DevOps Tools na prática (IAC)

[AWS + Gitlab + terraform]

Porque Infraestrutura como código?

Porque Infraestrutura como código?

Há muito tempo em um ***Data Center*** distante, um antigo grupo de seres ***poderosos*** conhecidos como ***SysAmins***, criavam e gerenciavam a infraestrutura ***manualmente***.
Era uma era sombria e medonha se os administradores caíssem no lado escuro (ou seja, tirassem férias).
A boa notícia é:
- Graças à iniciativa ***DevOps*** agora temos uma maneira melhor de fazer as coisas:

Infrastructure-as-Code (IAC)

Benefícios da IAC

- Guardar todo o ***código*** da a sua ***infraestrutura*** em um versionador como por exemplo o ***Github/GitLab/Bitbucket***, isso torna mais fácil o Debug de problemas ou rollback se necessário

- Você pode validar a sua infraestrutura com ***code-reviews*** e ***testes automatizados***.

- Você pode ***reaproveitar*** o ***código***

Benefícios da IAC

- ***Automatizar*** todo o processo de provisionamento o que torna tudo mais ***rápido*** e ***resiliente*** do que processos manuais

- Possibilita o ***time de desenvolvimento*** a implementar modificações na infraestrutura de forma ***autônoma***

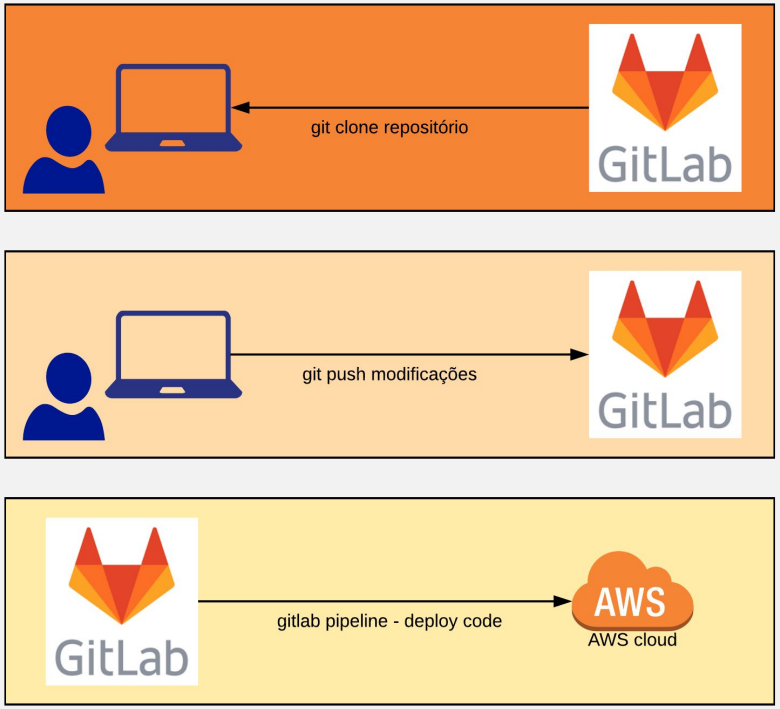
Tecnologias utilizadas

GitLab

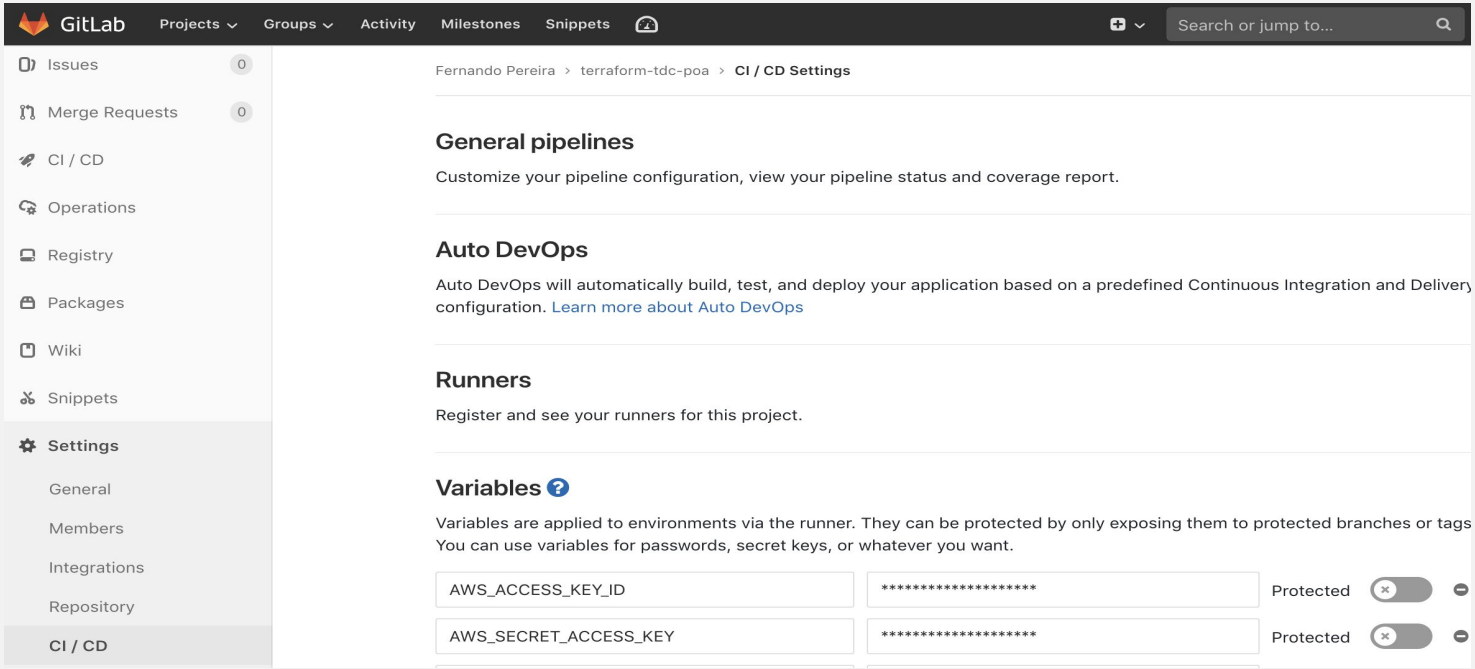
- gitlab *(repositório)*
- gitlab-ci/cd *(pipeline)*

