IaC (Infrastructure as code)

IaC is a code (human readable) that deploys your infrastructure resources onto various platforms instead of managing them manually through a user interface.

Provisioning infrastructure through software to achieve consistent and predictable environment.

Types of IaC tools

Configuration Management

ANSIBLE

puppet

SALTSTACK

- Designed to Install and Manage Software
- Maintains Standard Structure
- Version Control
- Idempotent

Server Templating

docker

HashiCorp Packer

HashiCorp Vagrant

- Pre Installed Software and Dependencies
- Virtual Machine or Docker Images
- Immutable Infrastructure

Provisioning Tools

HashiCorp Terraform

CloudFormation

- Designed to Install and Manage Software
- Maintains Standard Structure
- Version Control
- Idempotent

Why Terraform?

1. No more clicks

2. Enable DevOps

3. Declarative Infrastructure

4. Speed, cost and reduce risk

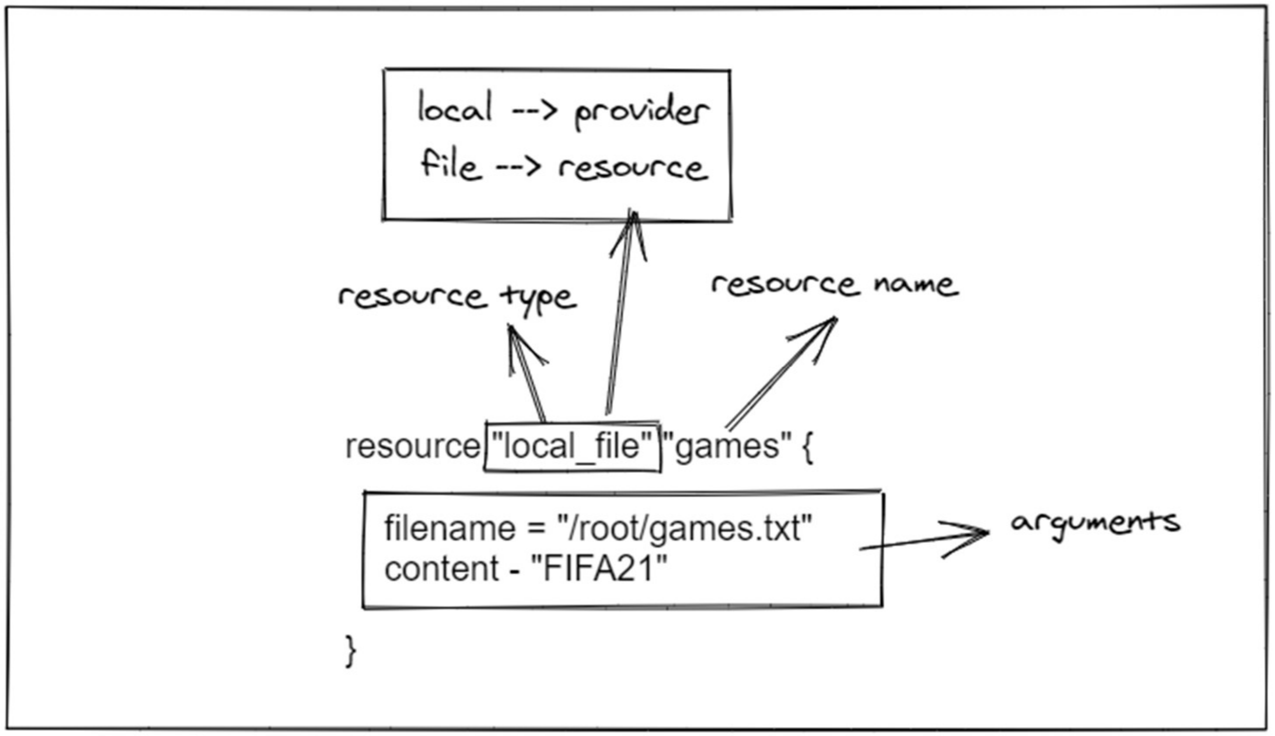
5. Supports various private and public cloud vendors

