

UNISYS



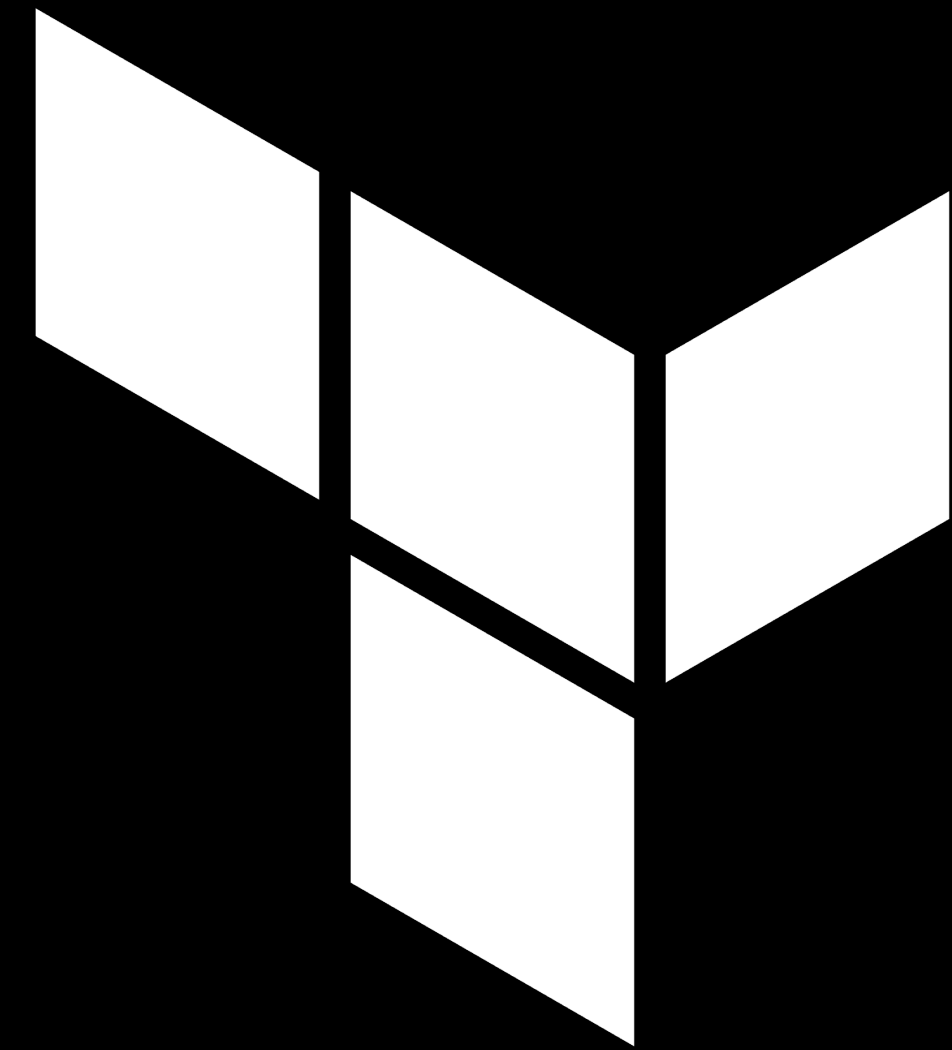
/bin/hatch

Terraform Alapjai

UNISYS képzés - Első előadás

Alapozó

Láng Máté, binhatch.com



HashiCorp

Terraform

Ki vagyok én?

- binhatch.com
- **Ezelőtt:**
 - CTO, Welthee
 - Infra Director @ Connatix
 - CTO, SmartUp
- **Tapasztalat:**
 - Java, Go
 - AWS
 - Kubernetes
 - Terraform



Miért a DevOps?

Avagy a frusztráció ellenszere

- Fizetés feldolgozással foglalkozó kliens
- Gold SLA szolgáltatás
- A “Carlos” box története
 - WebLogic
 - Adatbázis
 - Tanusítványok
- A DevOps a félreértések elkerülésének módszere (nem, nem egy mérnök)

Miért a Terraform?