- ***Multi Cloud*** (AWS, GCP, Azure)
- Escrito em **HCL**

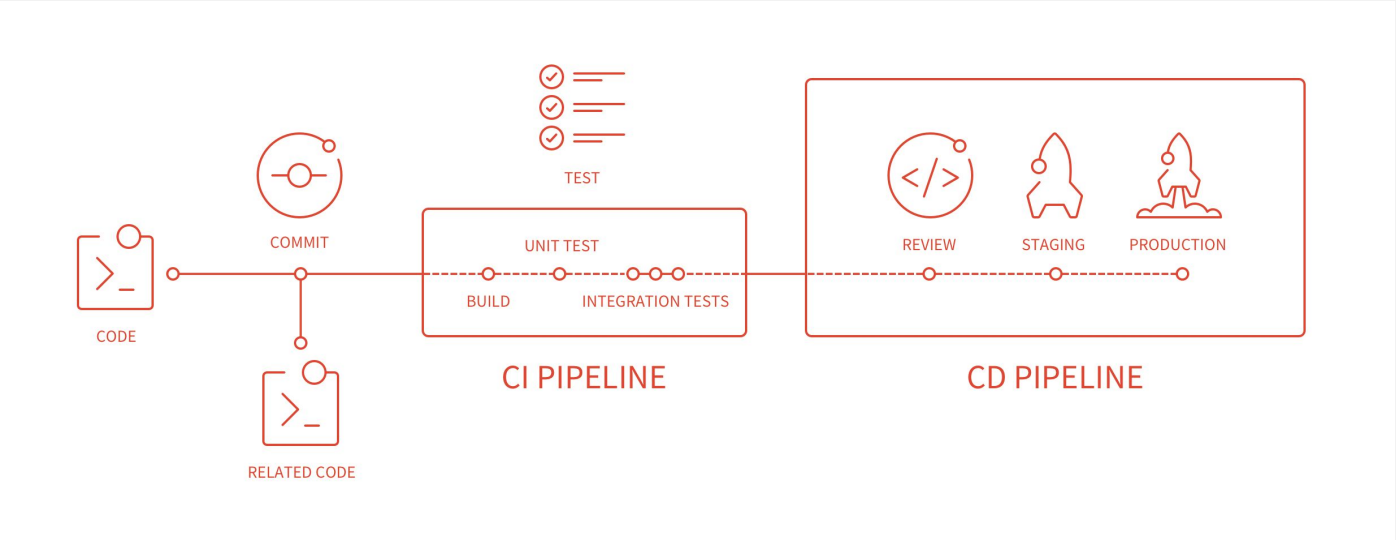
Fluxo de trabalho



Pré-Requisito: Gitlab Pipeline



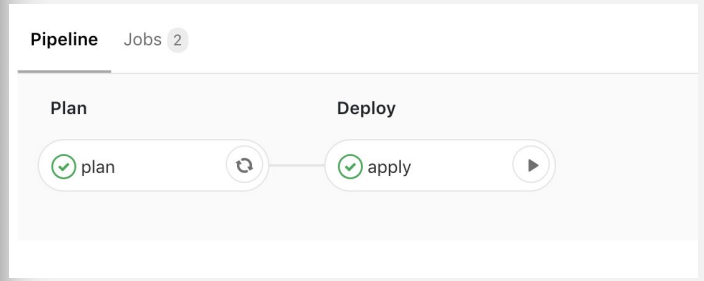
Fluxo Pipeline



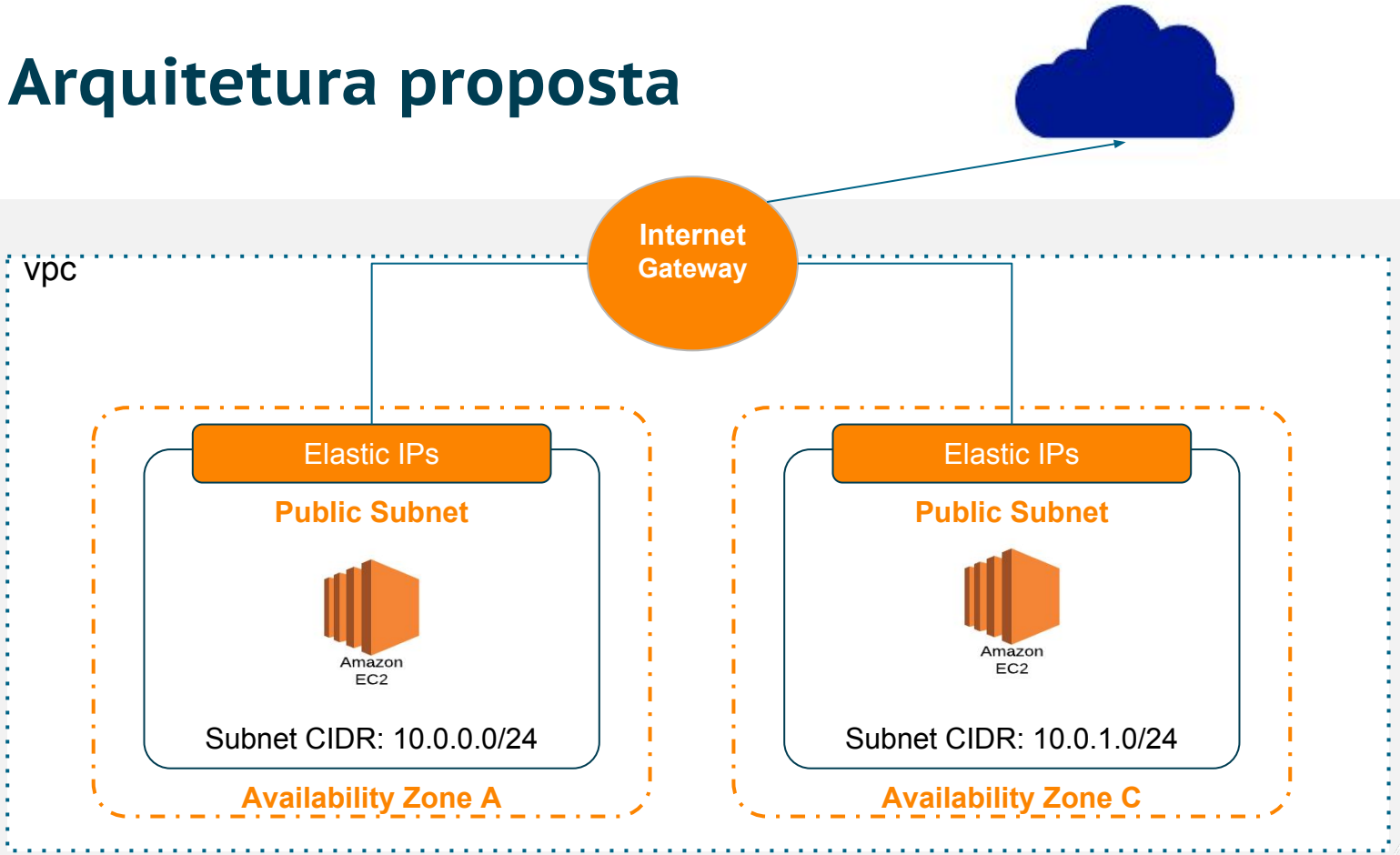
Gitlab Pipeline: .gitlab-ci.yml

2. ubuntu@ip-10-0-1-98: ~ (vim)

```
image:
  name: querodevops/gitlab-tf-build

before_script:
    - terraform init

stages:
  - plan
