VARIABLES IN TERRAFORM

To keep the things dynamic in terraform configuration and not hardcoding the values, we can make use of variables.
We can define variables in a separate file and refer them to the code.

Approaches of variable Assignment

- Variables can be defined and mentioned in a number of ways.
- Default file is variable.tf (To define variables and default values)
- To specify explicit values, we have another file called terraform.tfvars
- terraform.tfvars is the default file name, if we have a custom file name, we can mention it with command

Variable data types

We can explicitly define the data type of a variable; this will restrict the variable to accept only that specific type of value.

```
variable "region-name" {
  default = "us-east-1"
  type = string
}
```

Here region-name variable can only accept string data type and if provided with any other data type, it will throw an error.

Variable data types supported by terraform

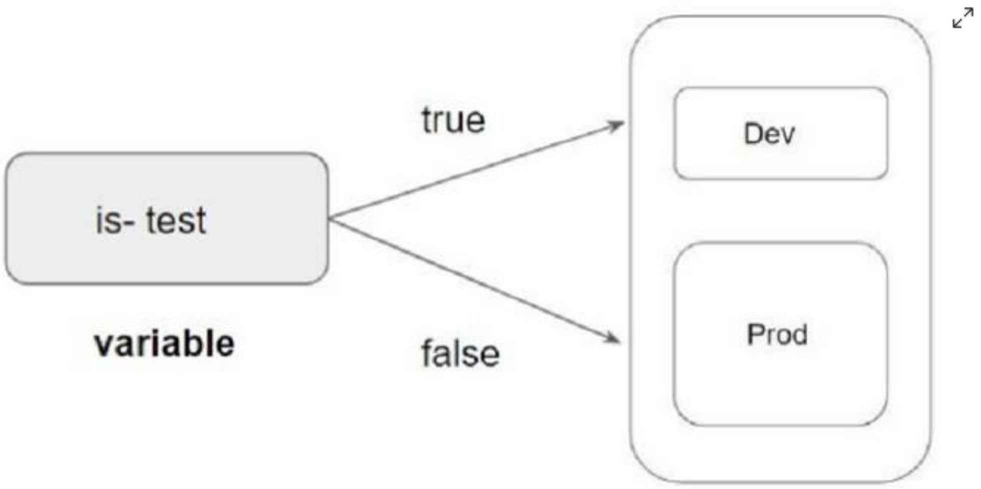
Terraform supports a variety of data types like string, map, number, bool, list tuple.
<u>To read in detail, please follow the below link</u>

1. string
2. map
3. number
4. bool
5. list
6. any
7. tuple

CONDITIONAL EXPRESSIONS

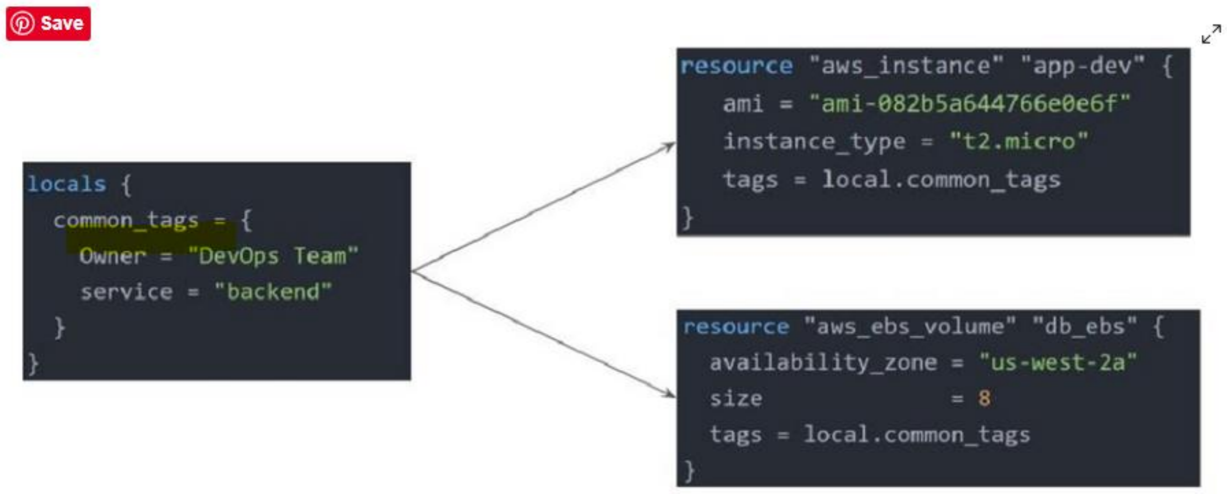
A conditional expression uses the value of a bool expression to select one of two values.
Syntax of Conditional expression:
condition ? true_val : false_val
If the condition is true then the result is true_val. If the condition is false then the result is false_val.
Let’s assume that there are two resource blocks as part of Terraform configuration.

Depending on the variable value, one of the resource blocks will run.



Locals

A local value assigns a name to an expression, allowing it to be used multiple times within a module without repeating it.



```
locals {
  common_tags = {
    Owner = "DevOps Team"
    service = "backend"
  }
}
```

```
resource "aws_instance" "app-dev" {
   ami = "ami-082b5a644766e0e6f"
   instance_type = "t2.micro"
   tags = local.common_tags
}
```

```
resource "aws_ebs_volume" "db_ebs" {
  availability_zone = "us-west-2a"
  size              = 8
  tags = local.common_tags
}
```

- Local values can be helpful to avoid repeating the same values or expressions multiple times in a configuration.

- If overused they can also make a configuration hard to read by future maintainers by hiding the actual values used
- Use local values only in moderation, in situations where a single value or result is used in many places and that value is likely to be changed in the future.