A gyerekcipőben járó felnőtt

- Első release: 2014 július
- Azóta néhány “keményebb” töréspont
- 2016-ban AWS infrastruktúra létrehozására jó jelöltnek tűnt
 - Ansible
 - CloudFormation
 - Chef / Puppet
- Jelenleg 1.5-ös (1.6 csövön) verzió (aktuális cirkusz - rövid ismertető, OpenTofu)

Képzés áttekintése, célkitűzés

Ott vagyunk már?

- Első alkalom - Terraform alapozó
 - Mi a Terraform és mire használható?
 - Rövid történelmi áttekintés és fejlesztési mérföldkövek
 - Szokványos Terraform munkamenet
 - Terraform leíró nyelve, a HCL
 - Néhány Terraform provider és erőforrás ismertetése

Képzés áttekintése, célkitűzés

Ott vagyunk már?

- Második alkalom - Gyakorlat teszi a mestert
 - Tipikus REST API + Single Page Application kitelepítése Terraformmal
 - Rövid bemutató a használt cloud-os erőforrásokról
 - Újrafelhasználhatóság elemzése és kód finomítás

Képzés áttekintése, célkitűzés

Ott vagyunk már?

- Harmadik alkalom - Spagetti kódból karbantartható kód
 - Előző alkalommal fejlesztett infrastruktúra átalakítása modulok használatával
 - Hogyan szervezzük modulokba a kódunkat
 - Saját modul létrehozása
 - Közös munkamenet (backendekek) és stratégiák
 - Terraform Cloud

Képzés áttekintése, célkitűzés

Ott vagyunk már?

- Negyedik alkalom - Haladó Terraform rutinok és egzotikumok
 - Haladó Terraform témák
 - Tesztelés
 - Saját provider írása

