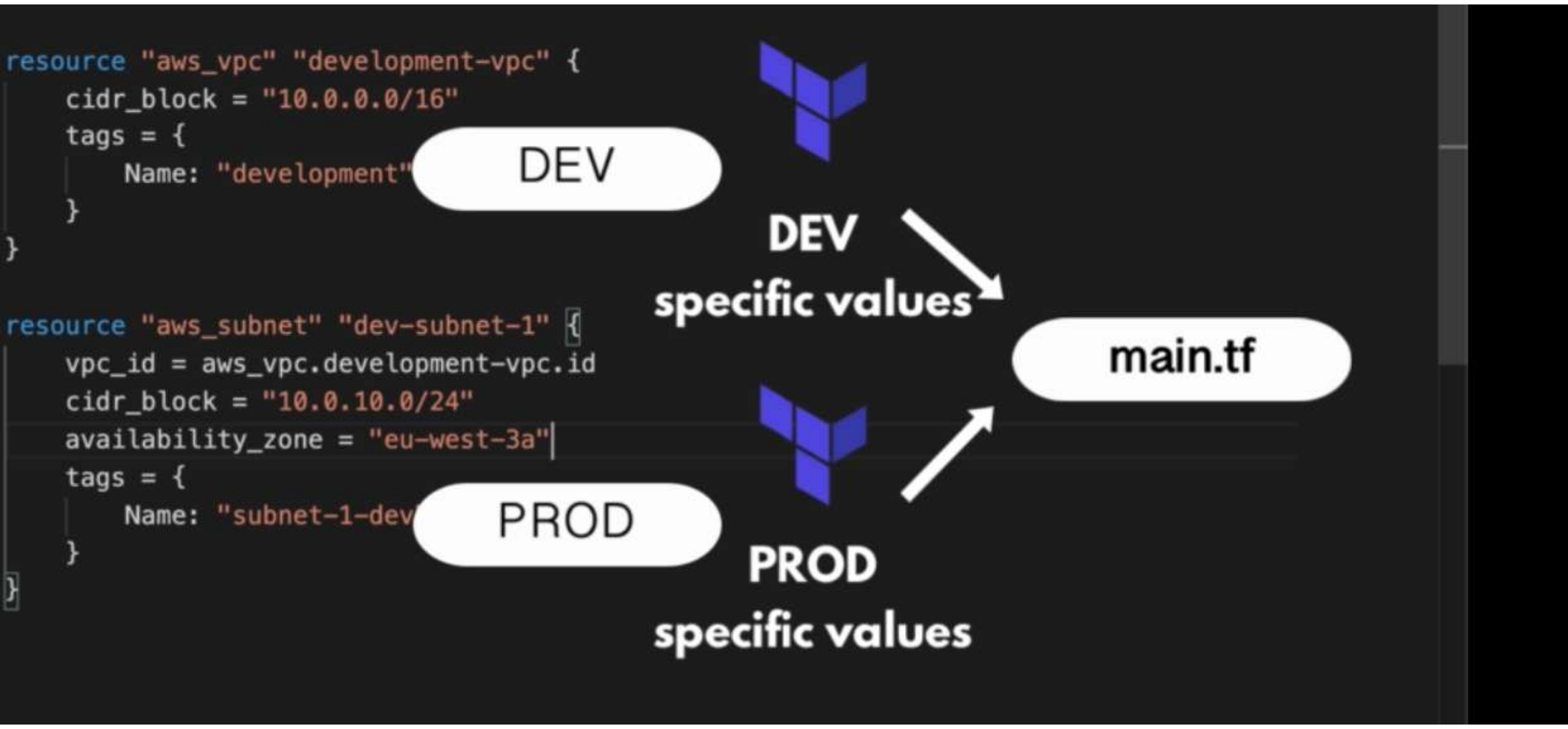
```
[\W]$ terraform state list
data.aws_vpc.existing_vpc
aws_subnet.dev-subnet-1
aws_subnet.dev-subnet-2
aws_vpc.development-vpc
[\W]$
```



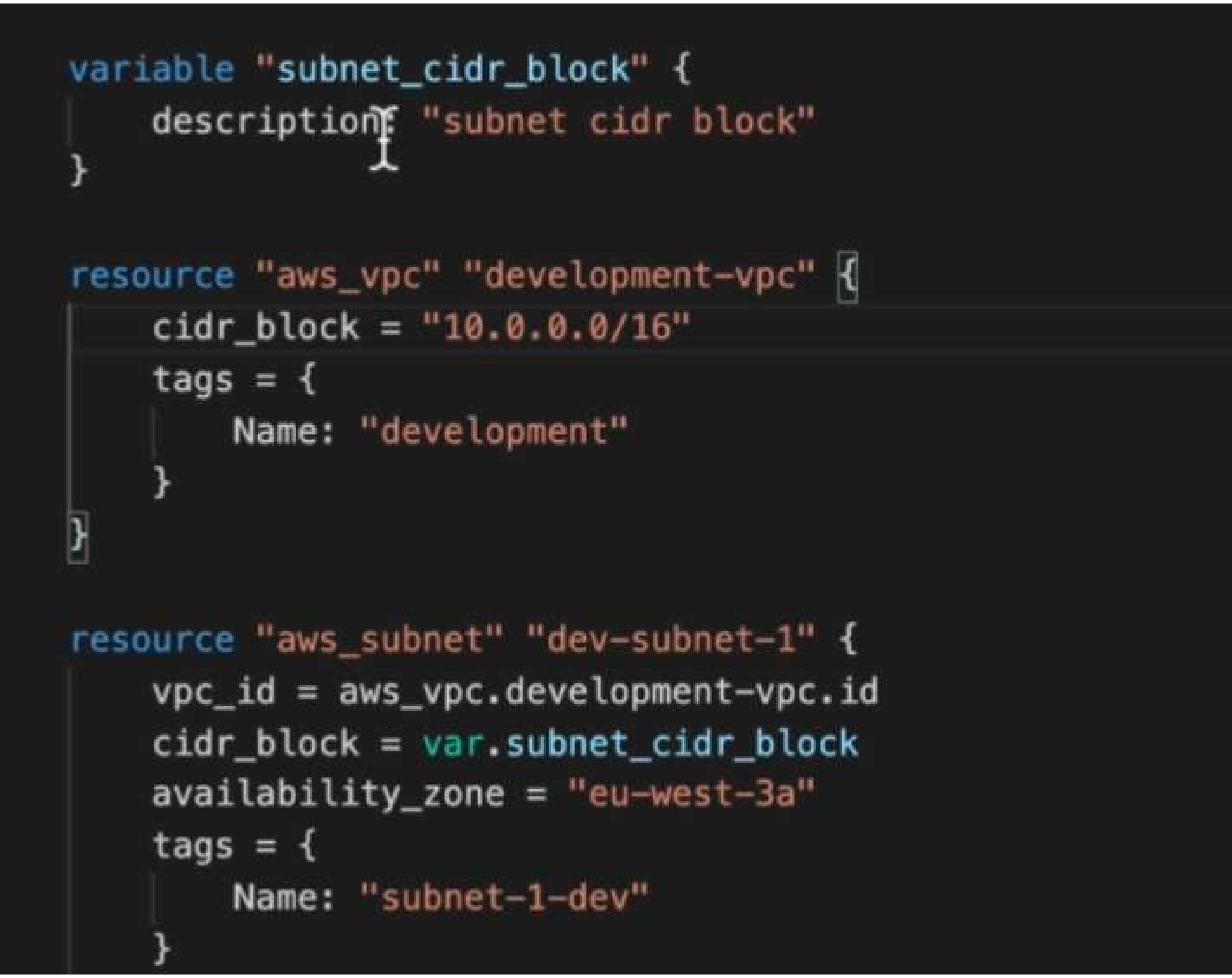
```
output "dev-vpc-id" {
    value = aws_vpc.development-vpc.id
}

output "dev-subnet-id" {
    value = aws_subnet.dev-subnet-1.id
}
```



```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

dev-subnet-id = "subnet-00c297a6b2c8a2788"
dev-vpc-id = "vpc-0851829c800431180"
[\W]$
```

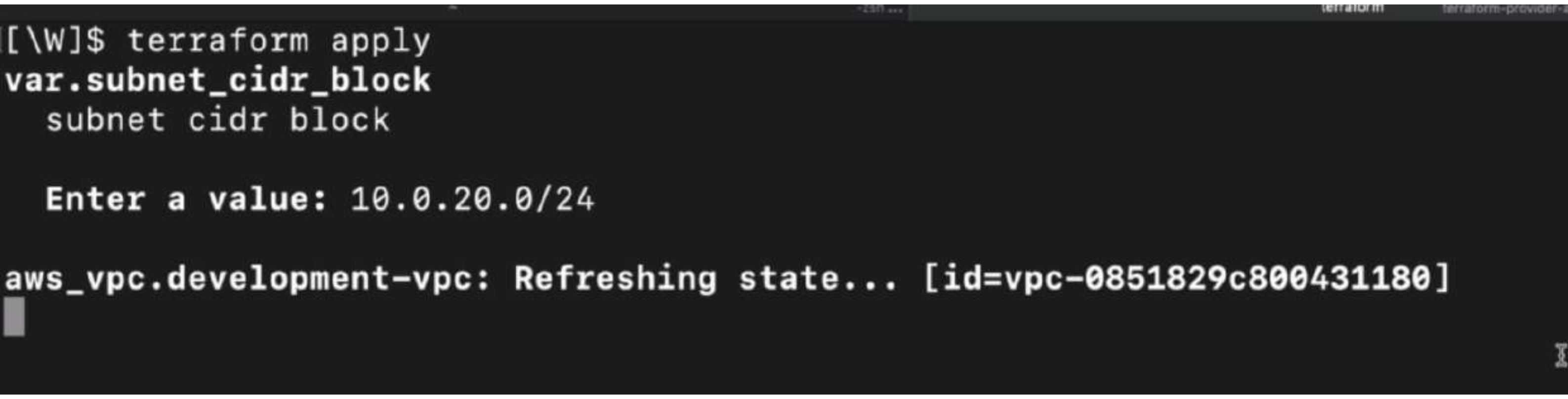


```
resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = "10.0.10.0/24"
    availability_zone = "eu-west-3a"
    tags = {
        Name: "subnet-1-dev
    }
}
```

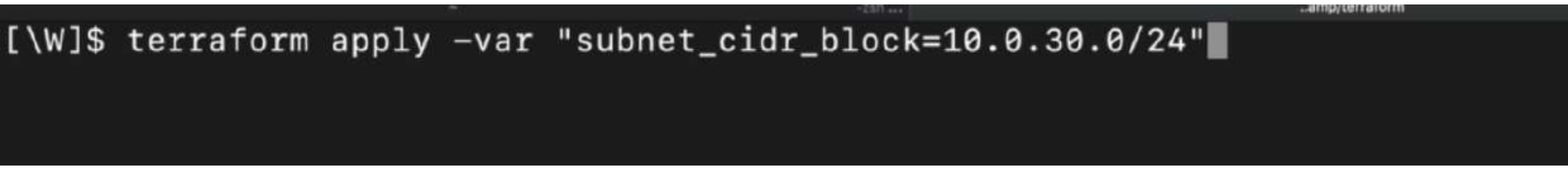


```
variable "subnet_cidr_block" {
    description: "subnet cidr block"
}

resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = var.subnet_cidr_block
    availability_zone = "eu-west-3a"
    tags = {
        Name: "subnet-1-dev"
    }
```

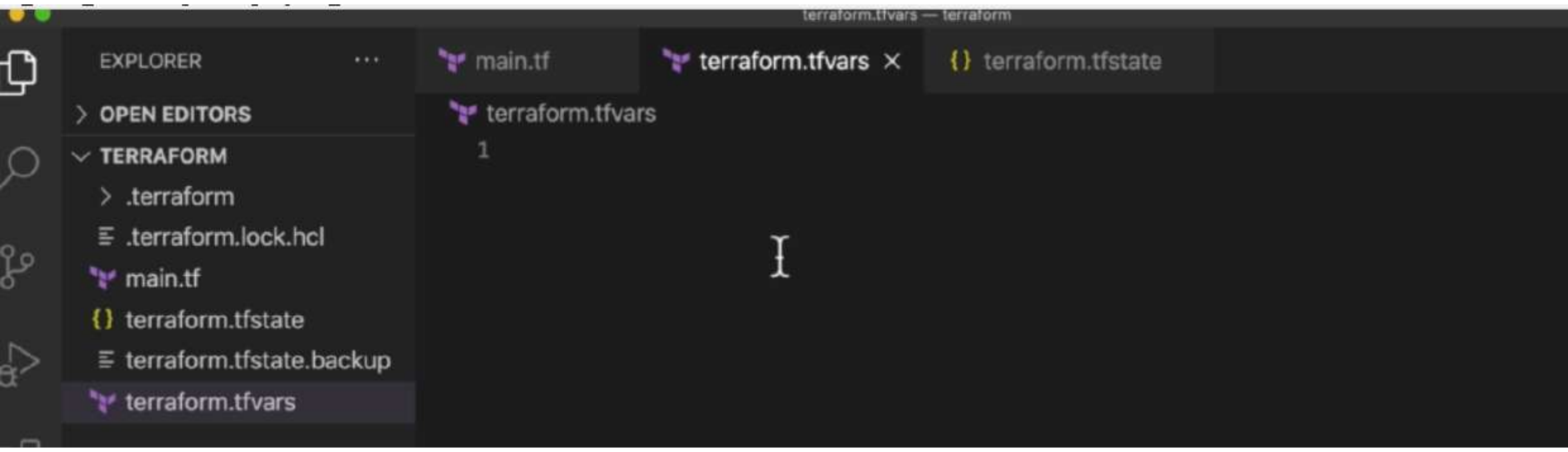
3 ways to pass value to the input variable