TERRAFORM FUNCTIONS

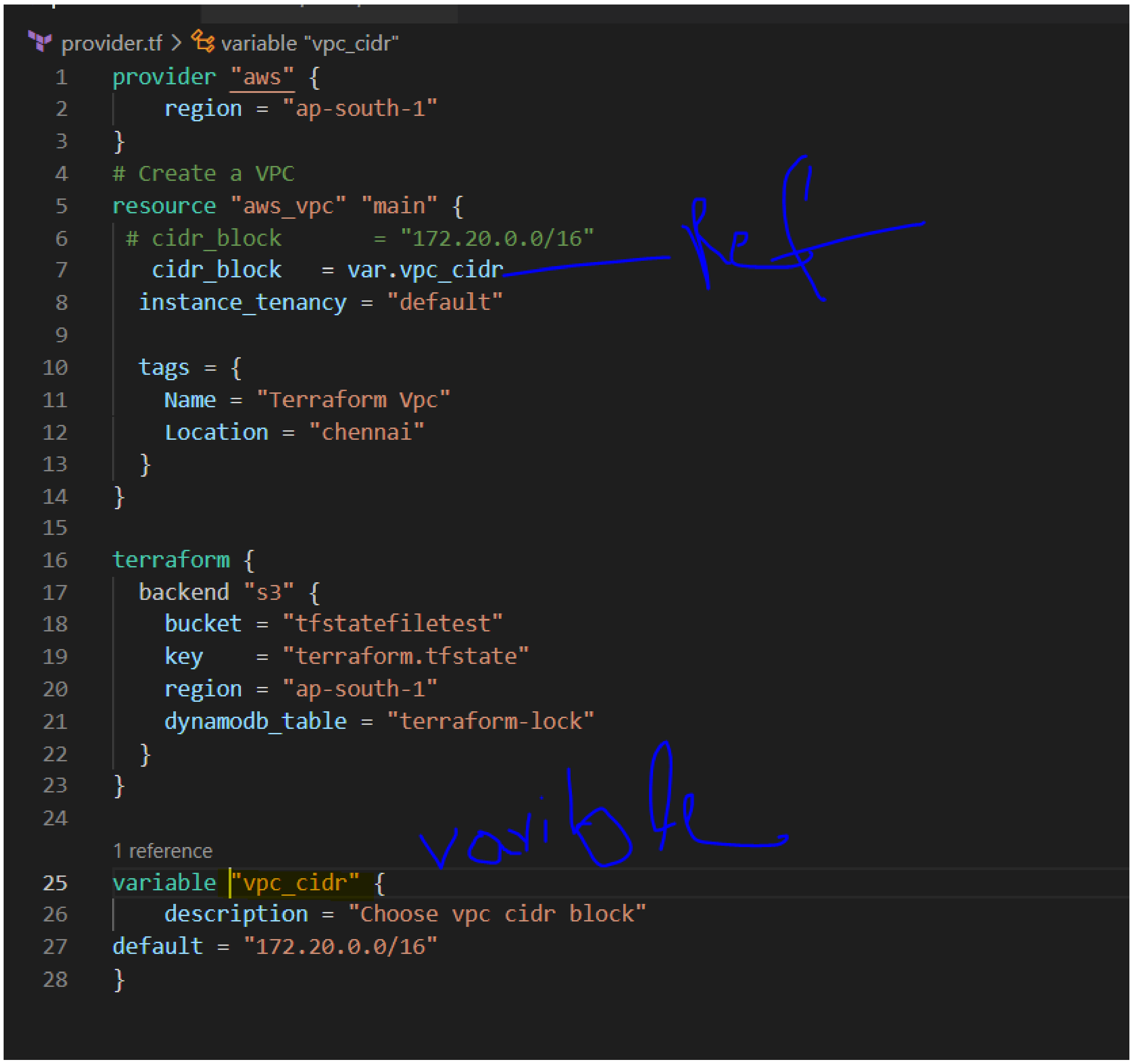
- The Terraform language includes a number of built-in functions that you can use to transform and combine values.
- The general syntax for function calls is a function name followed by comma-separated arguments in parentheses:

function (argument1, argument2)
Example:
```
> max(5, 12, 9)
12
```
The Terraform language does not support user-defined functions, and so only the functions built into the language are available for use.

- Numeric
- String
- Collection
- Encoding
- Filesystem
- Date and Time
- Hash and Crypto
- IP Network
- Type Conversion

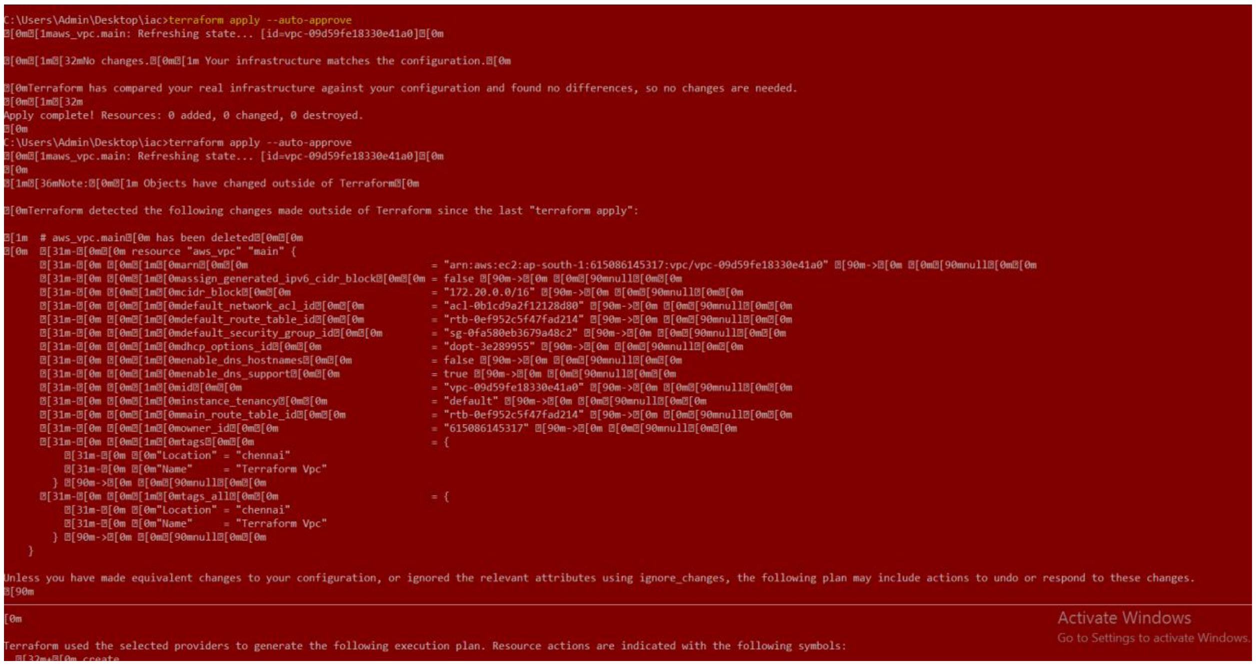
There is practically a large number of functions that terraform supports, to check how these work, we can make use of terraform console command and test them out.