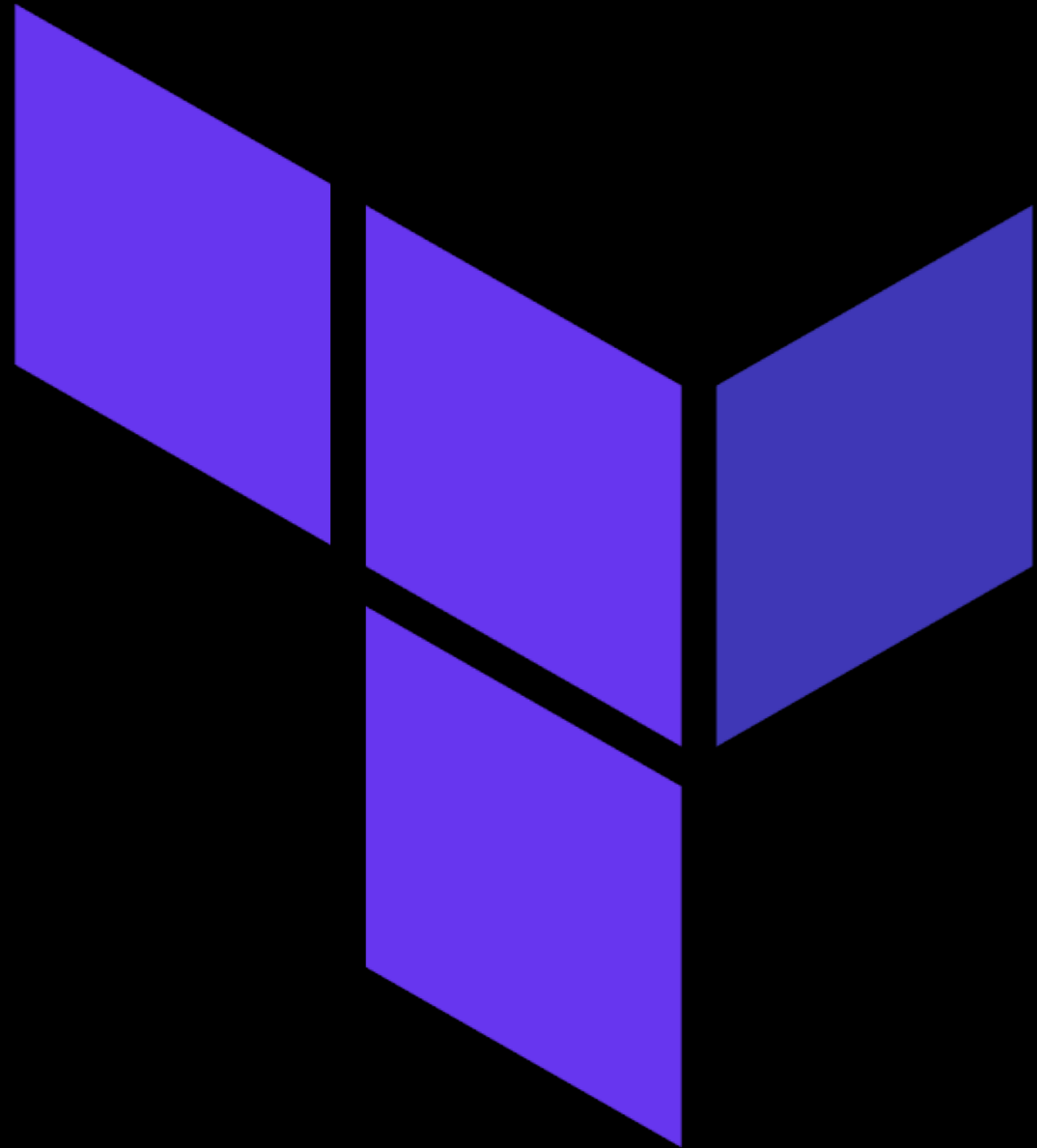


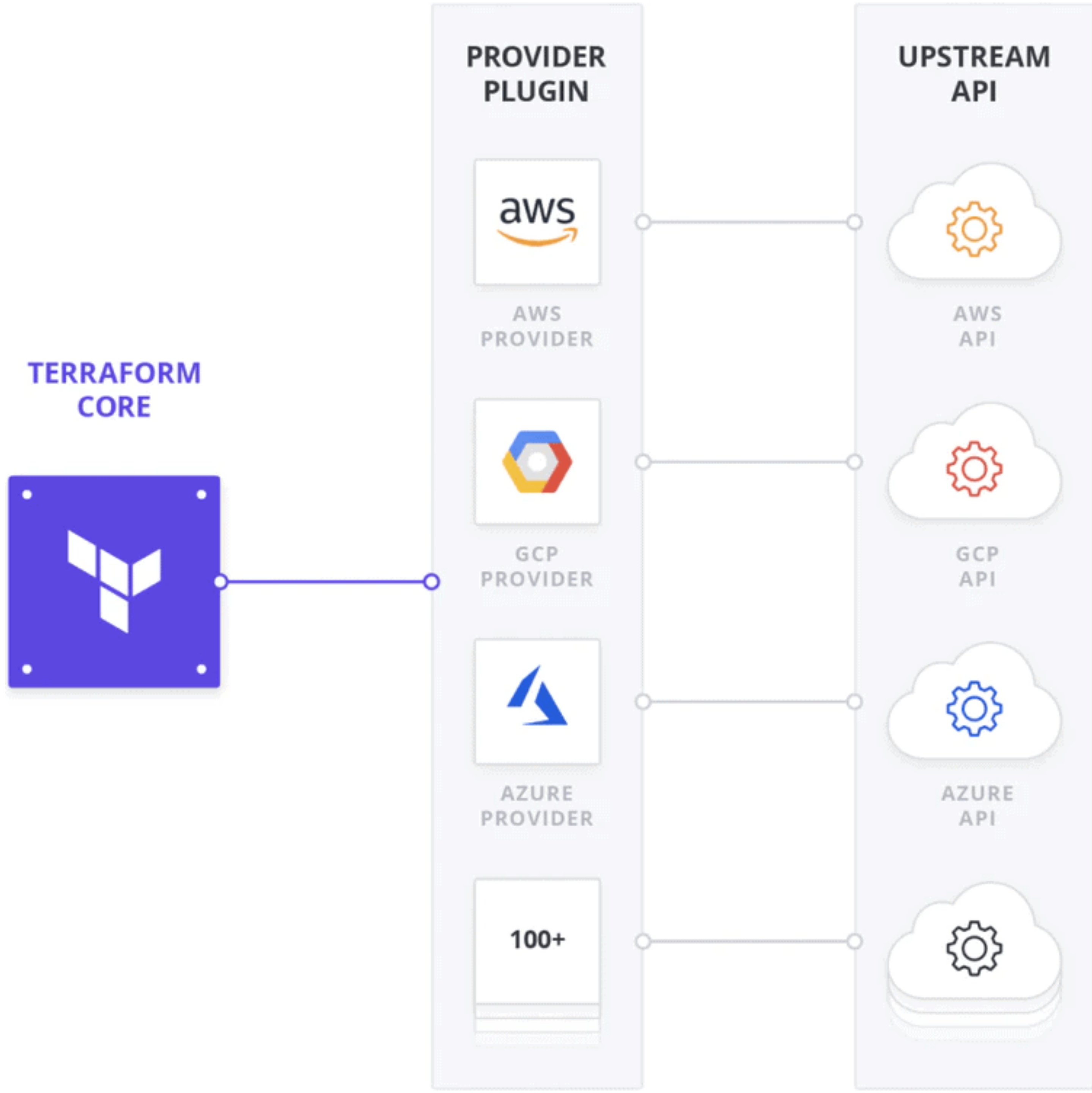
IT'S SHOW TIME

Helló Terraform!

Az IaC svájci bicskája

- Verziózható
- Deklaratív
- Barátságos leírónyelv (HCL)
- Rengeteg provider (és bővíthető)
- Beépített dry run
- Interpoláció
- Go alapú





BIG BANG THEORY

METRIC EXPANSION OF SPACE

The Big Bang theory is an effort to explain what happened at the very beginning of our universe. Discoveries in astronomy and physics have shown beyond a reasonable doubt that our universe did in fact have a beginning. Prior to that moment there was nothing; during and after that moment there was something: our universe. The big bang theory is an effort to explain what happened during and after that moment.



Big Bang. The universe burst open, expanding faster than the speed of light and flinging all the matter and energy in the universe apart in all directions.

13,6 billion years ago

First stars formed. Nuclear fusion lights up the stars.

13,2 billion years ago

First galaxies formed. Dense gas clouds collapse under their own gravity to eventually form galaxies.

Today

Stellar era. Dark energy prevails, the universe expands, becoming ever less dense.

after 10^{11} billion years

Under the influence of dark energy all galaxies outside the local group will pass behind the cosmological horizon.

after 10^{14} billion years

Star formation ends, leaving all stellar objects in the form of degenerate remnants. Black holes dominate the universe.

after 10^{60} billion years

The only organized units are black holes, but even black holes are unstable and evaporate into electrons and positrons.