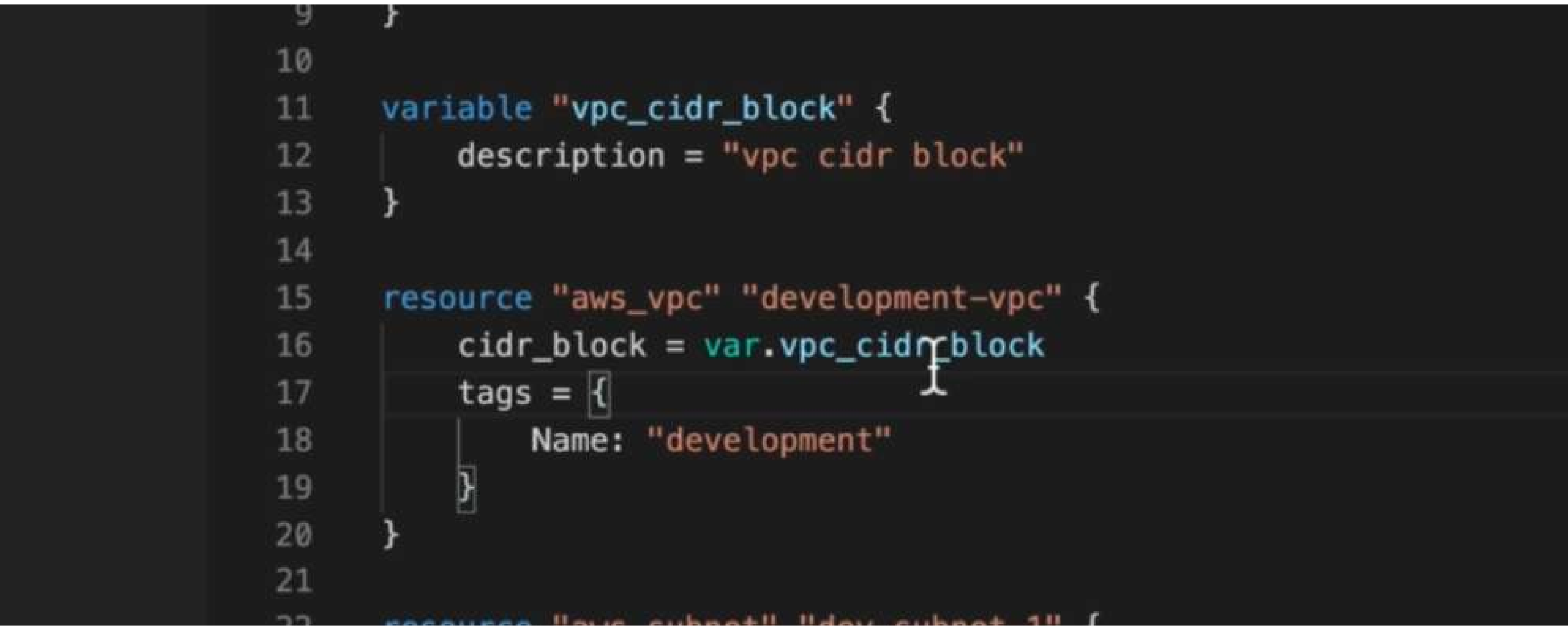


```
[\W]$ terraform apply
var.subnet_cidr_block
  subnet cidr block

  Enter a value: 10.0.20.0/24

aws_vpc.development-vpc: Refreshing state... [id=vpc-0851829c800431180]
```

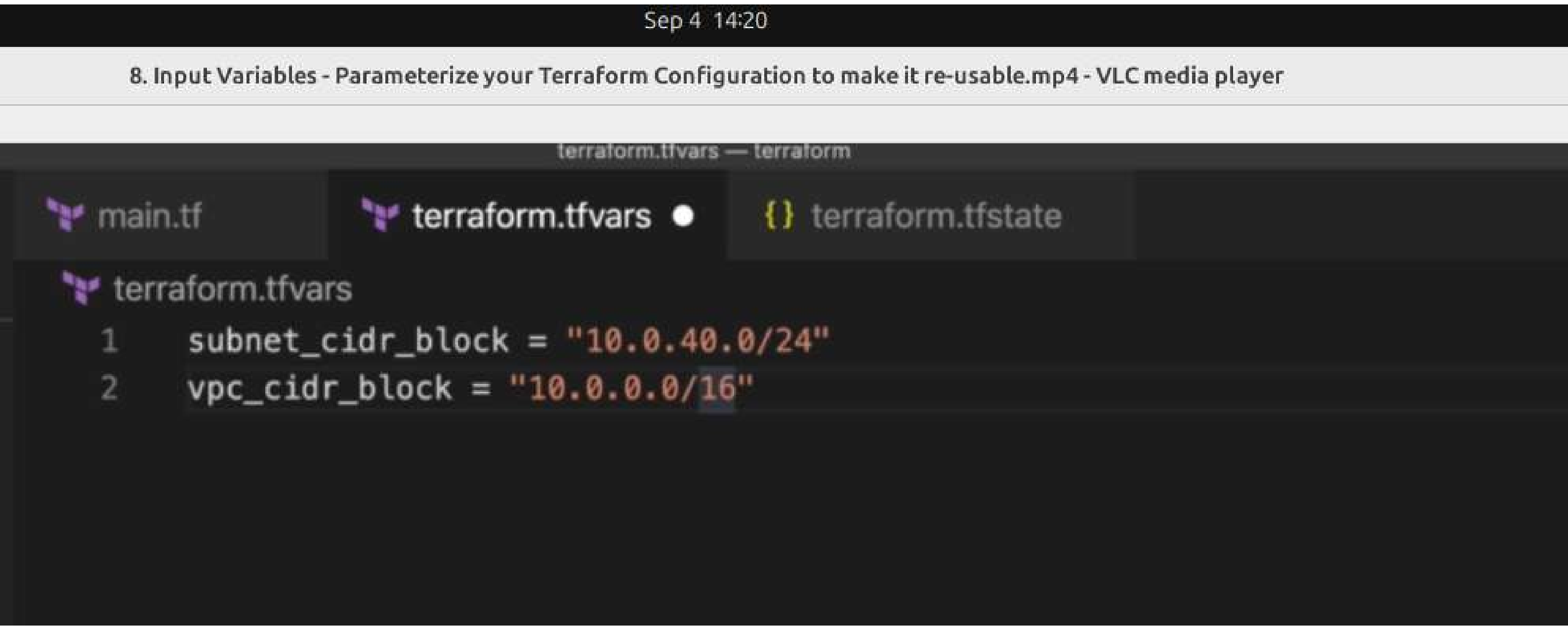


```
[\W]$ terraform apply -var "subnet_cidr_block=10.0.30.0/24"
```


terraform.tfvars — terraform
EXPLORER
...
OPEN EDITORS
TERRAFORM
.terraform
.terraform.lock.hcl
main.tf
terraform.tfstate
terraform.tfstate.backup
terraform.tfvars
main.tf
terraform.tfvars
terraform.tfstate
terraform.tfvars



```
}

variable "vpc_cidr_block" {
    description = "vpc cidr block"
}

resource "aws_vpc" "development-vpc" {
    cidr_block = var.vpc_cidr_block
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
```



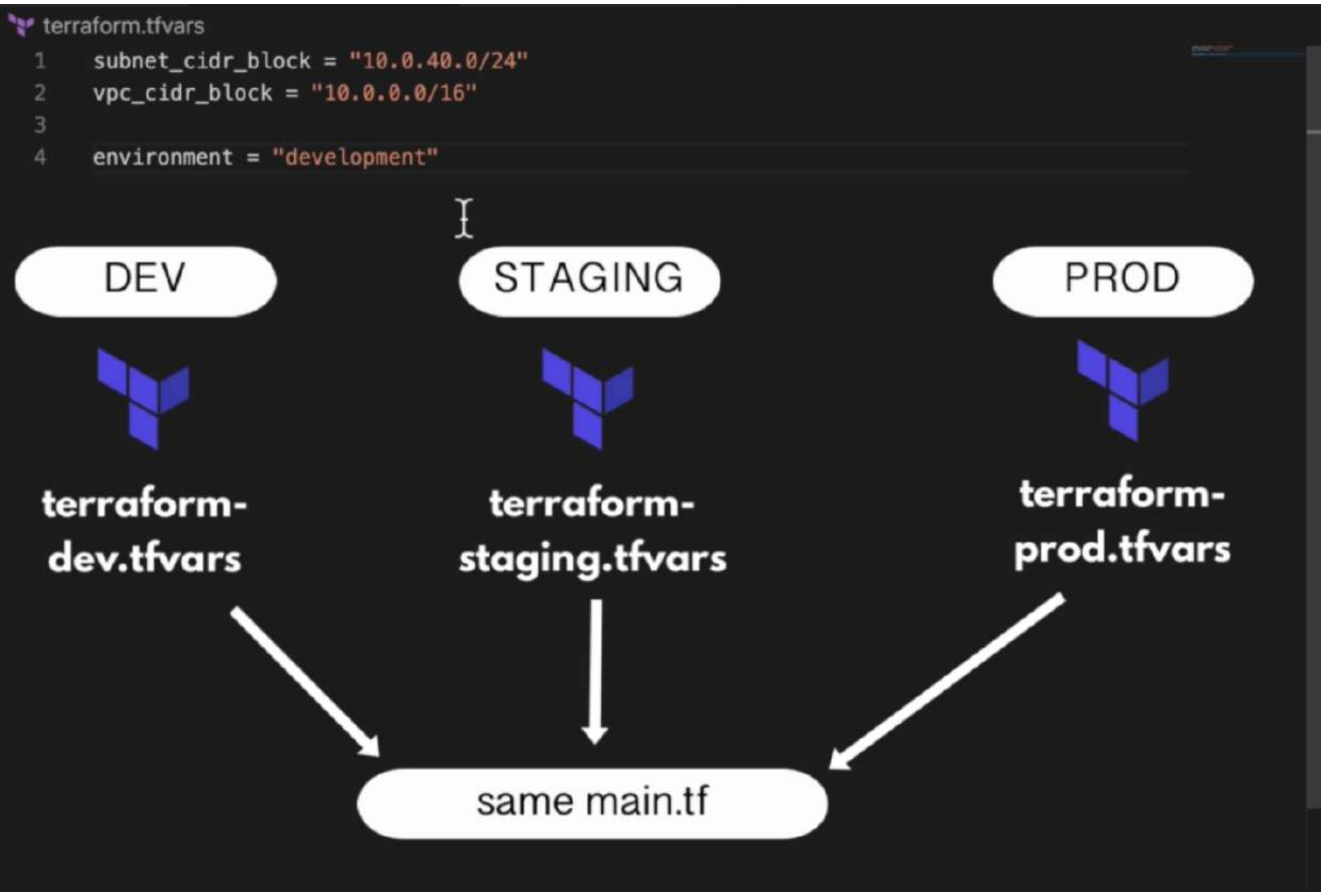
Sep 4 14:20

8. Input Variables - Parameterize your Terraform Configuration to make it re-usable.mp4 - VLC media player

terraform.tfvars — terraform

```
subnet_cidr_block = "10.0.40.0/24"
vpc_cidr_block = "10.0.0.0/16"
```

Use Case for Input Variables



terraform.tfvars

```
subnet_cidr_block = "10.0.40.0/24"
vpc_cidr_block = "10.0.0.0/16"

environment = "development"
```