Ex

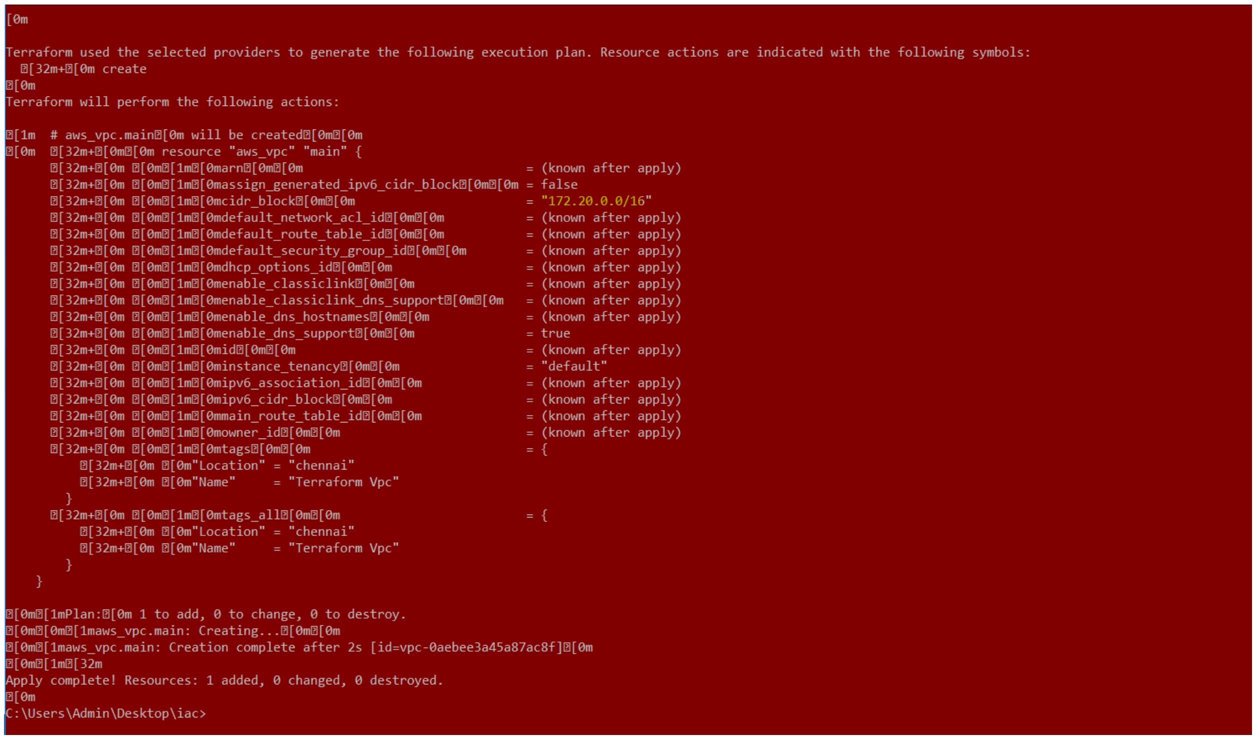


provider.tf > variable "vpc_cidr"

```
provider "aws" {
    region = "ap-south-1"
}
# Create a VPC
resource "aws_vpc" "main" {
 # cidr_block        = "172.20.0.0/16"
   cidr_block    = var.vpc_cidr
  instance_tenancy = "default"

  tags = {
    Name = "Terraform Vpc"
    Location = "chennai"
  }
}

terraform {
  backend "s3" {
    bucket = "tfstatefiletest"
    key    = "terraform.tfstate"
    region = "ap-south-1"
    dynamodb_table = "terraform-lock"
  }
}

1 reference
variable "vpc_cidr" {
    description = "Choose vpc cidr block"
default = "172.20.0.0/16"
}
```



```
C:\Users\Admin\Desktop\iac>terraform apply --auto-approve
[0m[1maws_vpc.main: Refreshing state... [id=vpc-09d59fe18330e41a0][0m

[0m[1m[32mNo changes.[0m[1m Your infrastructure matches the configuration.[0m

[0mTerraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
[0m[1m[32m
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
[0m
C:\Users\Admin\Desktop\iac>terraform apply --auto-approve
[0m[1maws_vpc.main: Refreshing state... [id=vpc-09d59fe18330e41a0][0m
[0m
[1m[36mNote:[0m[1m Objects have changed outside of Terraform[0m

[0mTerraform detected the following changes made outside of Terraform since the last "terraform apply":

[1m  # aws_vpc.main[0m has been deleted[0m[0m
[0m  [31m-[0m[0m resource "aws_vpc" "main" {
      [31m-[0m [0m[1m[0marn[0m[0m                              = "arn:aws:ec2:ap-south-1:615086145317:vpc/vpc-09d59fe18330e41a0" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0massign_generated_ipv6_cidr_block[0m[0m = false [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mcidr_block[0m[0m                       = "172.20.0.0/16" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mdefault_network_acl_id[0m[0m           = "acl-0b1cd9a2f12128d80" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mdefault_route_table_id[0m[0m           = "rtb-0ef952c5f47fad214" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mdefault_security_group_id[0m[0m        = "sg-0fa580eb3679a48c2" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mdhcp_options_id[0m[0m                  = "dopt-3e289955" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0menable_dns_hostnames[0m[0m             = false [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0menable_dns_support[0m[0m               = true [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mid[0m[0m                               = "vpc-09d59fe18330e41a0" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0minstance_tenancy[0m[0m                 = "default" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mmain_route_table_id[0m[0m              = "rtb-0ef952c5f47fad214" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mowner_id[0m[0m                         = "615086145317" [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mtags[0m[0m                             = {
          [31m-[0m [0m"Location" = "chennai"
          [31m-[0m [0m"Name"     = "Terraform Vpc"
        } [90m->[0m [0m[90mnull[0m[0m
      [31m-[0m [0m[1m[0mtags_all[0m[0m                         = {
          [31m-[0m [0m"Location" = "chennai"
          [31m-[0m [0m"Name"     = "Terraform Vpc"
        } [90m->[0m [0m[90mnull[0m[0m
    }

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to undo or respond to these changes.
[90m

[0m

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
```



```
[0m

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  [32m+[0m create
[0m
Terraform will perform the following actions:

[1m  # aws_vpc.main[0m will be created[0m[0m
[0m  [32m+[0m[0m resource "aws_vpc" "main" {
      [32m+[0m [0m[1m[0marn[0m[0m                              = (known after apply)
      [32m+[0m [0m[1m[0massign_generated_ipv6_cidr_block[0m[0m = false
      [32m+[0m [0m[1m[0mcidr_block[0m[0m                       = "172.20.0.0/16"
      [32m+[0m [0m[1m[0mdefault_network_acl_id[0m[0m           = (known after apply)
      [32m+[0m [0m[1m[0mdefault_route_table_id[0m[0m           = (known after apply)
      [32m+[0m [0m[1m[0mdefault_security_group_id[0m[0m        = (known after apply)
      [32m+[0m [0m[1m[0mdhcp_options_id[0m[0m                  = (known after apply)
      [32m+[0m [0m[1m[0menable_classiclink[0m[0m               = (known after apply)
      [32m+[0m [0m[1m[0menable_classiclink_dns_support[0m[0m    = (known after apply)
      [32m+[0m [0m[1m[0menable_dns_hostnames[0m[0m             = (known after apply)
      [32m+[0m [0m[1m[0menable_dns_support[0m[0m               = true
      [32m+[0m [0m[1m[0mid[0m[0m                               = (known after apply)
      [32m+[0m [0m[1m[0minstance_tenancy[0m[0m                 = "default"
      [32m+[0m [0m[1m[0mipv6_association_id[0m[0m              = (known after apply)
      [32m+[0m [0m[1m[0mipv6_cidr_block[0m[0m                  = (known after apply)
      [32m+[0m [0m[1m[0mmain_route_table_id[0m[0m              = (known after apply)
      [32m+[0m [0m[1m[0mowner_id[0m[0m                         = (known after apply)
      [32m+[0m [0m[1m[0mtags[0m[0m                             = {
          [32m+[0m [0m"Location" = "chennai"
          [32m+[0m [0m"Name"     = "Terraform Vpc"
        }
      [32m+[0m [0m[1m[0mtags_all[0m[0m                         = {
          [32m+[0m [0m"Location" = "chennai"
          [32m+[0m [0m"Name"     = "Terraform Vpc"
        }
    }

[0m[1mPlan:[0m 1 to add, 0 to change, 0 to destroy.
[0m[0m[1maws_vpc.main: Creating...[0m[0m
[0m[1maws_vpc.main: Creation complete after 2s [id=vpc-0aebee3a45a87ac8f][0m
[0m[1m[32m
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[0m
C:\Users\Admin\Desktop\iac>
```