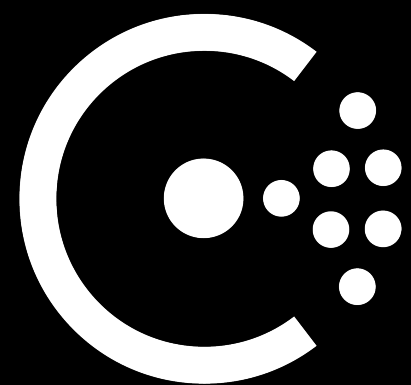
Terraform Provider-ek



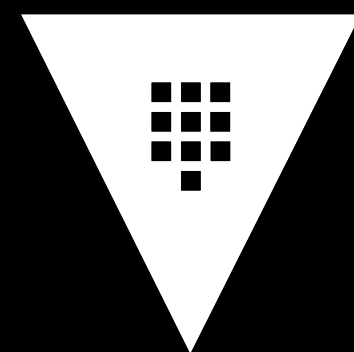
vmware®



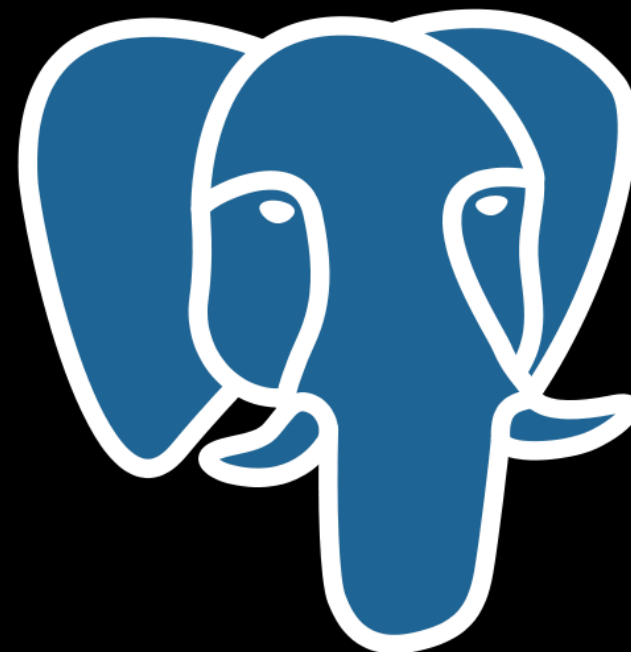
fastly®



HashiCorp
Consul



HashiCorp
Vault



Terraform erőforrások

Azaz a Terraform építő kockája

- Egy telepített provider által kezelt erőforrás (resource)
- Azonosítóval ellátott
- Változók jelenléte
- Referencia más erőforrásokra
- Implicit függőségi viszony
- Erősen típusos

```
resource "aws_instance" "iac_in_action" {  
    ami            = var.ami_id  
    instance_type  = var.instance_type  
    availability_zone = var.availability_zone  
  
    // dynamically retrieve SSH Key Name  
    key_name = aws_key_pair.iac_in_action.key_name  
  
    // dynamically set Security Group ID (firewall)  
    vpc_security_group_ids = [aws_security_group.iac_in_action.id]  
  
    tags = {  
        Name = "Terraform-managed EC2 Instance for IaC in Action"  
    }  
}
```

