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.

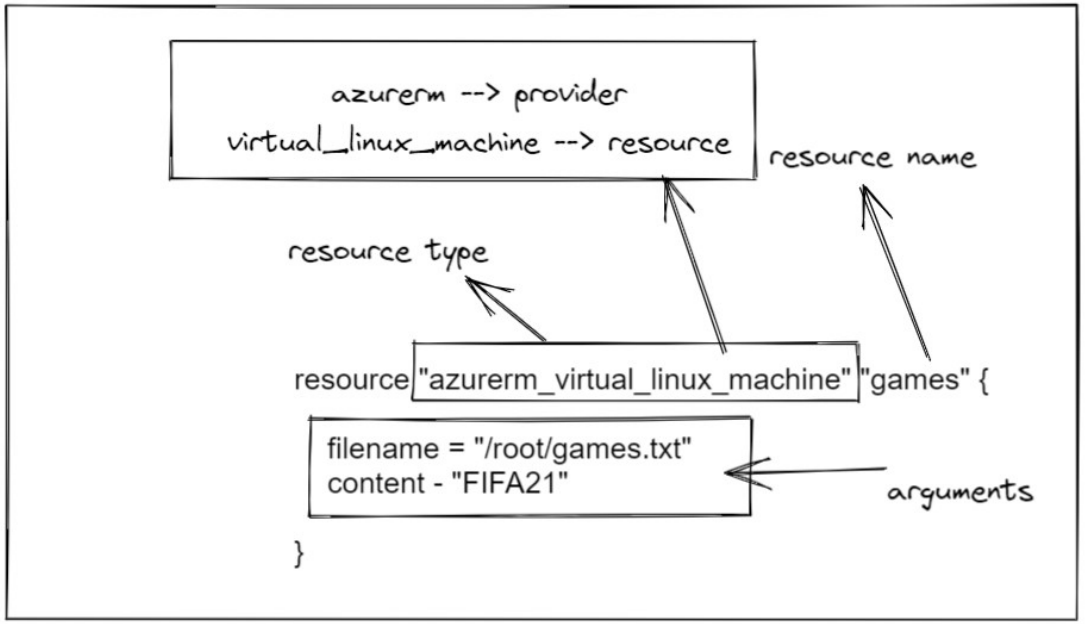
Provisioning infrastructure through software to achieve consistent and predictable environment.

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

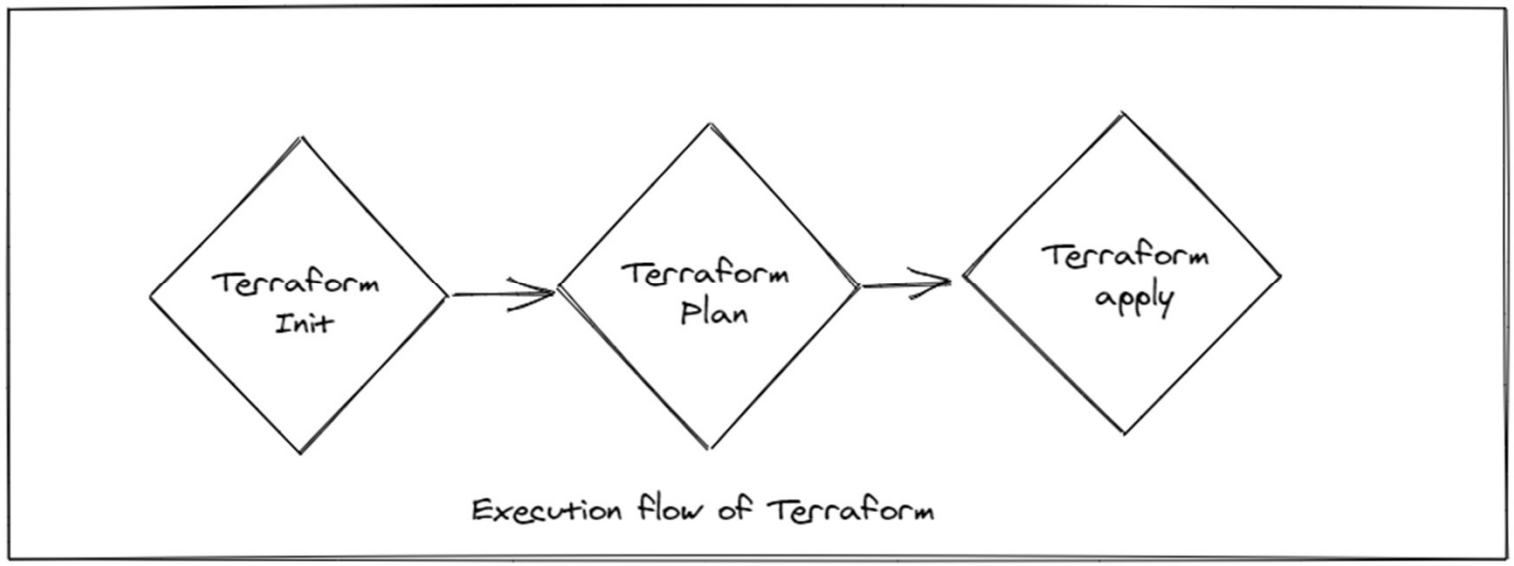
- HashiCorp Configuration Language (HCL) allows for concise descriptions of resources using blocks, arguments, and expressions.
- ".tf" is the extension of the configuration file



How Terraform works?

- Terraform allows infrastructure to be expressed as code in a simple, human readable language called HCL (HashiCorp Configuration Language). It reads configuration files and provides an execution plan of changes, which can be reviewed for safety and then applied and provisioned.
- Extensible providers allow Terraform to manage a broad range of resources, including IaaS, PaaS, SaaS, and hardware services.

Terraform Execution Flow



Azure provider (azurerm)

- The Azure Provider (azurerm) can be used to configure infrastructure in Microsoft Azure using the Azure Resource Manager API's.

```
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "=2.46.0"
    }
  }
}

# Configure the Microsoft Azure Provider
provider "azurerm" {
  features {}
}
```

ARM Template Vs Terraform

• JSON format	HCL format
• Parameters	Variables
• Variables	Local variables
• Resources	Resources
• Function	Functions
• Nested Template	Modules
• Explicit Dependency	Automatic Dependency
• Refer by reference or resource Id	Refer by resource or Data Source

Standalone Virtual Machine