Using loops in terraform

Declaring count variables in a resource create multiple resources in a loop.

variable "subnet_cidrs" {

type = list(string)

default = ["172.21.0.0/24", "172.21.1.0/24", "172.21.2.0/24"]

}

resource "aws_subnet" "main" {

count = "3"

vpc_id = aws_vpc.main.id

cidr_block = var.subnet_cidrs[count.index]

tags = {

Name = "Subnet-${count.index + 1}"

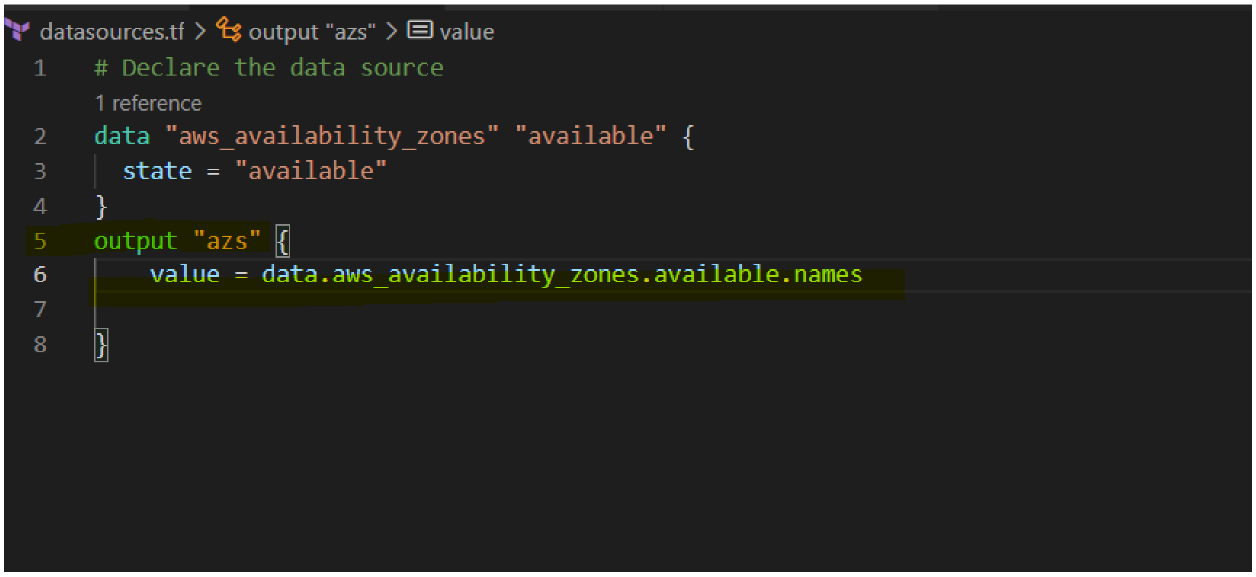
}

}

Terraform Datasource

Using data source we can fetch certain information from aws account dynamically

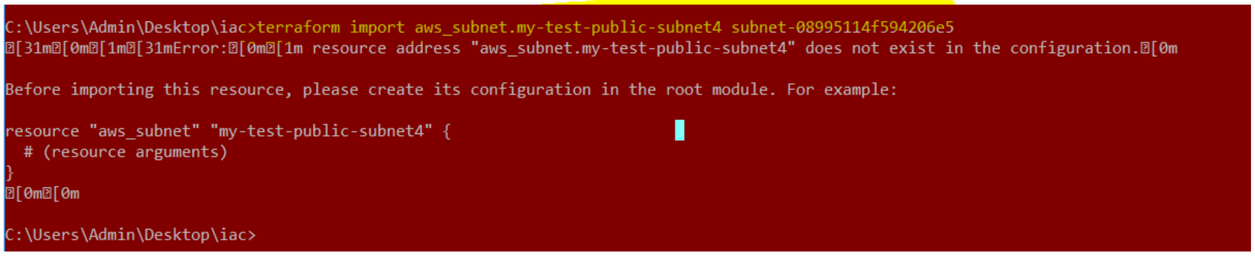
For example i want of fetch list of azs in current region



```
# Declare the data source
data "aws_availability_zones" "available" {
  state = "available"
}
output "azs" {
    value = data.aws_availability_zones.available.names

}
```

Terraform Import

Let's say there is a resource manually created and we want that resource to be managed by terraform then use “terraform import”