DEV

terraform-dev.tfvars

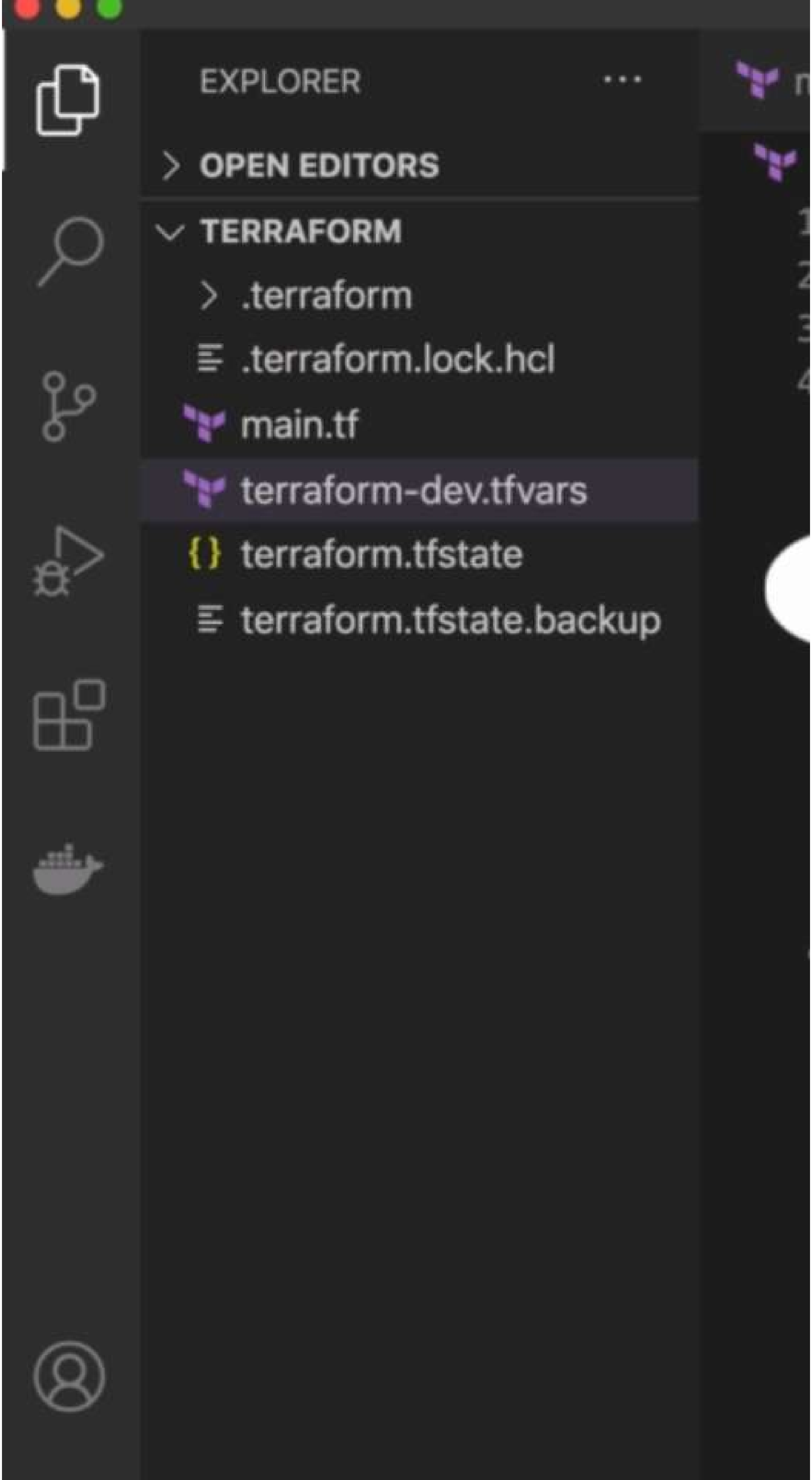
STAGING

terraform-staging.tfvars

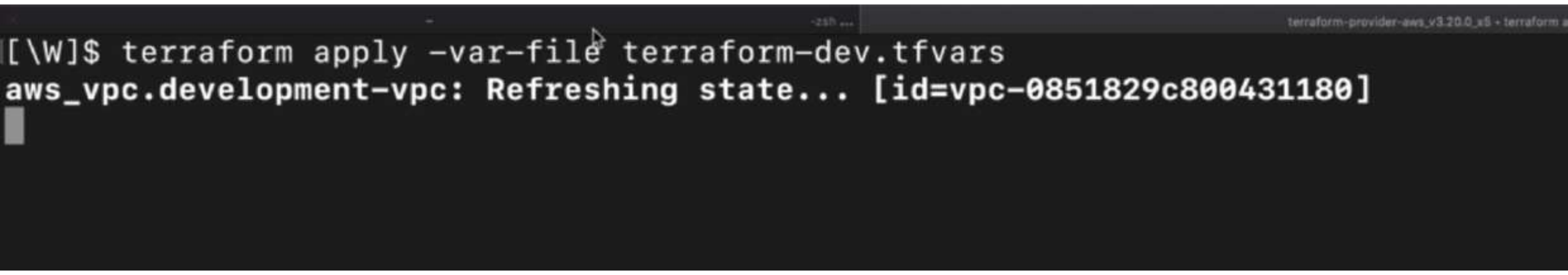
PROD

terraform-prod.tfvars

same main.tf

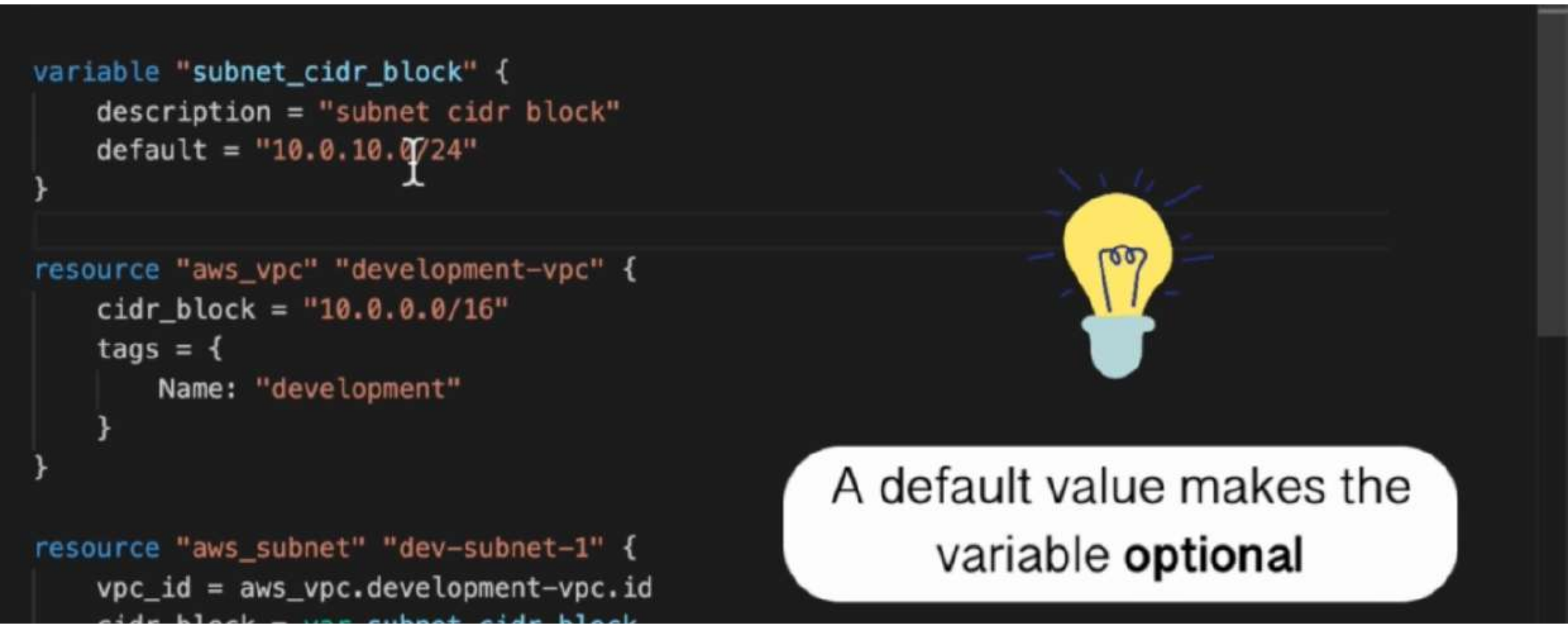


```
TERRAFORM
  .terraform
  .terraform.lock.hcl
  main.tf
  terraform-dev.tfvars
  terraform.tfstate
  terraform.tfstate.backup
```



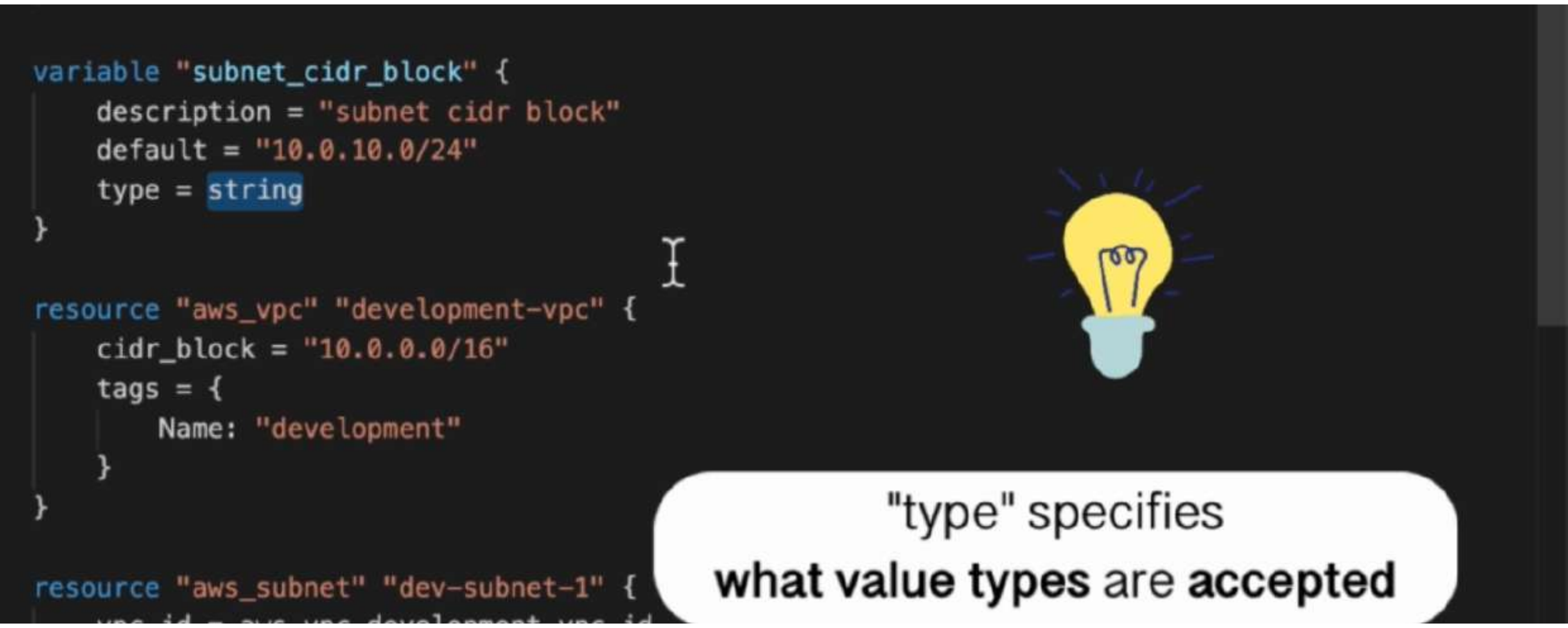
```
[\W]$ terraform apply -var-file terraform-dev.tfvars
aws_vpc.development-vpc: Refreshing state... [id=vpc-0851829c800431180]
```

Default Values



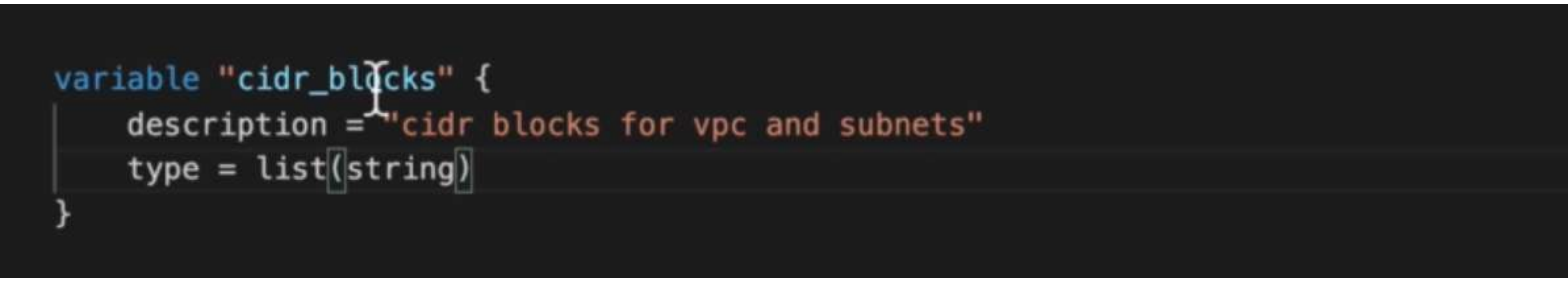
```
variable "subnet_cidr_block" {
    description = "subnet cidr block"
    default = "10.0.10.0/24"
}

resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = var.subnet_cidr_block
```

A default value makes the variable **optional**

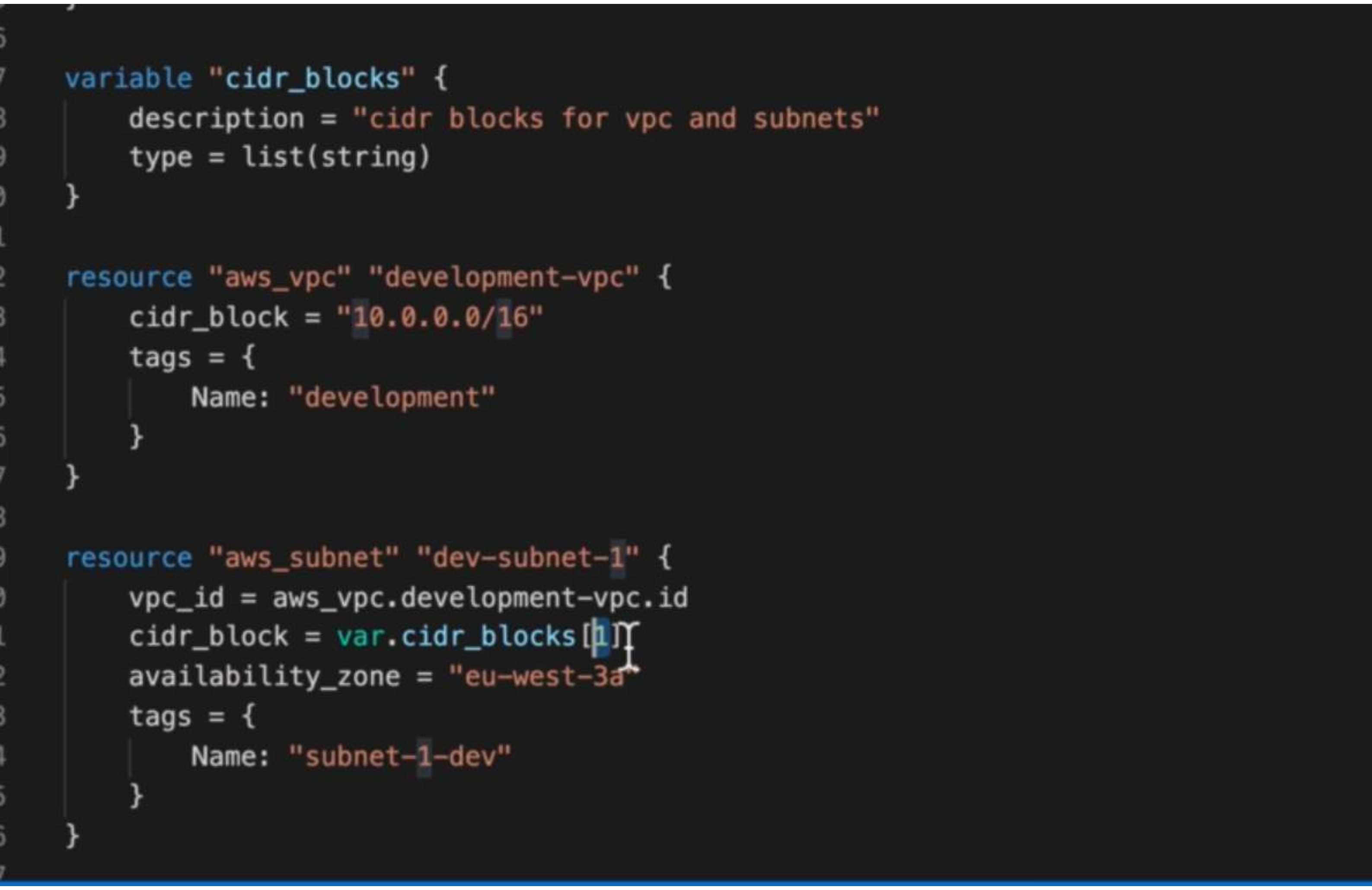


```
variable "subnet_cidr_block" {
    description = "subnet cidr block"
    default = "10.0.10.0/24"
    type = string
}

resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
```