6. Idempotent (Automatically tracks the state of resources deployed)

7.Consistent Infrastructure

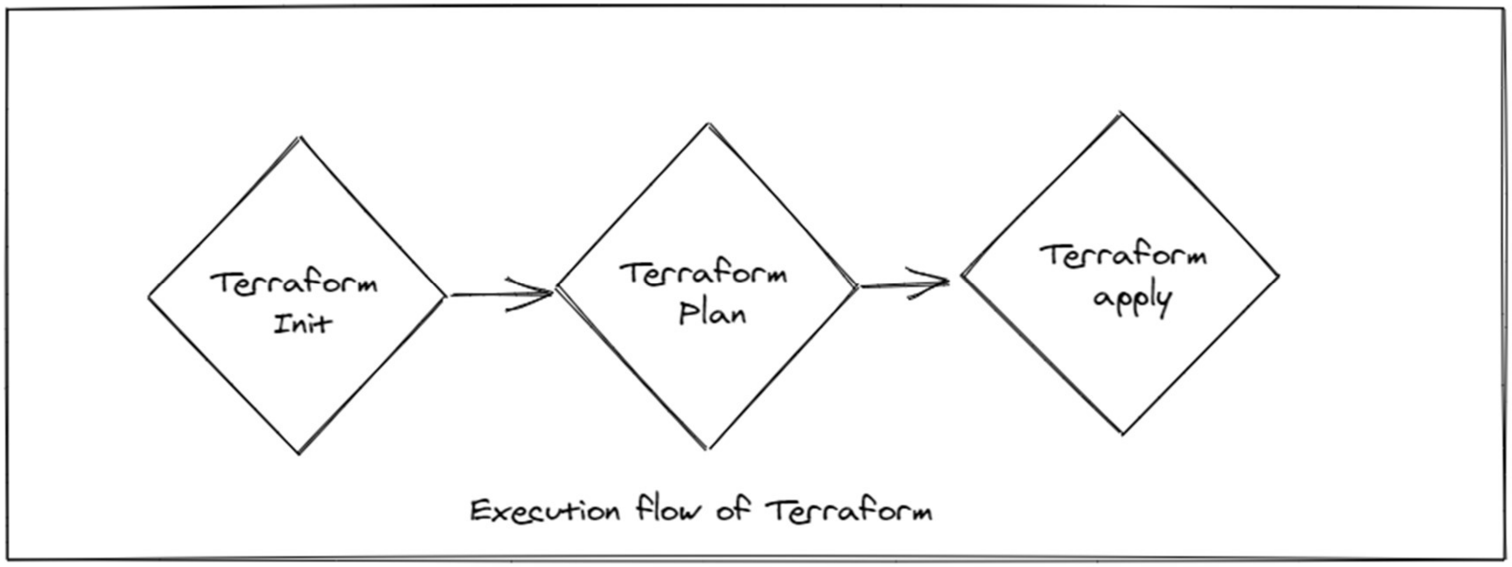
HCL Basics

- HCL is the domain specific language built by Harshicorp (Terraform)
- HCL – Harshicorp configuration language
- “.tf” is the extension of the configuration file



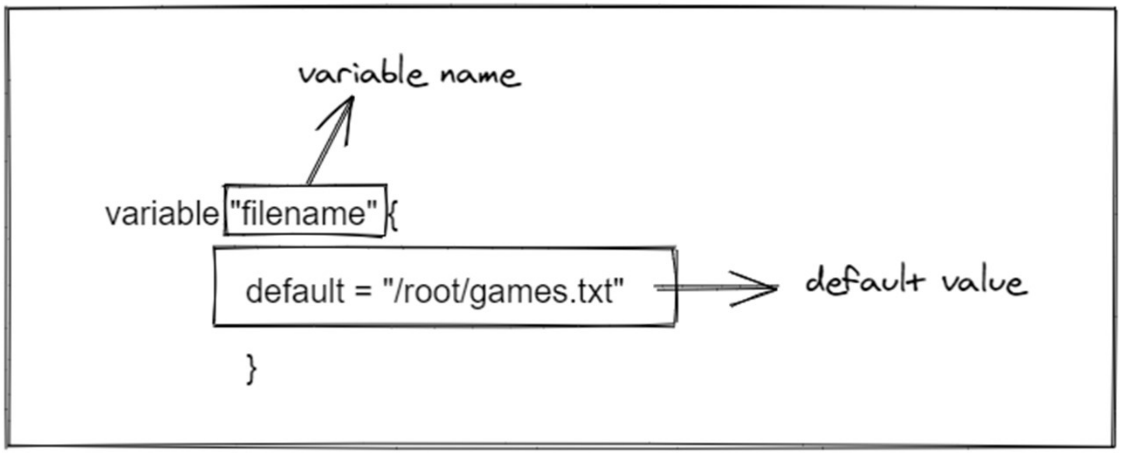
```
resource "local_file" "games" {
 filename = "/root/games.txt"
 content - "FIFA21"
}
```