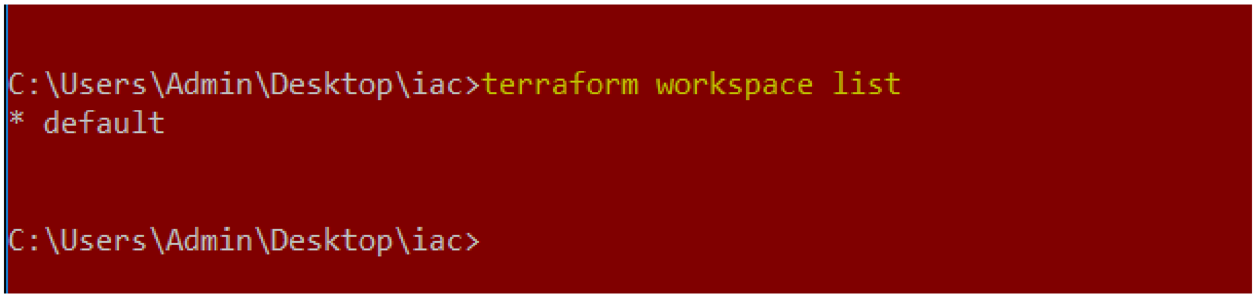


```
C:\Users\Admin\Desktop\iac>terraform import aws_subnet.my-test-public-subnet4 subnet-08995114f594206e5
[31m[0m[1m[31mError:[0m[1m resource address "aws_subnet.my-test-public-subnet4" does not exist in the configuration.[0m

Before importing this resource, please create its configuration in the root module. For example:

resource "aws_subnet" "my-test-public-subnet4" {
  # (resource arguments)
}
[0m[0m

C:\Users\Admin\Desktop\iac>
```

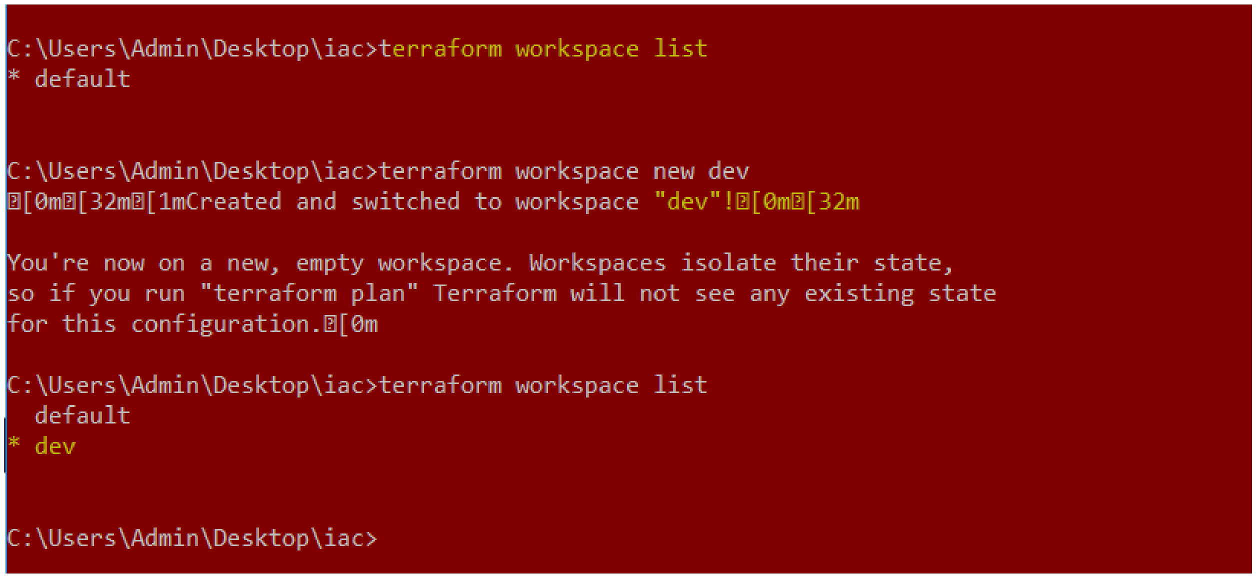
Terraform Workspace

Using terraform workspace we can manage multiple environments with the same configuration files and using different state files.

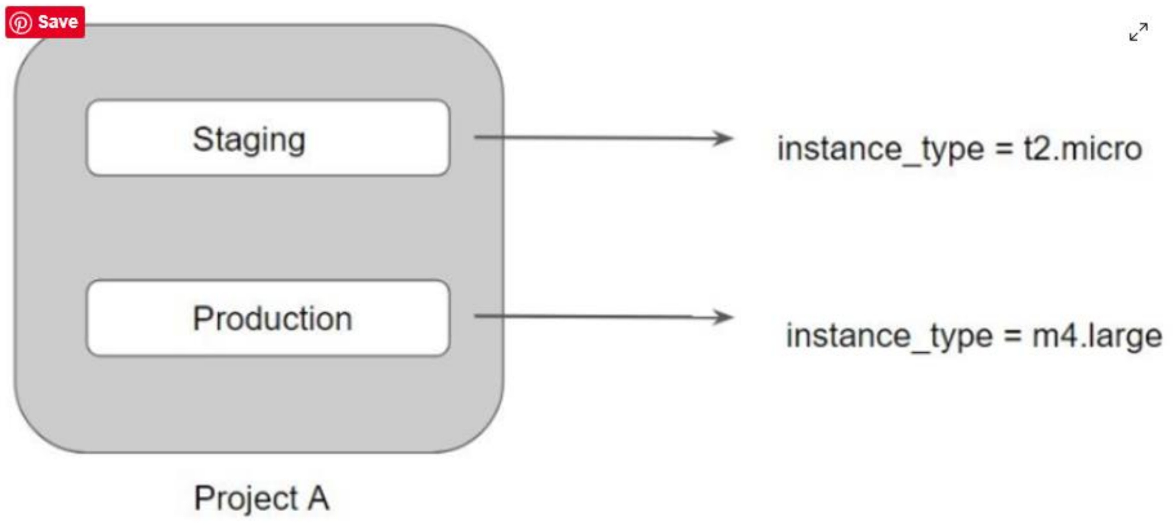
Terraform maintains “default” workspace by default.