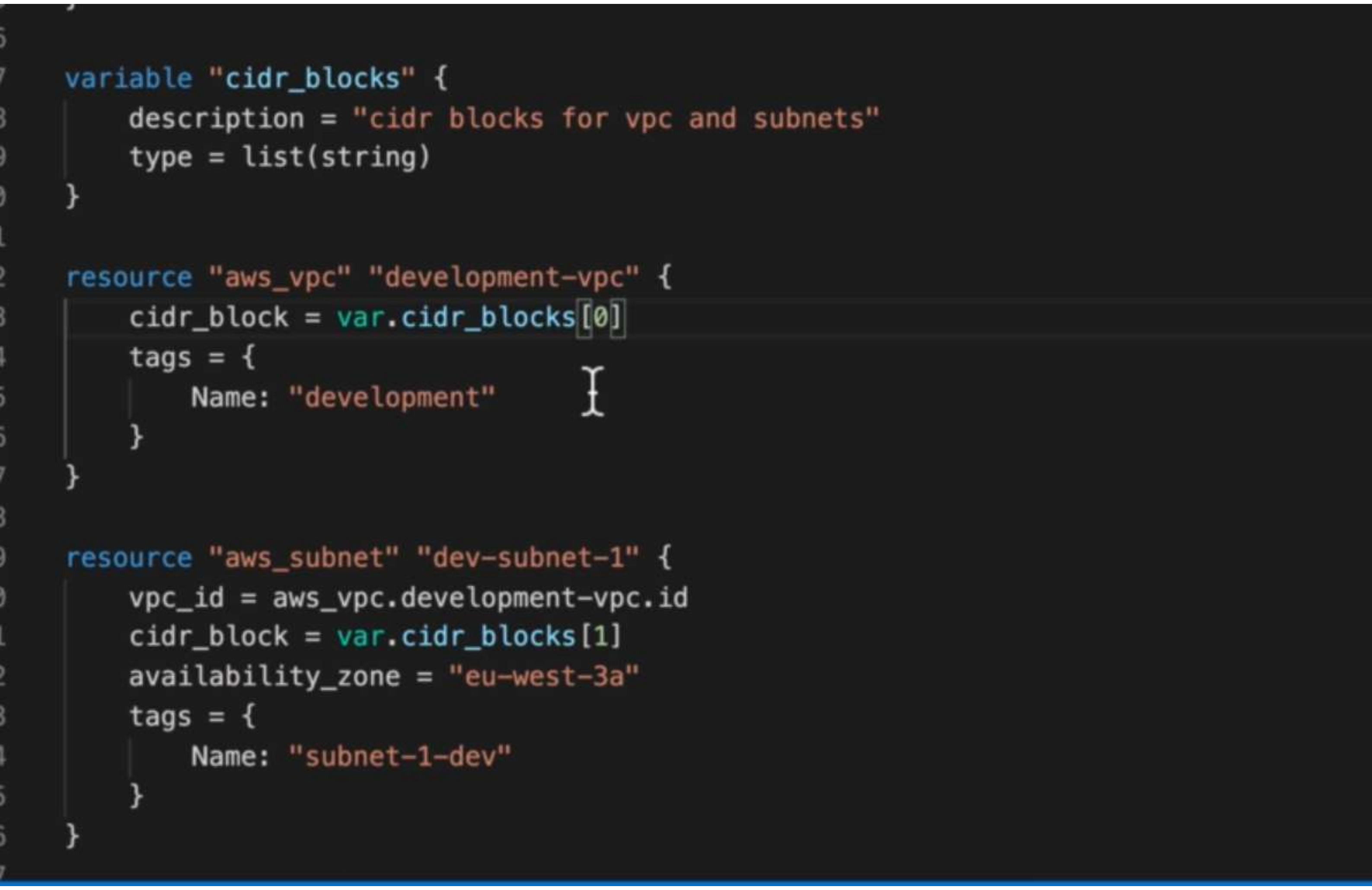
"type" specifies
what value types are **accepted**



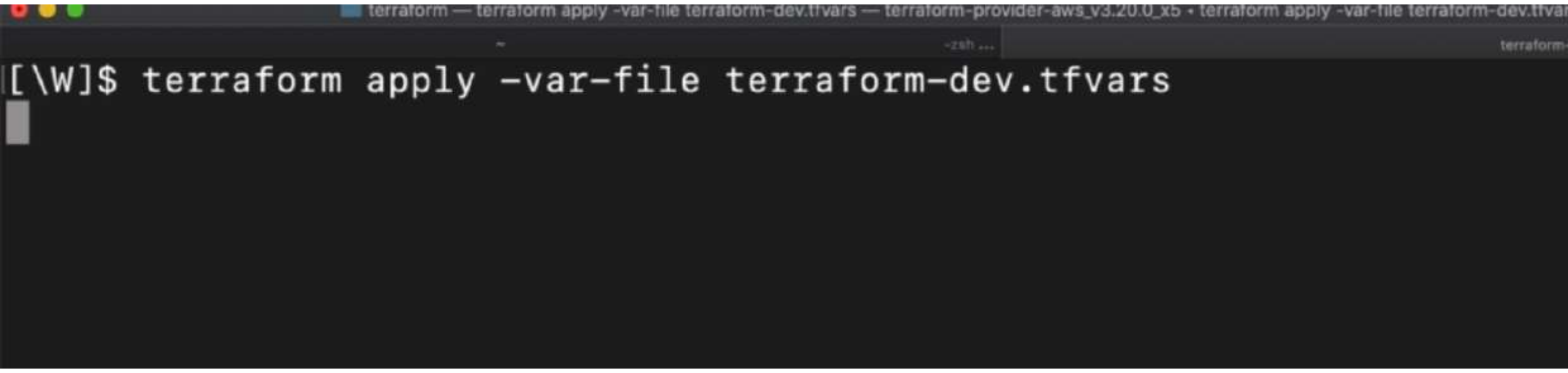
```
variable "cidr_blocks" {
    description = "cidr blocks for vpc and subnets"
    type = list(string)
}
```



```
variable "cidr_blocks" {
    description = "cidr blocks for vpc and subnets"
    type = list(string)
}

resource "aws_vpc" "development-vpc" {
    cidr_block = "10.0.0.0/16"
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = var.cidr_blocks[1]
    availability_zone = "eu-west-3a"
    tags = {
        Name: "subnet-1-dev"
    }
}
```

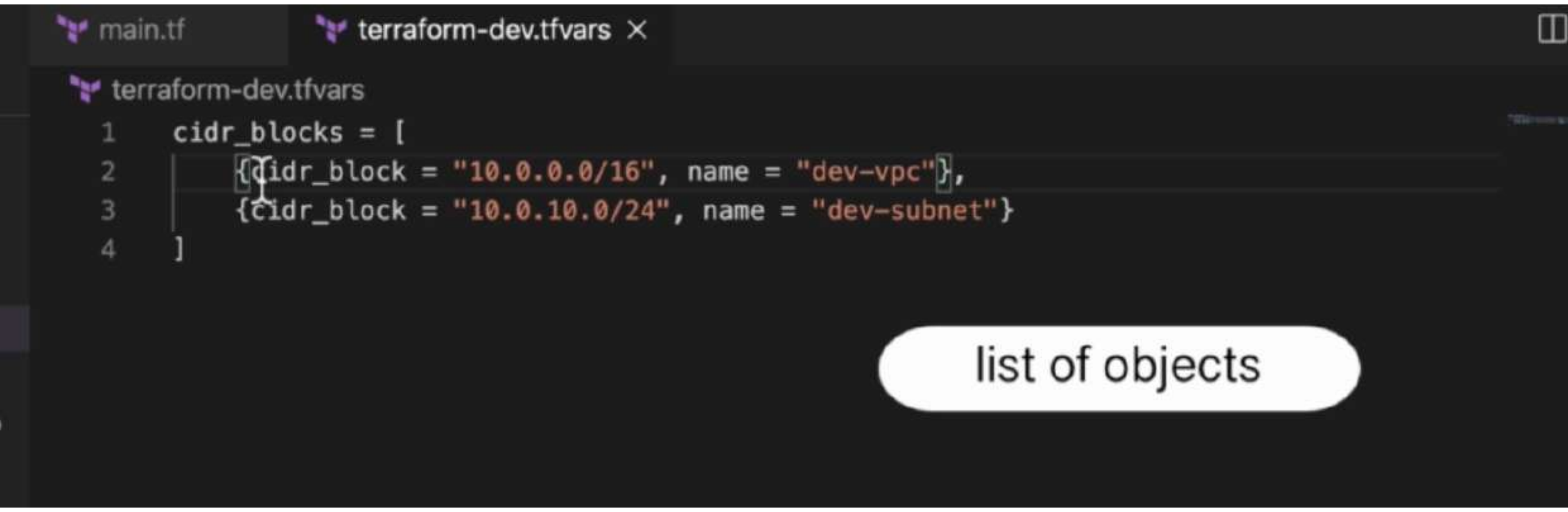


```
variable "cidr_blocks" {
    description = "cidr blocks for vpc and subnets"
    type = list(string)
}

resource "aws_vpc" "development-vpc" {
    cidr_block = var.cidr_blocks[0]
    tags = {
        Name: "development"
    }
}

resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = var.cidr_blocks[1]
    availability_zone = "eu-west-3a"
    tags = {
        Name: "subnet-1-dev"
    }
}
```



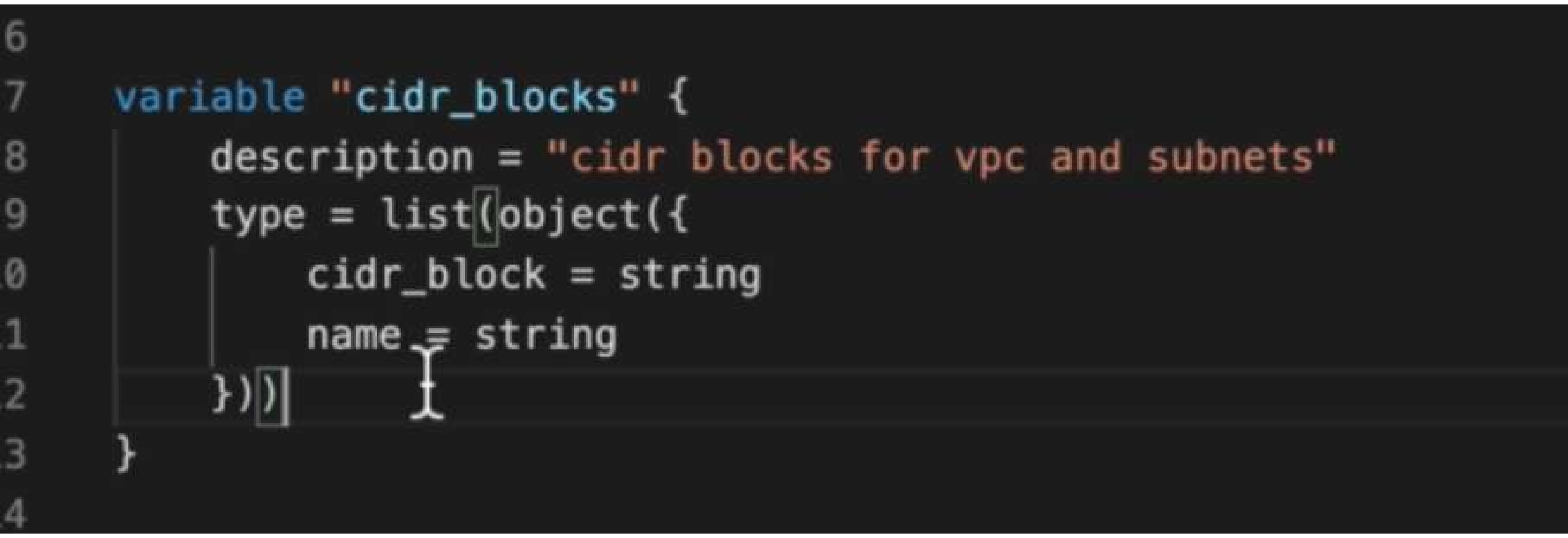
terraform — terraform apply -var-file terraform-dev.tfvars — terraform-provider-aws_v3.20.0_x5 • terraform apply -var-file terraform-dev.tfvai
~
-zsh ...
terraform-

```
[\W]$ terraform apply -var-file terraform-dev.tfvars
```