Terraform data providerek

Lekérdezések

- Nem általunk kezelt erőforrások tulajdonságainak lekérdezésére használható
- Szűrési lehetőségek

```
data "aws_ami" "example" {
  executable_users = ["self"]
  most_recent      = true
  name_regex       = "^myami-\\d{3}"
  owners           = ["self"]

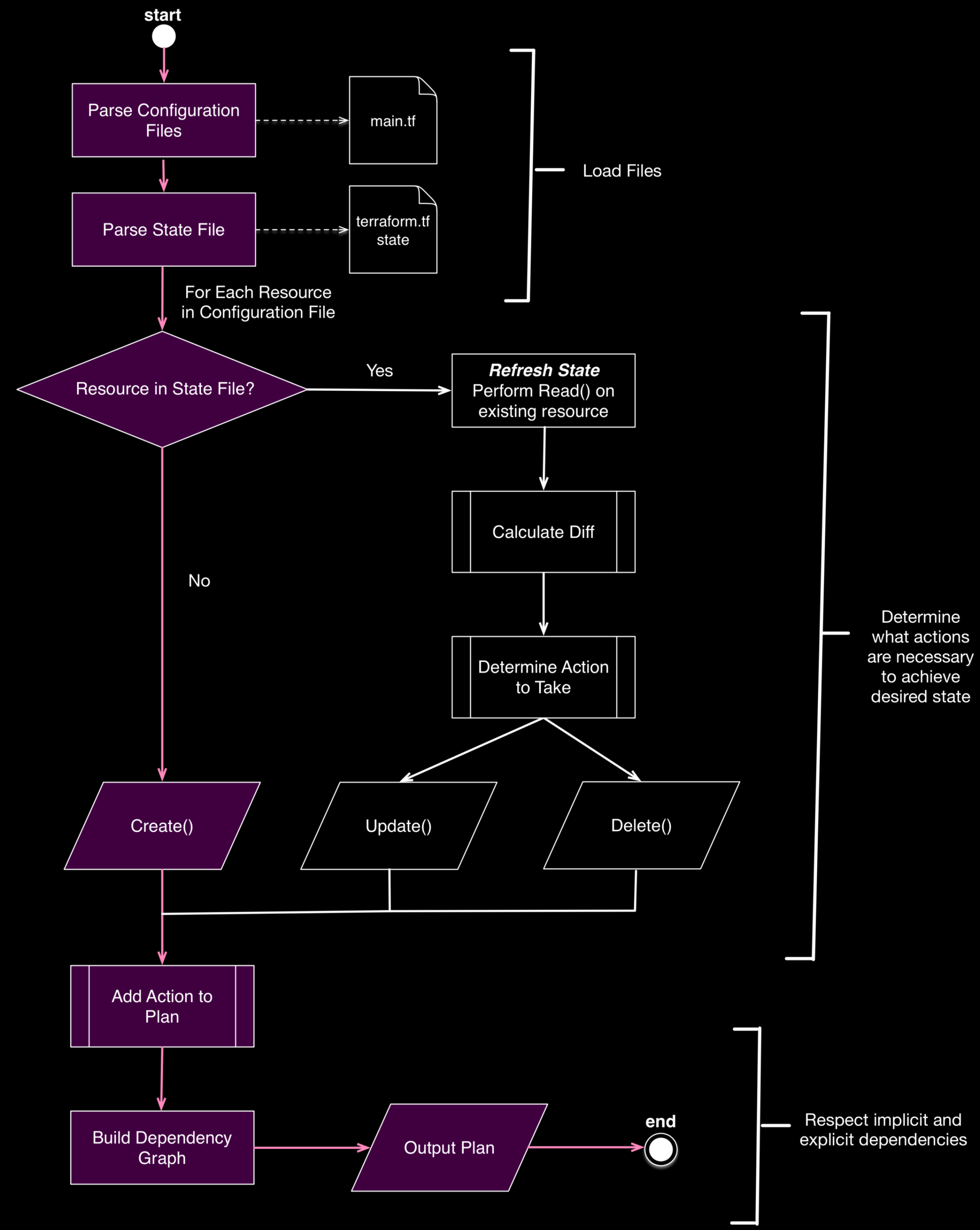
  filter {
    name   = "name"
    values = ["myami-*"]
  }

  filter {
    name   = "root-device-type"
    values = ["ebs"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }
}
```

Terraform - State

- A Terraform-nak szüksége van egy állapot file-ra
 - Egyfajta leltár, leképezés az eszköz által kezelt erőforrások és a valódi erőforrások között
 - Minden futtatás előtt frissítve van
- terraform.tfstate fájlban van tárolva (egyszerű esetben disken)
- Távoli tárolás (remote backend)
 - Engedélyezi a kollaboratív, párhuzamos fejlesztést.
Backendek: s3, consul, gcs, azurearm, artifactory, http
 - Párhuzamos futtatás esetén lock használata ajánlott.
Pl: Consul out-of-the box támogatja, S3+DynamoDB locking
 - Optimistic locking használata



Gyakori munkamenet és parancsok

- init
- validate
- plan
- apply
- destroy



Applicant



Identifying
Information



Public Key



Private Key
(Applicant)



1. Generate CSR

Certificate Authority (CA)



2. Validate Identity



3. Generate Signed
Certificate and Send
to Applicant



Private Key (CA)