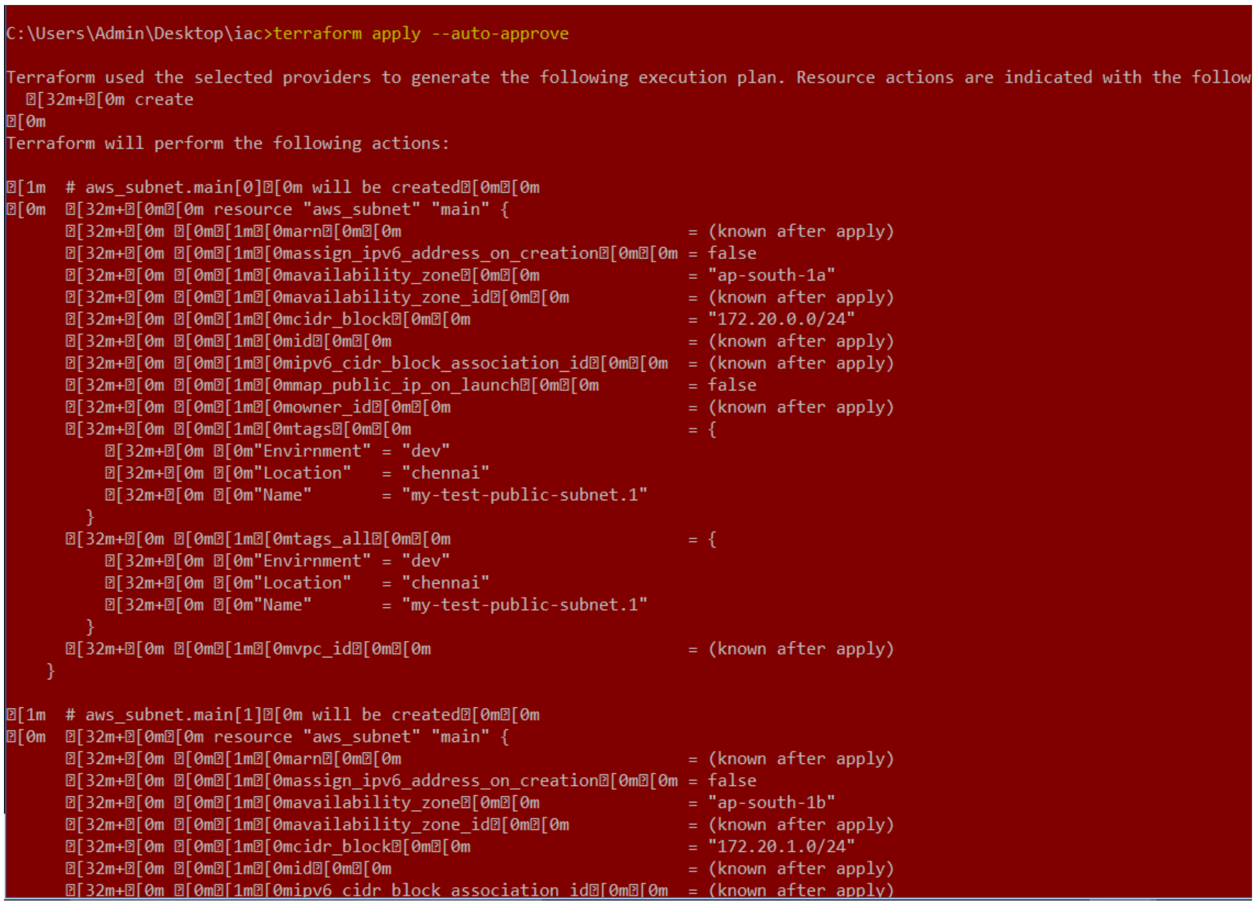


```
C:\Users\Admin\Desktop\iac>terraform workspace list
* default


C:\Users\Admin\Desktop\iac>
```

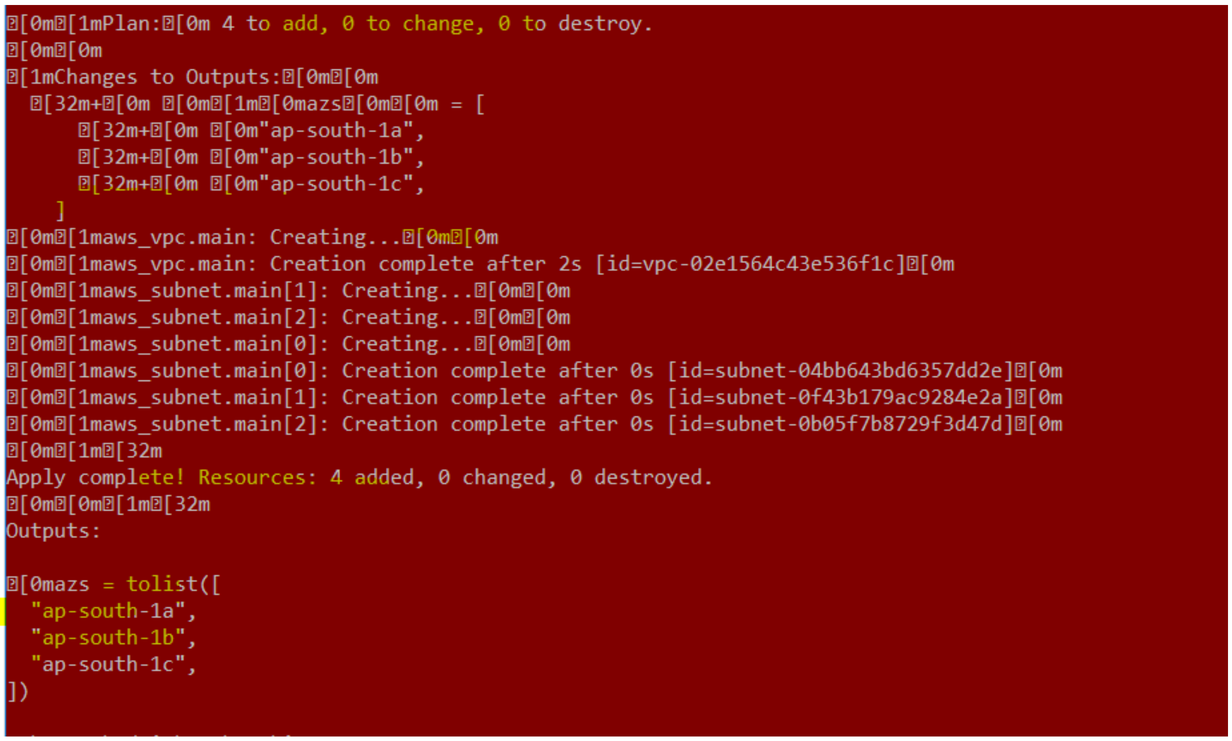


```
C:\Users\Admin\Desktop\iac>terraform workspace list
* default


C:\Users\Admin\Desktop\iac>terraform workspace new dev
[0m[32m[1mCreated and switched to workspace "dev"![0m[32m

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.[0m

C:\Users\Admin\Desktop\iac>terraform workspace list
  default
* dev


C:\Users\Admin\Desktop\iac>
```