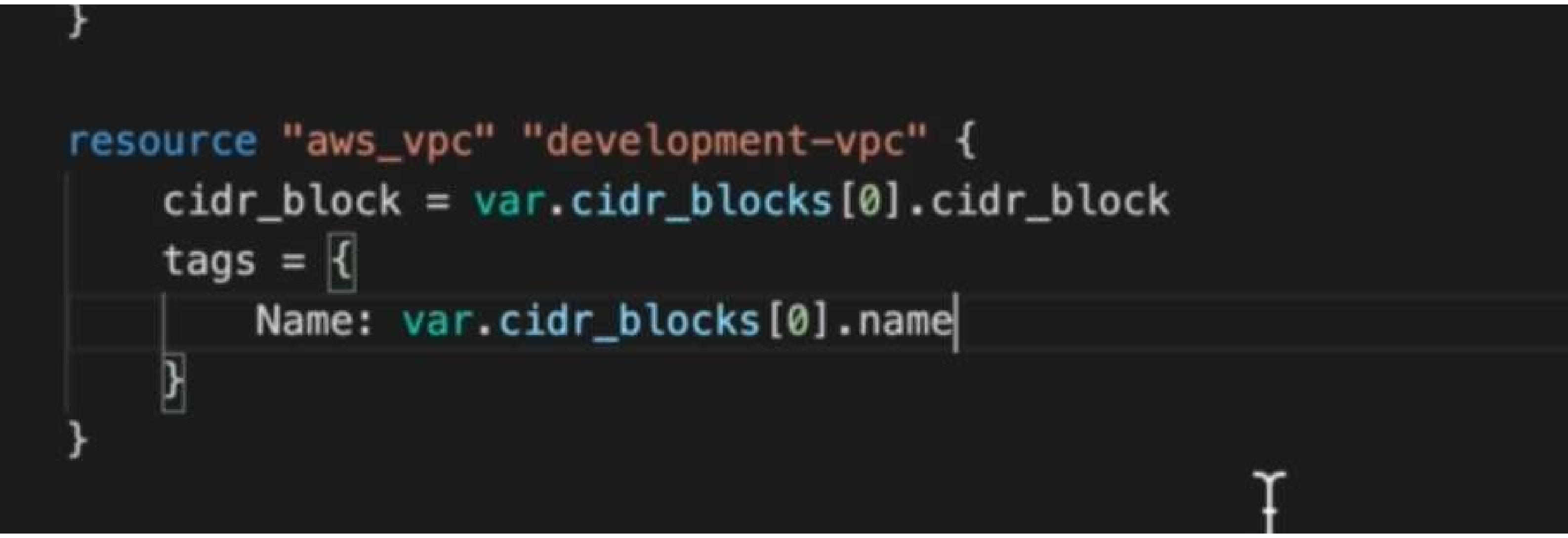


```
cidr_blocks = [
    {cidr_block = "10.0.0.0/16", name = "dev-vpc"},
    {cidr_block = "10.0.10.0/24", name = "dev-subnet"}
]
```

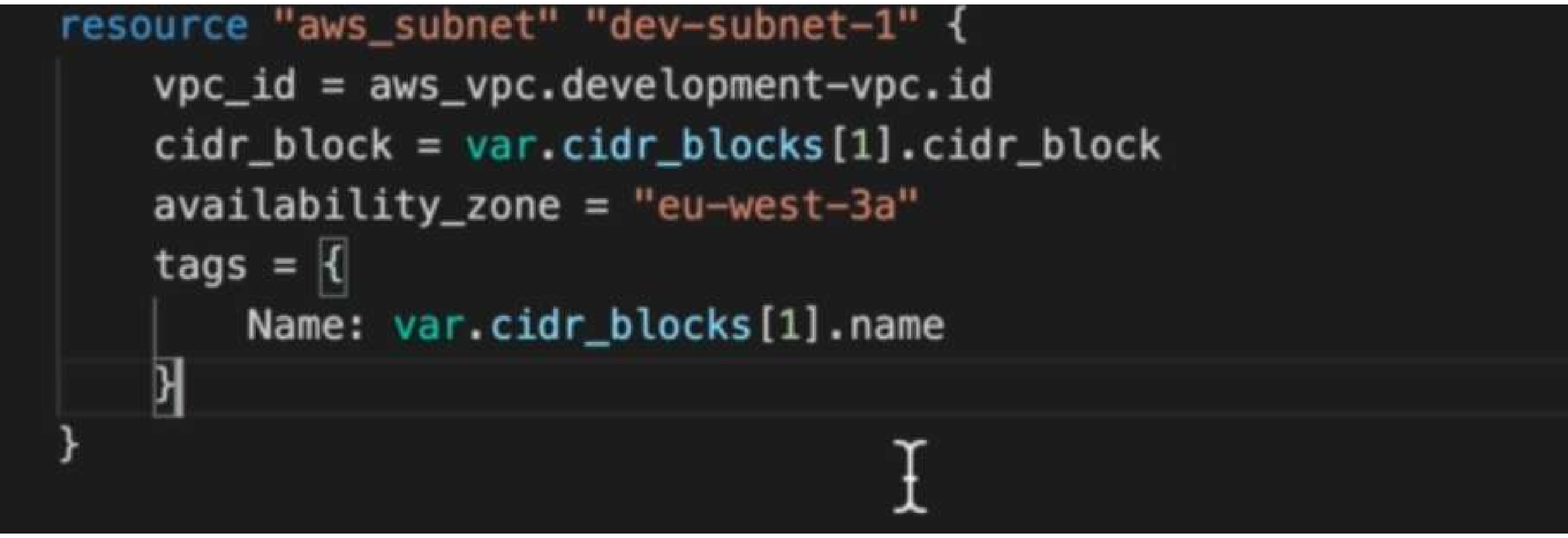
list of objects



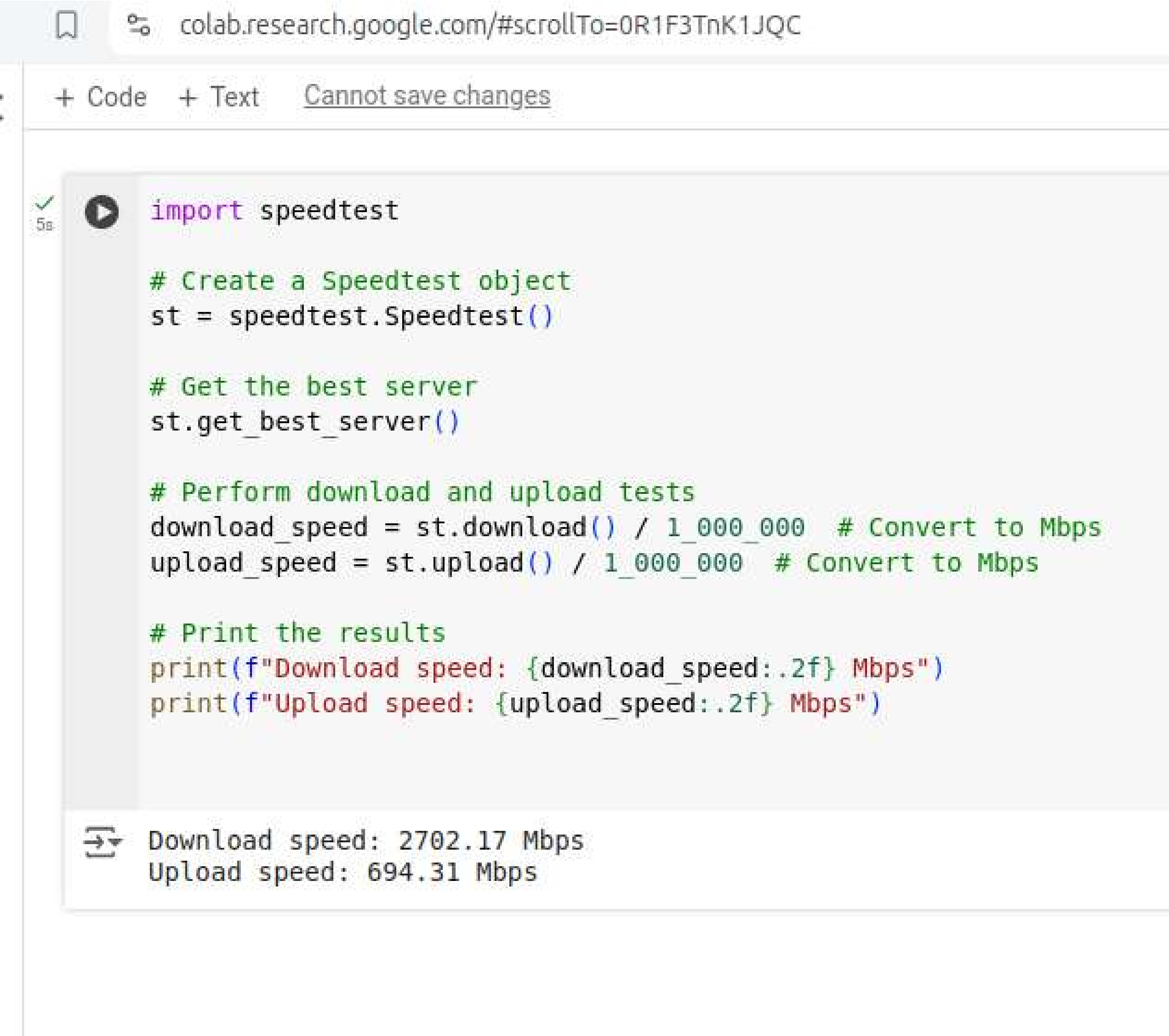
```
variable "cidr_blocks" {
    description = "cidr blocks for vpc and subnets"
    type = list(object({
        cidr_block = string
        name = string
    }))
}
```



```
}

resource "aws_vpc" "development-vpc" {
    cidr_block = var.cidr_blocks[0].cidr_block
    tags = {
        Name: var.cidr_blocks[0].name
    }
}
```

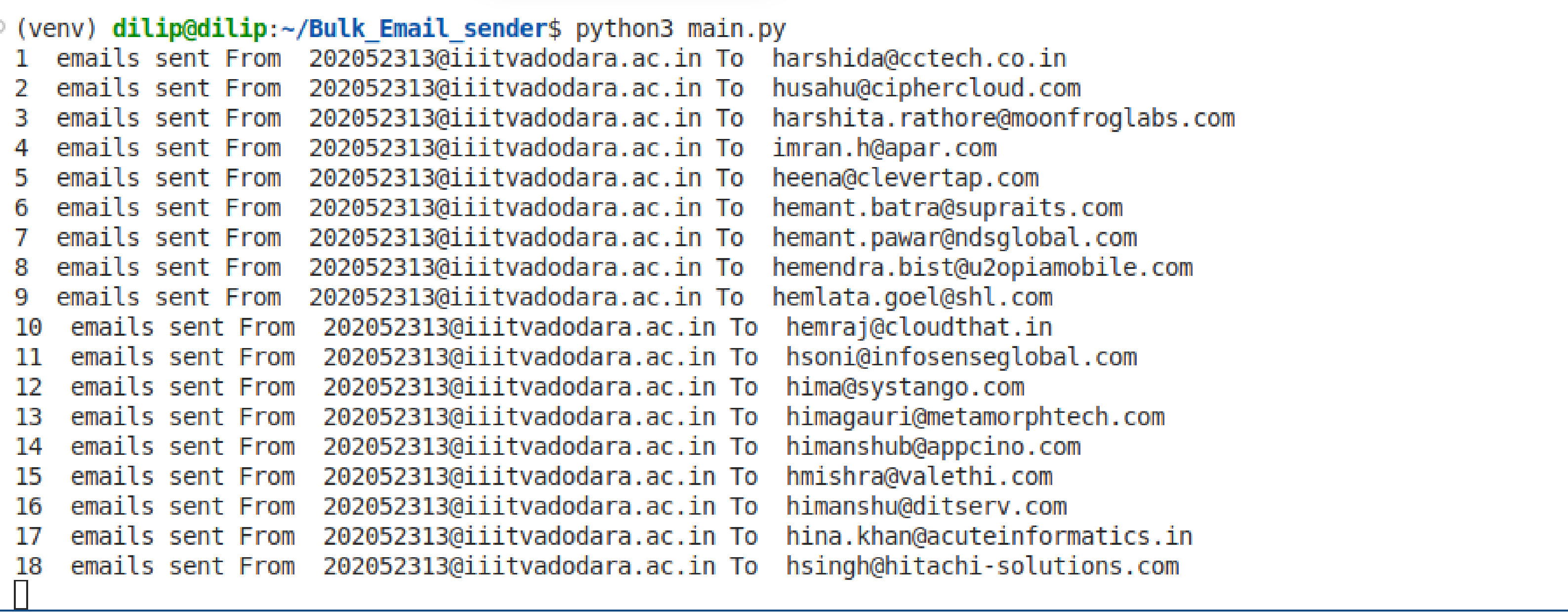


```
resource "aws_subnet" "dev-subnet-1" {
    vpc_id = aws_vpc.development-vpc.id
    cidr_block = var.cidr_blocks[1].cidr_block
    availability_zone = "eu-west-3a"
    tags = {
        Name: var.cidr_blocks[1].name
    }
}
```

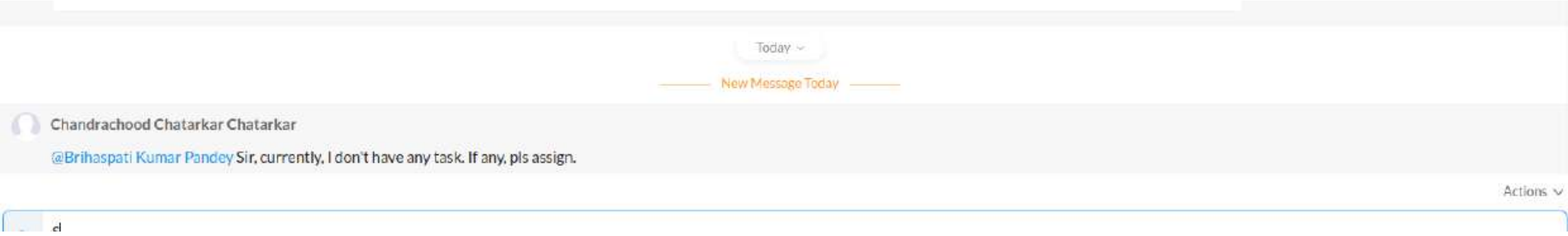


```python
import speedtest

# Create a Speedtest object
st = speedtest.Speedtest()

# Get the best server
st.get_best_server()

# Perform download and upload tests
download_speed = st.download() / 1_000_000  # Convert to Mbps
upload_speed = st.upload() / 1_000_000  # Convert to Mbps

# Print the results
print(f"Download speed: {download_speed:.2f} Mbps")
print(f"Upload speed: {upload_speed:.2f} Mbps")
```

```
Download speed: 2702.17 Mbps
Upload speed: 694.31 Mbps
```