Terraform Execution Flow

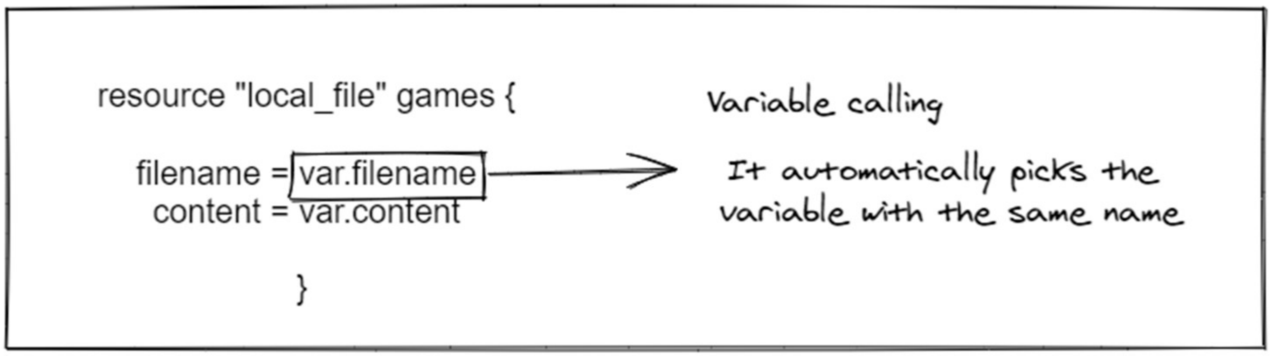


Terraform Variable

1. Declaring a variable



```
variable "filename" {
  default = "/root/games.txt"
}
```



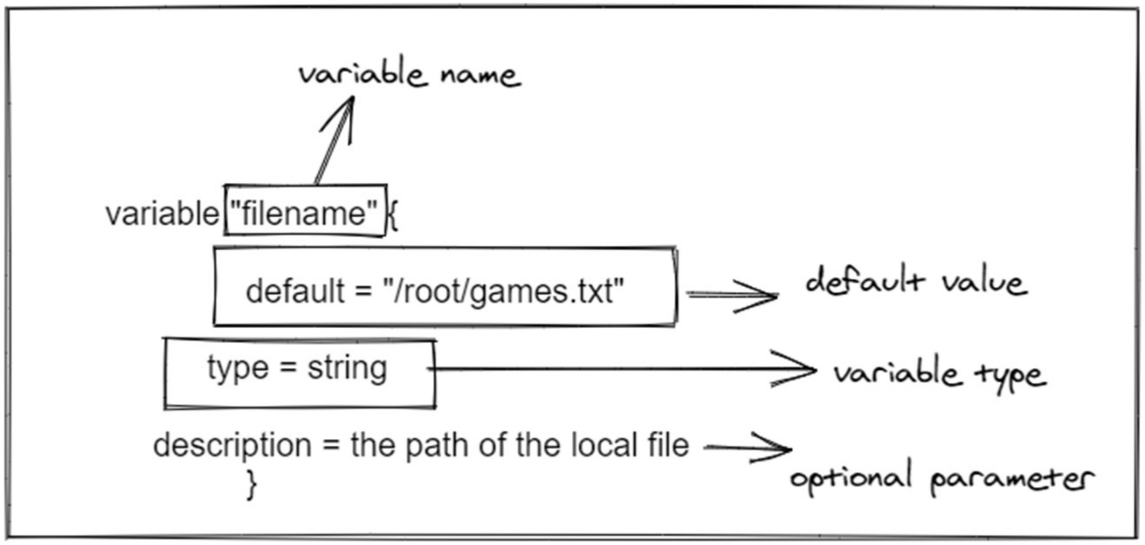
```
resource "local_file" games {
  filename = var.filename
  content = var.content
}
```

Variable calling

It automatically picks the variable with the same name

2. Referencing a variable

Variable Type



```
variable "filename" {
    default = "/root/games.txt"
   type = string
  description = the path of the local file
      }
```

Variable Type

Type	Example
string	“/root/games.txt”
number	1
bool	True/false
any	Default value
list	[“cat”, “Dog”]
map	Pet1 = cat Pet 2 = dog
object	Complex data structure

Passing values to a variable

1) Pass the values via interactive CLI

2) Pass the values via CLI arguments

3) Pass the values as environment variables

4) Define the variables in terraform.tfvars or terraform.tfvars.json file

5) Define the variables in any file and pass the reference of the variables file in terraform apply command

Passing values to a variable

1) Pass the values via interactive CLI

2) Pass the values via CLI arguments

3) Pass the values as environment variables

4) Define the variables in terraform.tfvars or terraform.tfvars.json file

5) Define the variables in any file and pass the reference of the variables file in terraform apply command