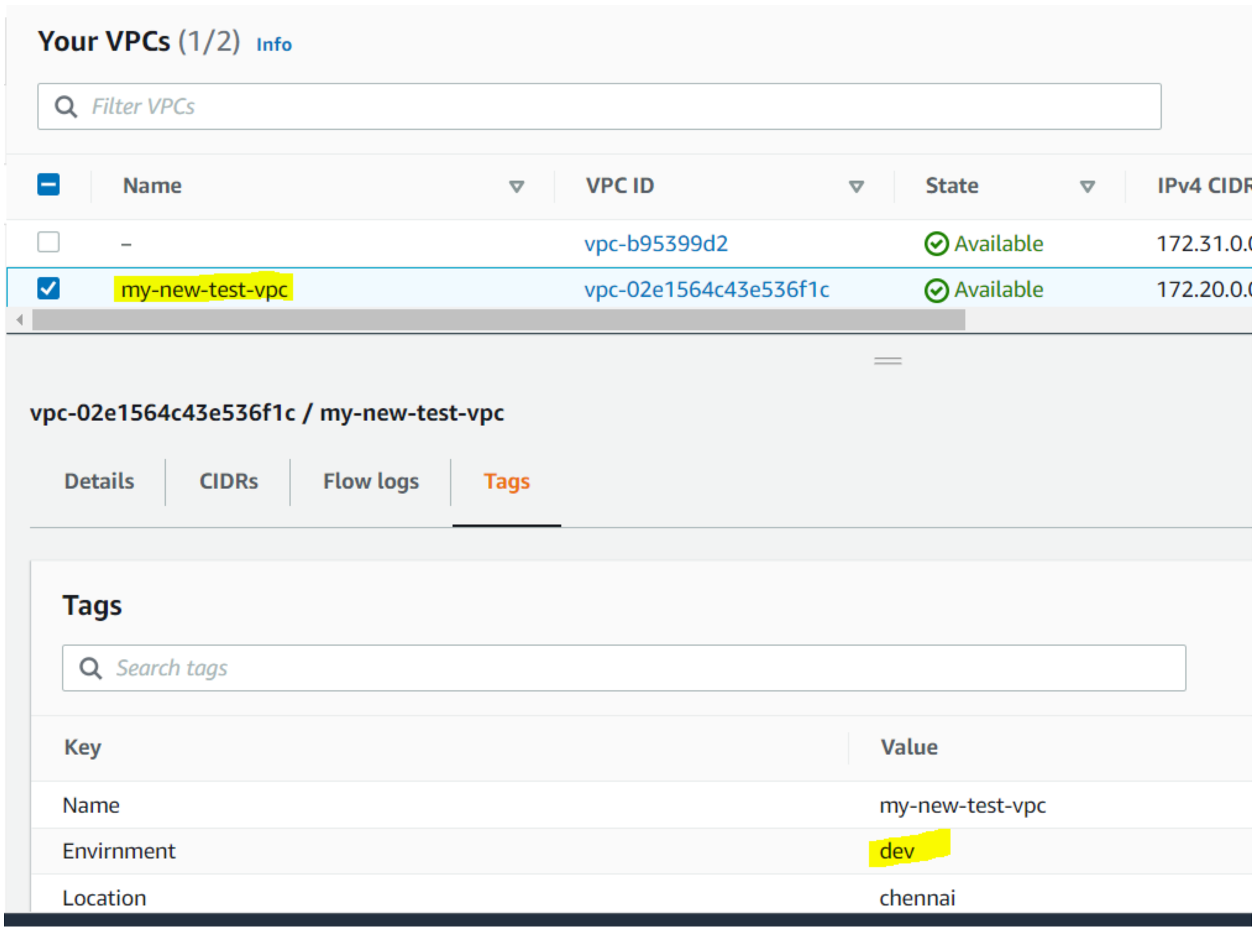


```
C:\Users\Admin\Desktop\iac>terraform apply --auto-approve

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the follow
  [32m+[0m create
[0m
Terraform will perform the following actions:

[1m  # aws_subnet.main[0][0m will be created[0m[0m
[0m  [32m+[0m[0m resource "aws_subnet" "main" {
      [32m+[0m [0m[1m[0marn[0m[0m                                    = (known after apply)
      [32m+[0m [0m[1m[0massign_ipv6_address_on_creation[0m[0m = false
      [32m+[0m [0m[1m[0mavailability_zone[0m[0m                     = "ap-south-1a"
      [32m+[0m [0m[1m[0mavailability_zone_id[0m[0m                  = (known after apply)
      [32m+[0m [0m[1m[0mcidr_block[0m[0m                            = "172.20.0.0/24"
      [32m+[0m [0m[1m[0mid[0m[0m                                    = (known after apply)
      [32m+[0m [0m[1m[0mipv6_cidr_block_association_id[0m[0m  = (known after apply)
      [32m+[0m [0m[1m[0mmap_public_ip_on_launch[0m[0m               = false
      [32m+[0m [0m[1m[0mowner_id[0m[0m                              = (known after apply)
      [32m+[0m [0m[1m[0mtags[0m[0m                                  = {
          [32m+[0m [0m"Envirnment" = "dev"
          [32m+[0m [0m"Location"   = "chennai"
          [32m+[0m [0m"Name"       = "my-test-public-subnet.1"
        }
      [32m+[0m [0m[1m[0mtags_all[0m[0m                              = {
          [32m+[0m [0m"Envirnment" = "dev"
          [32m+[0m [0m"Location"   = "chennai"
          [32m+[0m [0m"Name"       = "my-test-public-subnet.1"
        }
      [32m+[0m [0m[1m[0mvpc_id[0m[0m                                = (known after apply)
    }

[1m  # aws_subnet.main[1][0m will be created[0m[0m
[0m  [32m+[0m[0m resource "aws_subnet" "main" {
      [32m+[0m [0m[1m[0marn[0m[0m                                    = (known after apply)
      [32m+[0m [0m[1m[0massign_ipv6_address_on_creation[0m[0m = false
      [32m+[0m [0m[1m[0mavailability_zone[0m[0m                     = "ap-south-1b"
      [32m+[0m [0m[1m[0mavailability_zone_id[0m[0m                  = (known after apply)
      [32m+[0m [0m[1m[0mcidr_block[0m[0m                            = "172.20.1.0/24"
      [32m+[0m [0m[1m[0mid[0m[0m                                    = (known after apply)
      [32m+[0m [0m[1m[0mipv6_cidr_block_association_id[0m[0m  = (known after apply)
```



```
[0m[1mPlan:[0m 4 to add, 0 to change, 0 to destroy.
[0m[0m
[1mChanges to Outputs:[0m[0m
  [32m+[0m [0m[1m[0mazs[0m[0m = [
      [32m+[0m [0m"ap-south-1a",
      [32m+[0m [0m"ap-south-1b",
      [32m+[0m [0m"ap-south-1c",
    ]
[0m[1maws_vpc.main: Creating...[0m[0m
[0m[1maws_vpc.main: Creation complete after 2s [id=vpc-02e1564c43e536f1c][0m
[0m[1maws_subnet.main[1]: Creating...[0m[0m
[0m[1maws_subnet.main[2]: Creating...[0m[0m
[0m[1maws_subnet.main[0]: Creating...[0m[0m
[0m[1maws_subnet.main[0]: Creation complete after 0s [id=subnet-04bb643bd6357dd2e][0m
[0m[1maws_subnet.main[1]: Creation complete after 0s [id=subnet-0f43b179ac9284e2a][0m
[0m[1maws_subnet.main[2]: Creation complete after 0s [id=subnet-0b05f7b8729f3d47d][0m
[0m[1m[32m
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
[0m[0m[1m[32m
Outputs:

[0mazs = tolist([
  "ap-south-1a",
  "ap-south-1b",
  "ap-south-1c",
])
```