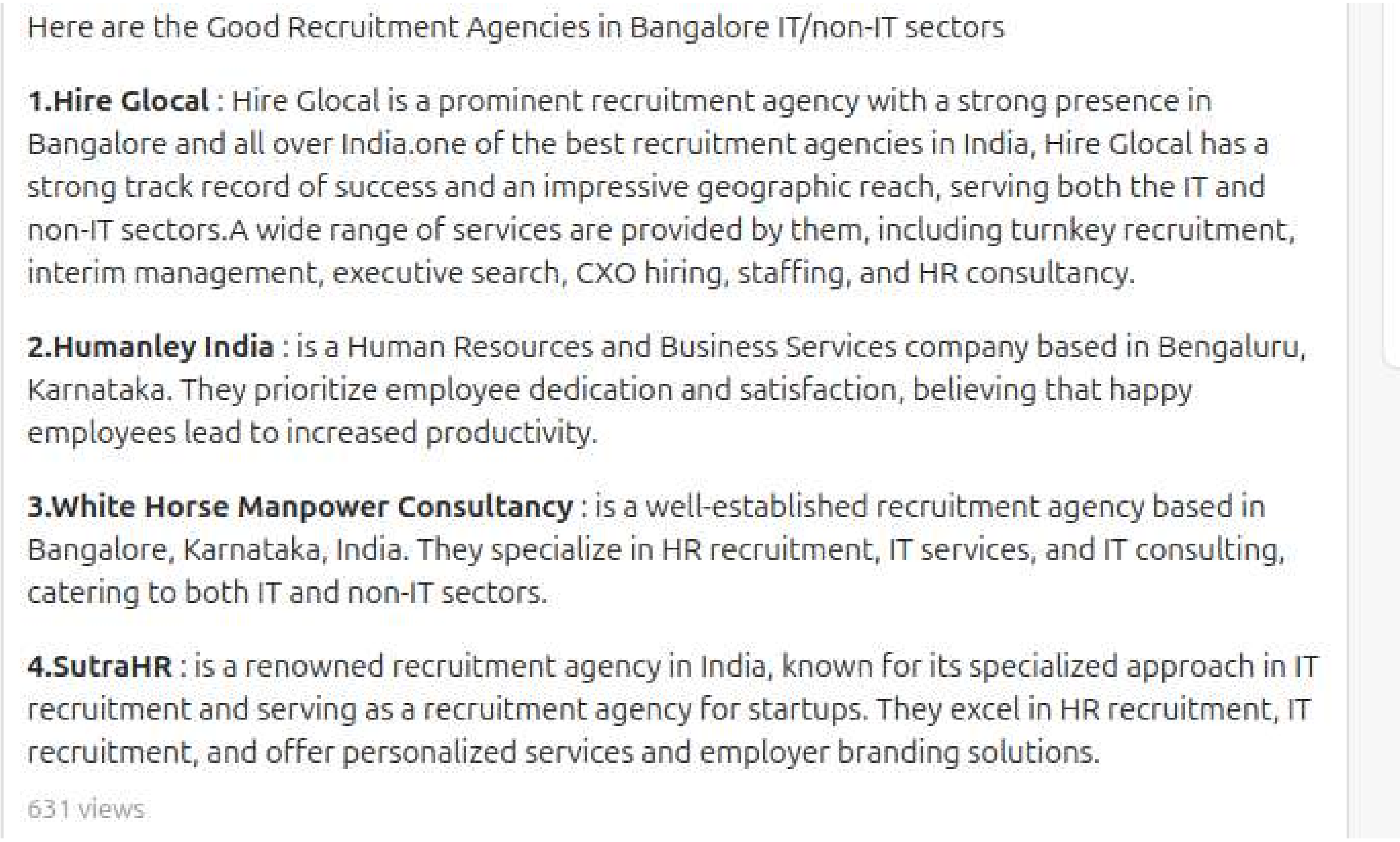
```
(venv) dilip@dilip:~/Bulk_Email_sender$ python3 main.py
1  emails sent From  202052313@iiitvadodara.ac.in To  harshida@cctech.co.in
2  emails sent From  202052313@iiitvadodara.ac.in To  husahu@ciphercloud.com
3  emails sent From  202052313@iiitvadodara.ac.in To  harshita.rathore@moonfroglabs.com
4  emails sent From  202052313@iiitvadodara.ac.in To  imran.h@apar.com
5  emails sent From  202052313@iiitvadodara.ac.in To  heena@clevertap.com
6  emails sent From  202052313@iiitvadodara.ac.in To  hemant.batra@supraits.com
7  emails sent From  202052313@iiitvadodara.ac.in To  hemant.pawar@ndsglobal.com
8  emails sent From  202052313@iiitvadodara.ac.in To  hemendra.bist@u2opiamobile.com
9  emails sent From  202052313@iiitvadodara.ac.in To  hemlata.goel@shl.com
10  emails sent From  202052313@iiitvadodara.ac.in To  hemraj@cloudthat.in
11  emails sent From  202052313@iiitvadodara.ac.in To  hsoni@infosenseglobal.com
12  emails sent From  202052313@iiitvadodara.ac.in To  hima@systango.com
13  emails sent From  202052313@iiitvadodara.ac.in To  himagauri@metamorphtech.com
14  emails sent From  202052313@iiitvadodara.ac.in To  himanshub@appcino.com
15  emails sent From  202052313@iiitvadodara.ac.in To  hmishra@valethi.com
16  emails sent From  202052313@iiitvadodara.ac.in To  himanshu@ditserv.com
17  emails sent From  202052313@iiitvadodara.ac.in To  hina.khan@acuteinformatics.in
18  emails sent From  202052313@iiitvadodara.ac.in To  hsingh@hitachi-solutions.com
```



Chandrachood Chatarkar Chatarkar

@Brihaspati Kumar Pandey Sir, currently, I don't have any task. If any, pls assign.

sl



Here are the Good Recruitment Agencies in Bangalore IT/non-IT sectors

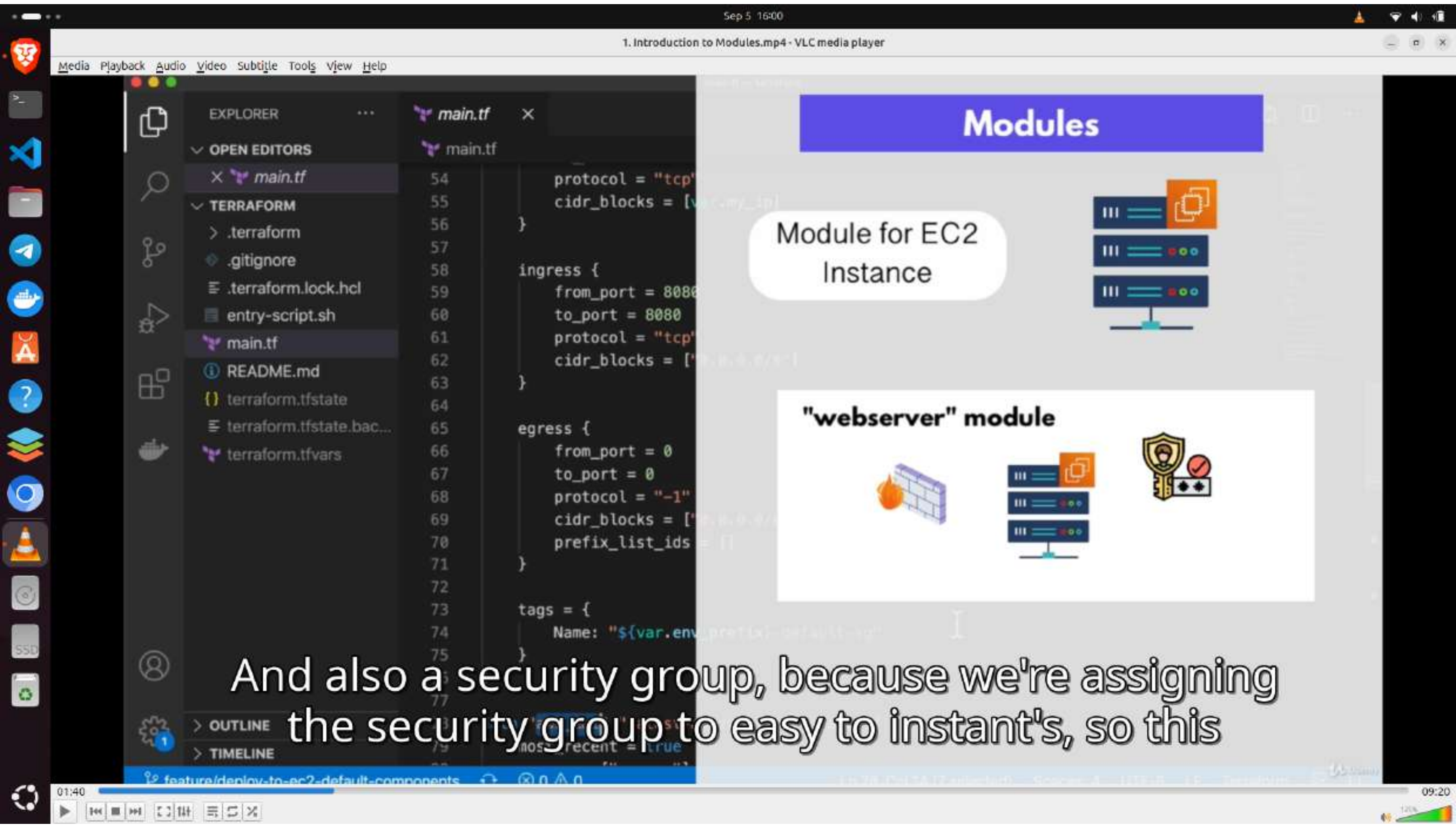
1.Hire Glocal : Hire Glocal is a prominent recruitment agency with a strong presence in Bangalore and all over India.one of the best recruitment agencies in India, Hire Glocal has a strong track record of success and an impressive geographic reach, serving both the IT and non-IT sectors.A wide range of services are provided by them, including turnkey recruitment, interim management, executive search, CXO hiring, staffing, and HR consultancy.

2.Humanley India : is a Human Resources and Business Services company based in Bengaluru, Karnataka. They prioritize employee dedication and satisfaction, believing that happy employees lead to increased productivity.

3.White Horse Manpower Consultancy : is a well-established recruitment agency based in Bangalore, Karnataka, India. They specialize in HR recruitment, IT services, and IT consulting, catering to both IT and non-IT sectors.

4.SutraHR : is a renowned recruitment agency in India, known for its specialized approach in IT recruitment and serving as a recruitment agency for startups. They excel in HR recruitment, IT recruitment, and offer personalized services and employer branding solutions.

631 views



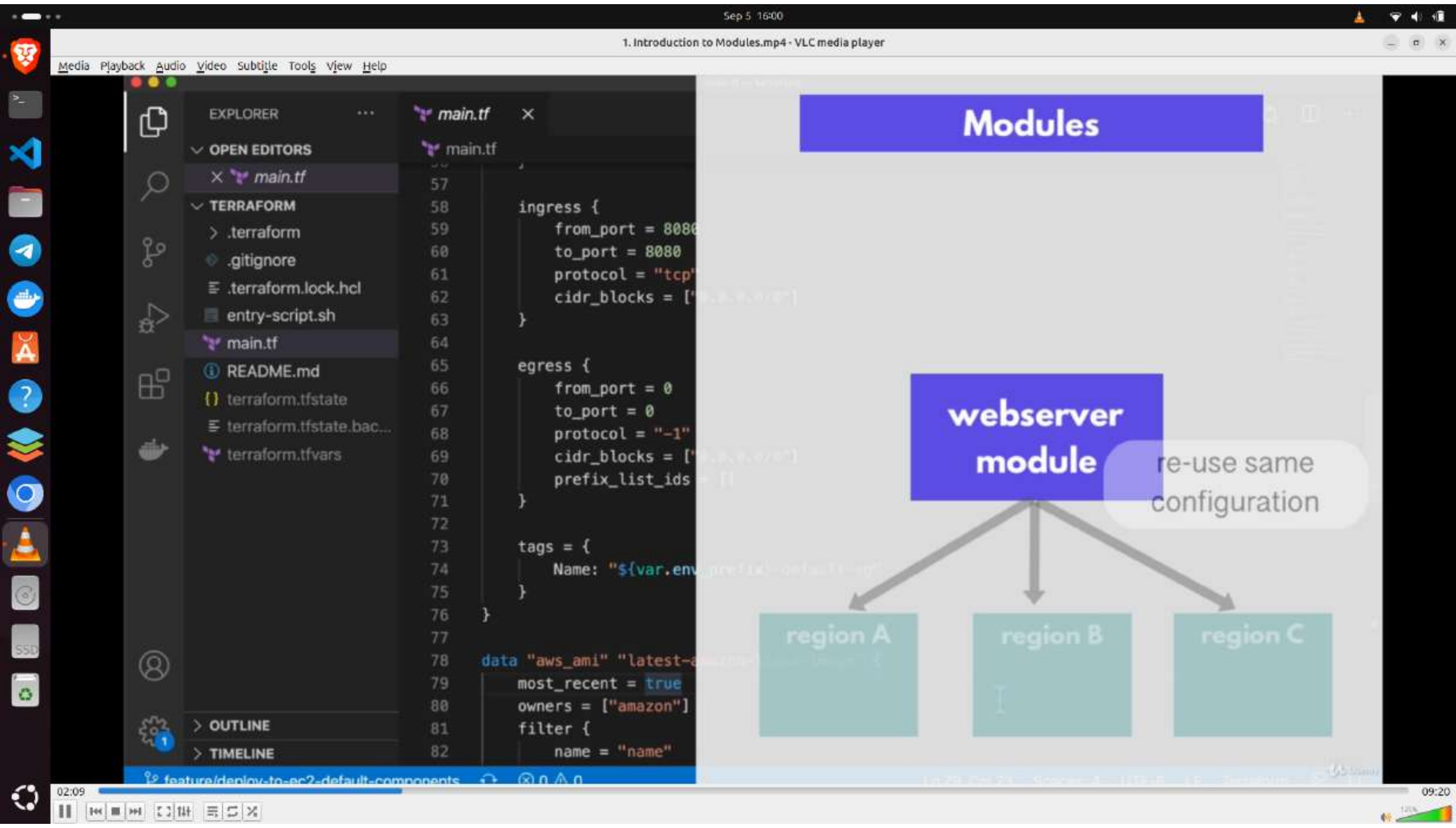
```
    protocol = "tcp"
    cidr_blocks = [v
  }

  ingress {
    from_port = 808
    to_port = 8080
    protocol = "tcp"
    cidr_blocks = ["
  }

  egress {
    from_port = 0
    to_port = 0
    protocol = "-1"
    cidr_blocks = ["
    prefix_list_ids
  }

  tags = {
    Name: "${var.env
  }
```