Vpc Env verification



Your VPCs (1/2) Info

Filter VPCs

Name	VPC ID	State	IPv4 CIDR
–	vpc-b95399d2	Available	172.31.0.(
my-new-test-vpc	vpc-02e1564c43e536f1c	Available	172.20.0.(

vpc-02e1564c43e536f1c / my-new-test-vpc

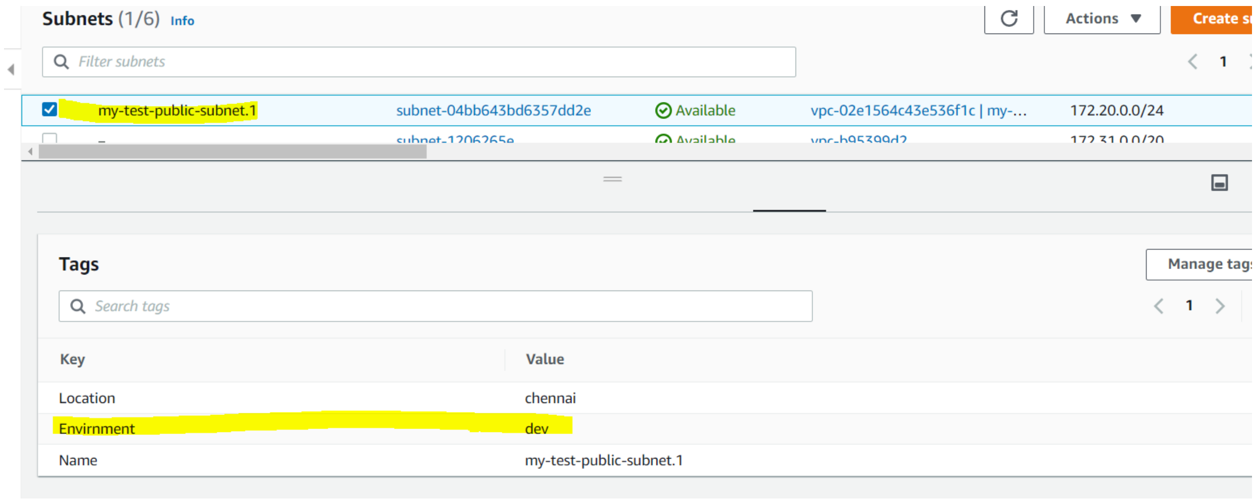
Details | CIDRs | Flow logs | Tags

Tags

Search tags

Key	Value
Name	my-new-test-vpc
Envirnment	dev
Location	chennai

Subnet Env verification



Subnets (1/6) Info

Actions ▼ | Create s

Filter subnets

Name	Subnet ID	State	VPC	IPv4 CIDR
my-test-public-subnet.1	subnet-04bb643bd6357dd2e	Available	vpc-02e1564c43e536f1c | my-…	172.20.0.0/24
–	subnet-1206265e	Available	vpc-b95399d2	172.31.0.0/20

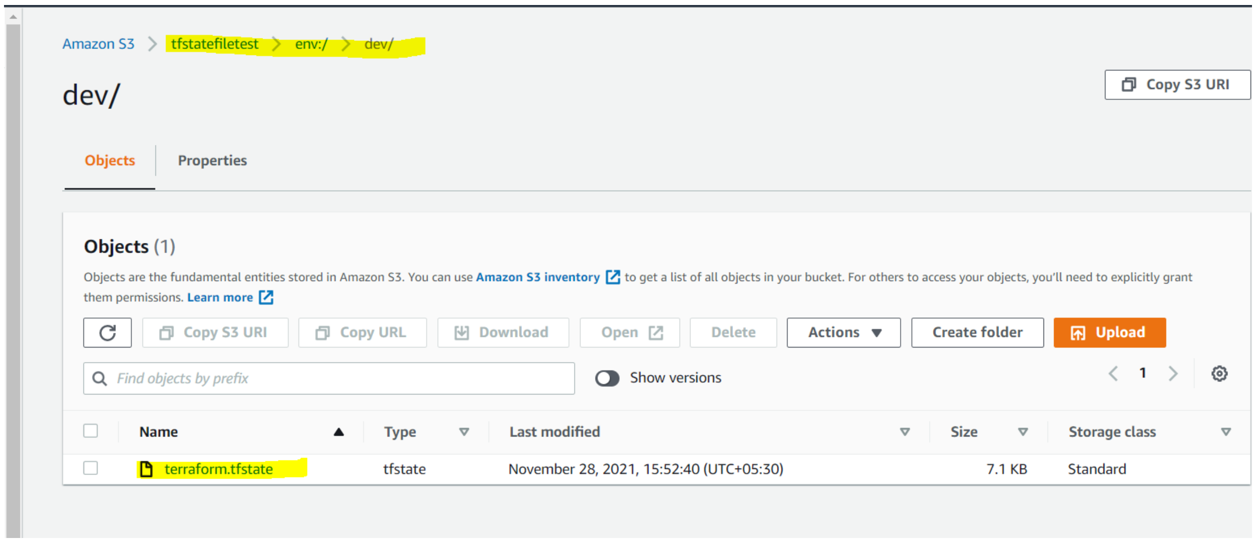
Tags

Manage tag

Search tags

Key	Value
Location	chennai
Envirnment	dev
Name	my-test-public-subnet.1

Env verification on aws s3 bucket:



dev/

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

Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	November 28, 2021, 15:52:40 (UTC+05:30)	7.1 KB	Standard