Modules

Module for EC2 Instance

"webserver" module

And also a security group, because we're assigning the security group to easy to instant's, so this



```
    }

    ingress {
        from_port = 8080
        to_port = 8080
        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"]
        prefix_list_ids = []
    }

    tags = {
        Name: "${var.env_prefix}-default-sg"
    }
}

data "aws_ami" "latest-amazon-linux-image" {
    most_recent = true
    owners = ["amazon"]
    filter {
        name = "name"
```

Modules

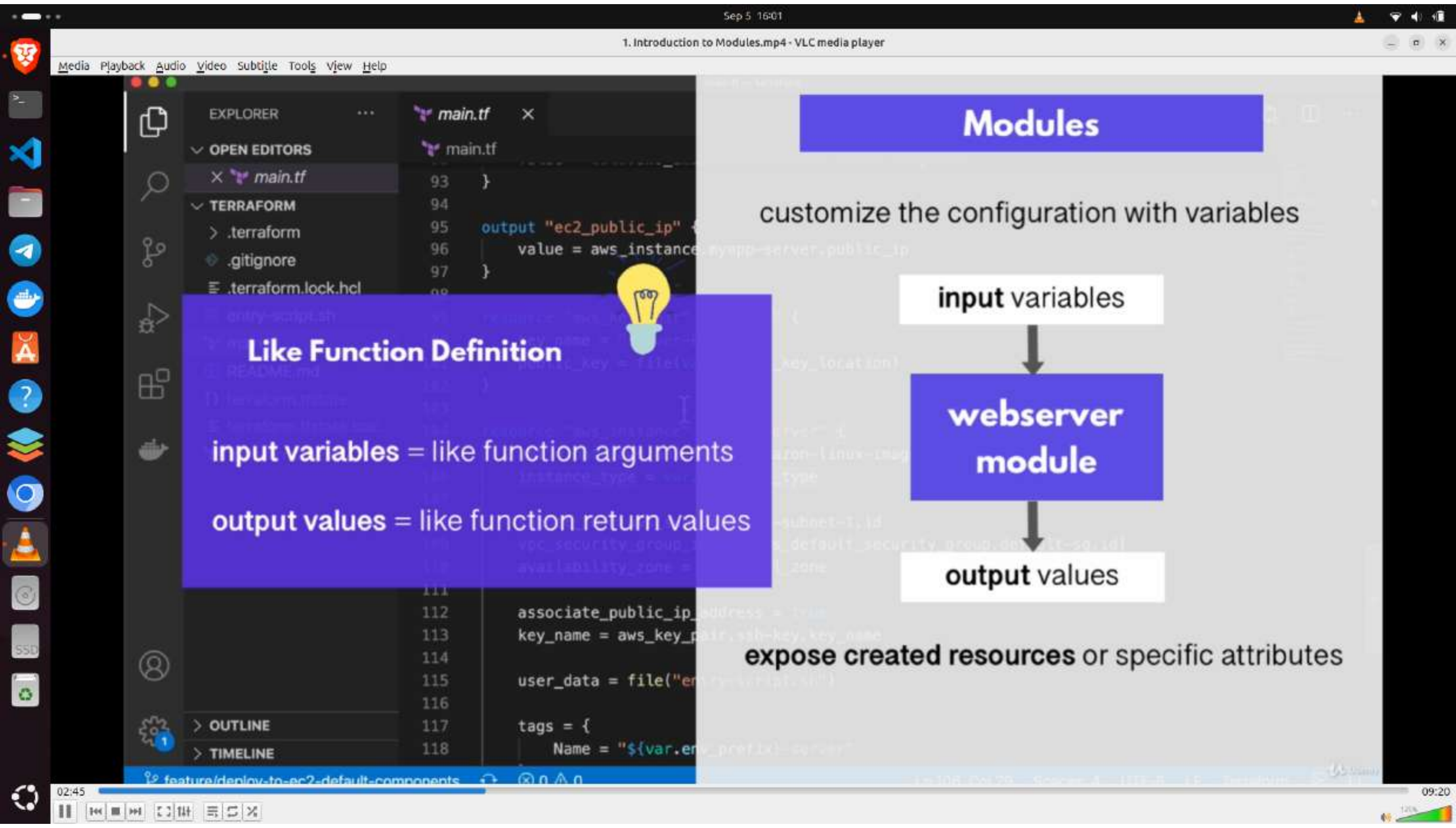
webserver module

re-use same configuration

region A

region B

region C


Sep 5 16:01
1. Introduction to Modules.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
EXPLORER
OPEN EDITORS
main.tf
TERRAFORM
.terraform
.gitignore
.terraform.lock.hcl
OUTLINE
TIMELINE
feature/deploy-to-ec2-default-components
02:45
09:20

```
}

output "ec2_public_ip" {
    value = aws_instance.myapp-server.public_ip
}

    associate_public_ip_address = true
    key_name = aws_key_pair.ssh-key.key_name

    user_data = file("entry-script.sh")

    tags = {
        Name = "${var.env_prefix}-server"
```

Modules

customize the configuration with variables

input variables

webserver module

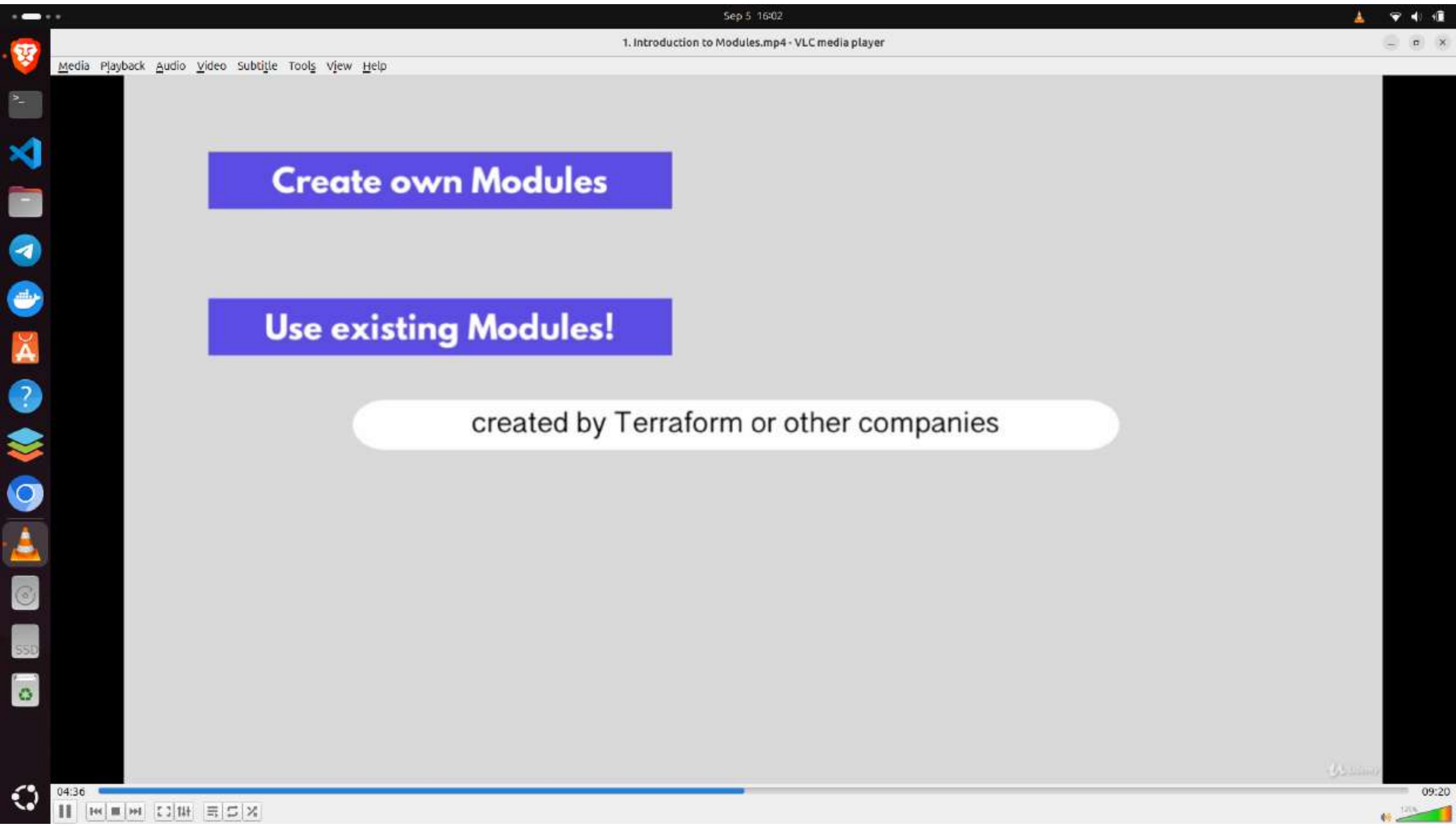
output values

expose created resources or specific attributes

Like Function Definition

input variables = like function arguments

output values = like function return values

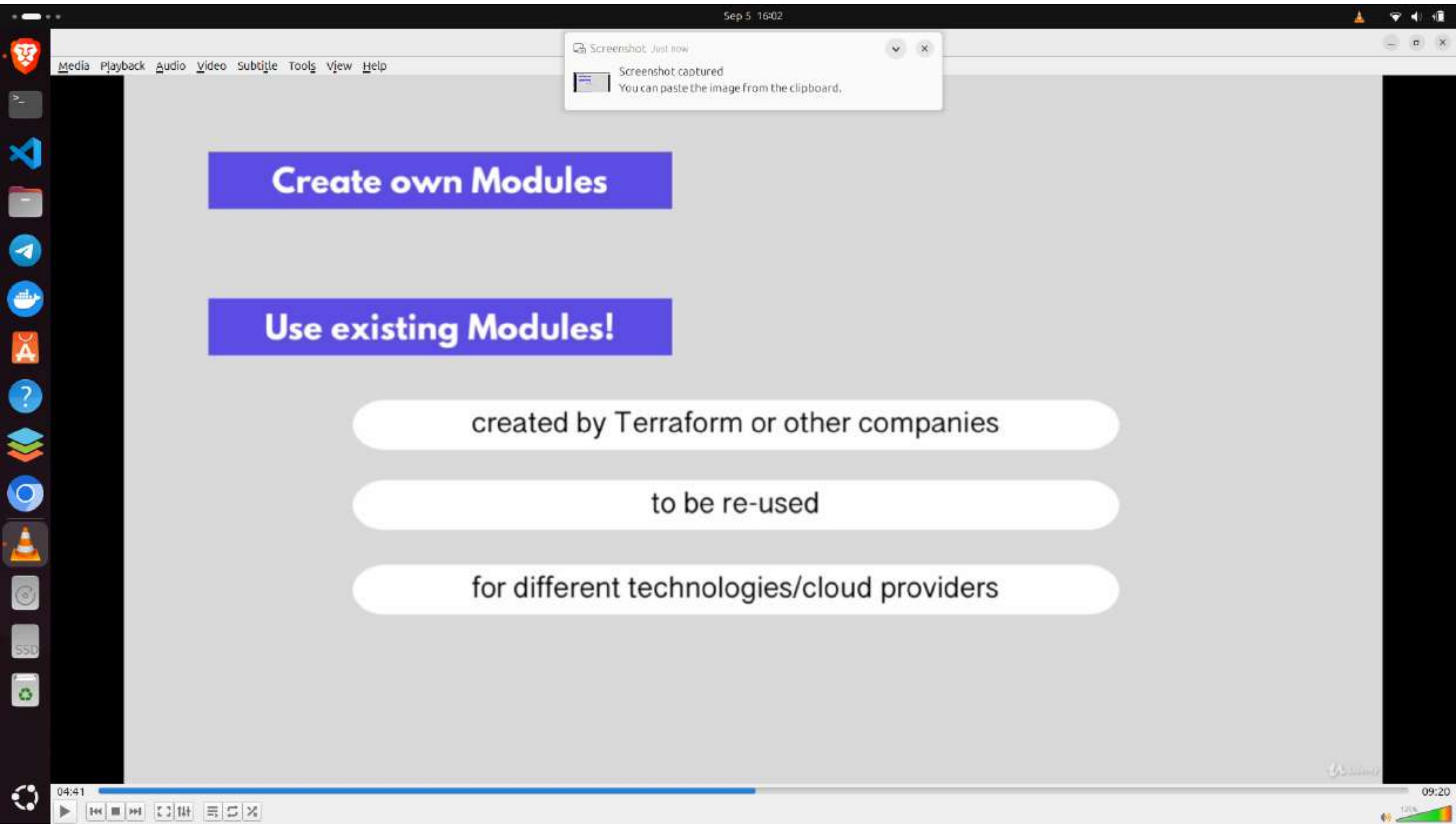


Sep 5 16:02
1. Introduction to Modules.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
04:36
09:20

Create own Modules

Use existing Modules!

created by Terraform or other companies



Screenshot Just now

Screenshot captured

You can paste the image from the clipboard.

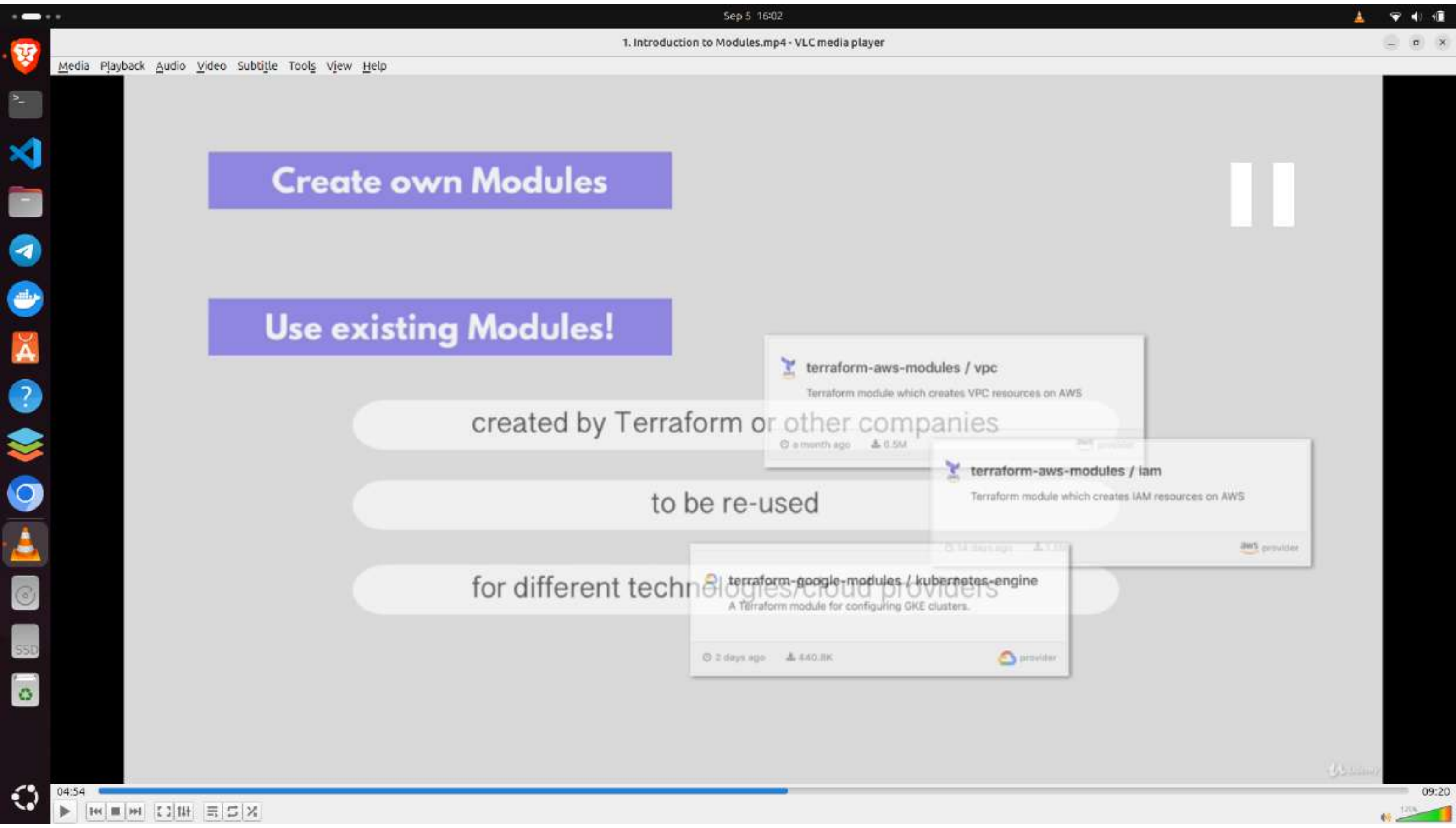
Create own Modules

Use existing Modules!

created by Terraform or other companies

to be re-used

for different technologies/cloud providers



Create own Modules

Use existing Modules!

created by Terraform or other companies

to be re-used

for different technologies/cloud providers

terraform-aws-modules / vpc

Terraform module which creates VPC resources on AWS

a month ago 0.5M

terraform-aws-modules / iam

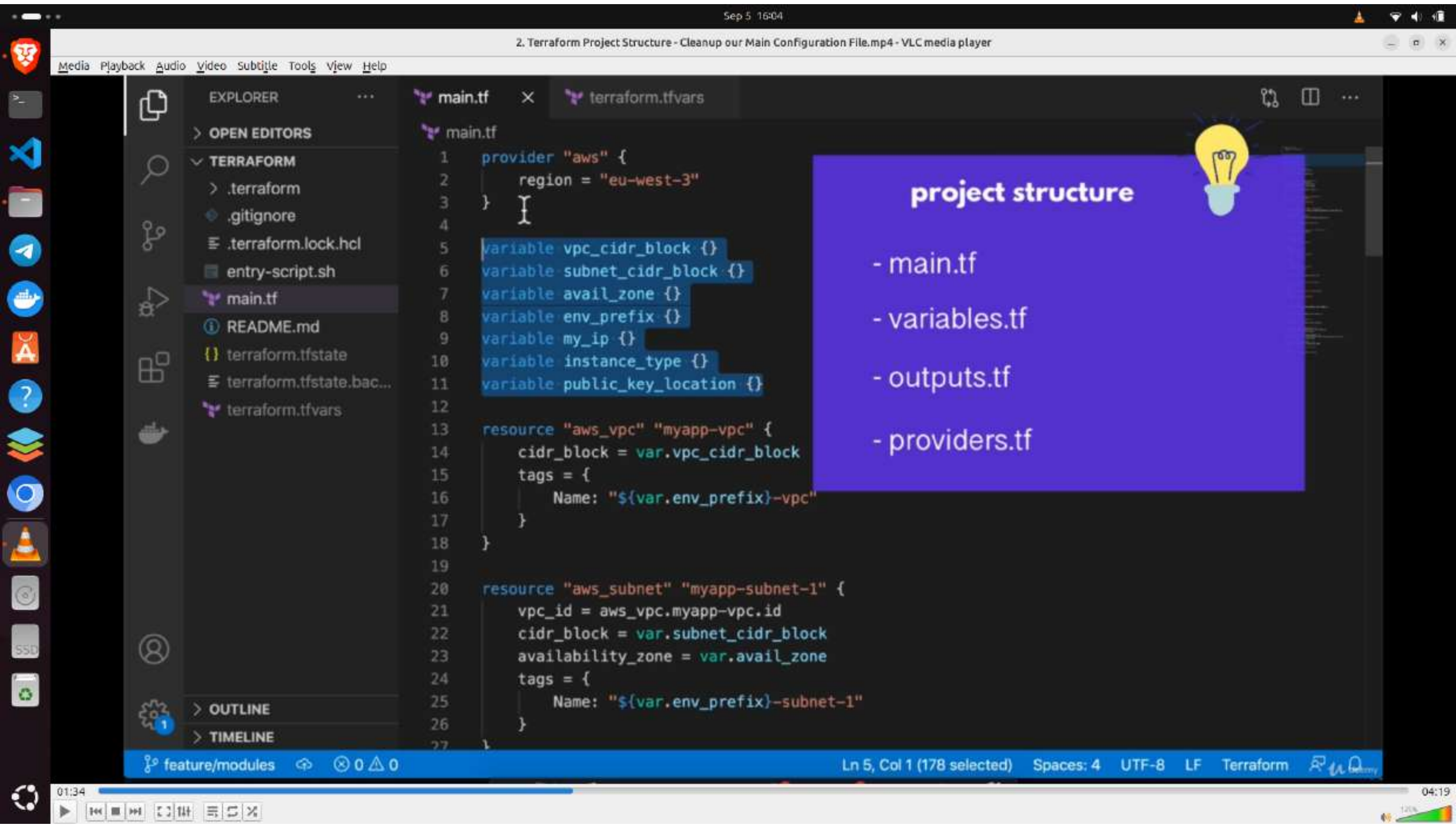
Terraform module which creates IAM resources on AWS

aws provider

terraform-google-modules / kubernetes-engine

A Terraform module for configuring GKE clusters.

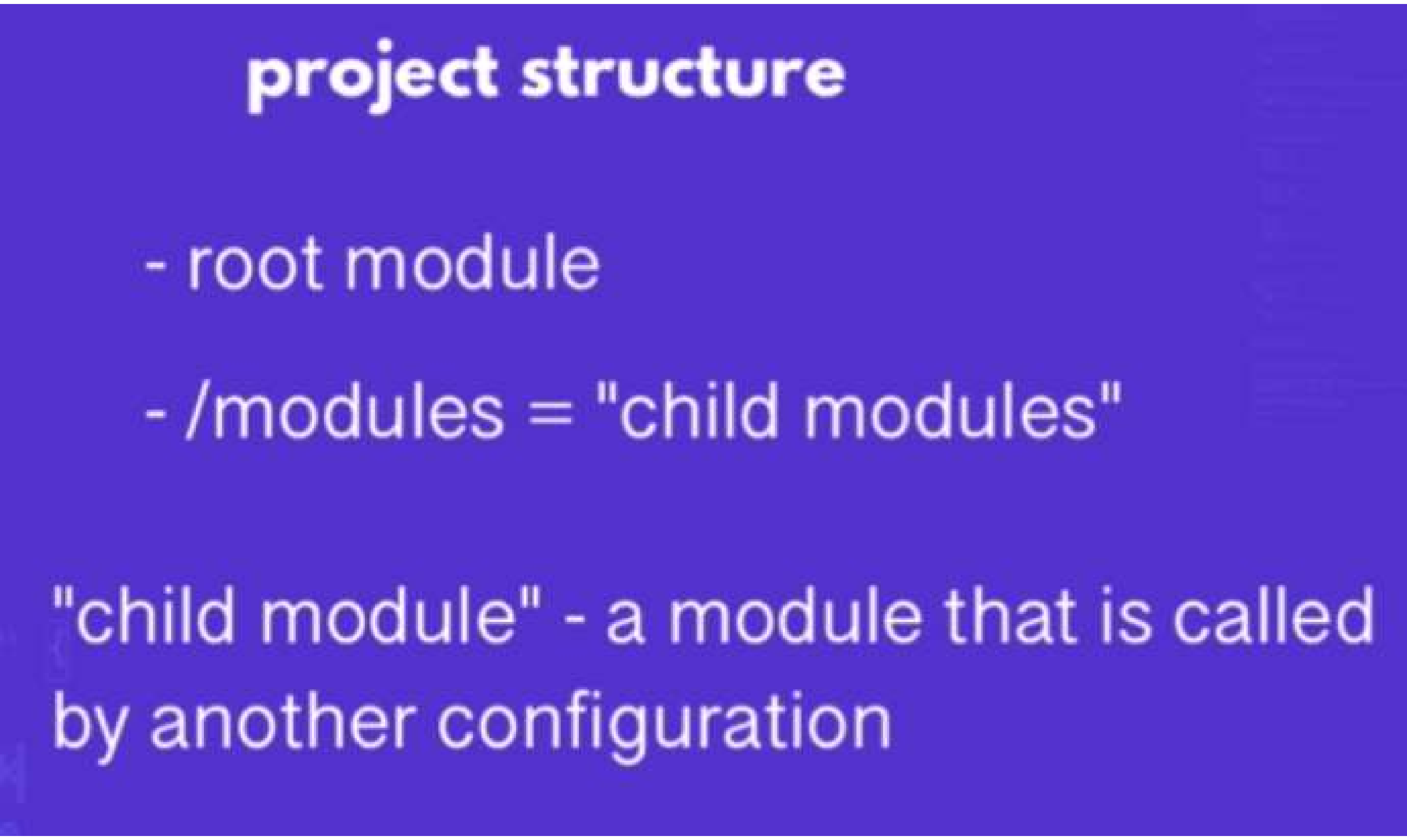
2 days ago 440.8K provider



```
provider "aws" {
    region = "eu-west-3"
}

variable vpc_cidr_block {}
variable subnet_cidr_block {}
variable avail_zone {}
variable env_prefix {}
variable my_ip {}
variable instance_type {}
variable public_key_location {}

resource "aws_vpc" "myapp-vpc" {
    cidr_block = var.vpc_cidr_block
    tags = {
        Name: "${var.env_prefix}-vpc"
    }
}

resource "aws_subnet" "myapp-subnet-1" {
    vpc_id = aws_vpc.myapp-vpc.id
    cidr_block = var.subnet_cidr_block
    availability_zone = var.avail_zone
    tags = {
        Name: "${var.env_prefix}-subnet-1"
    }
```

project structure

- main.tf
- variables.tf
- outputs.tf
- providers.tf



project structure

- root module
- /modules = "child modules"

"child module" - a module that is called by another configuration

```
me: "${var.env_prefix}-igw"

aws_d
t_route_table_id = aws_vpc.myapp-vpc.default_route_table_id
```

group multiple resources into a **logical** unit