  - deploy

plan:
  stage: plan
  script:
    - terraform init
    - terraform plan

# Separate apply job for manual launching Terraform as it can be destructive
# action.
apply:
  stage: deploy
  environment:
    name: production
  script:
    - terraform init
    - terraform apply -auto-approve
".gitlab-ci.yml" 31L, 480C
```



Arquitetura proposta

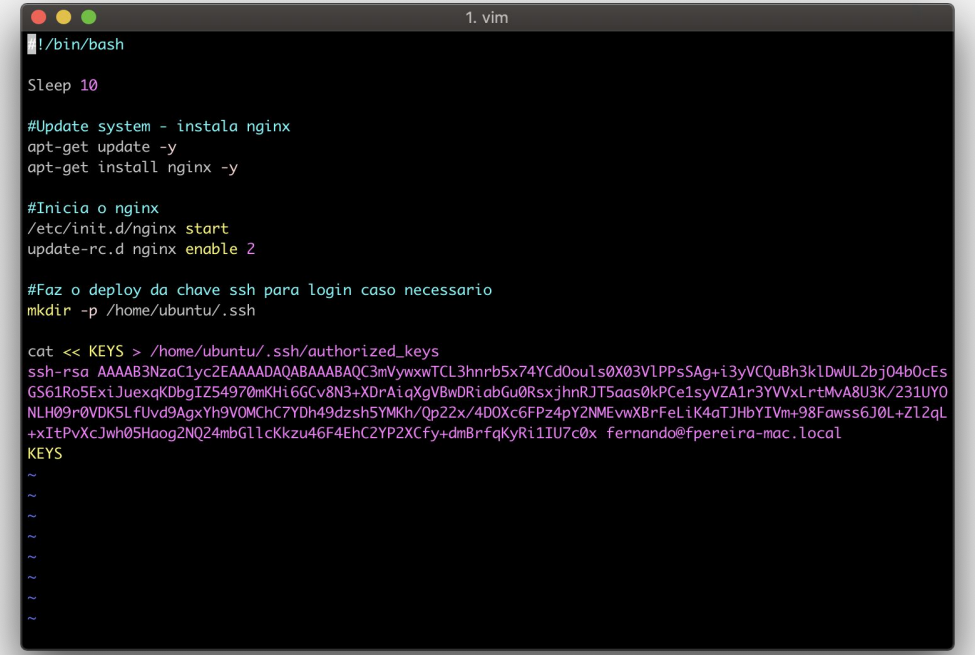


Código: terraform resources

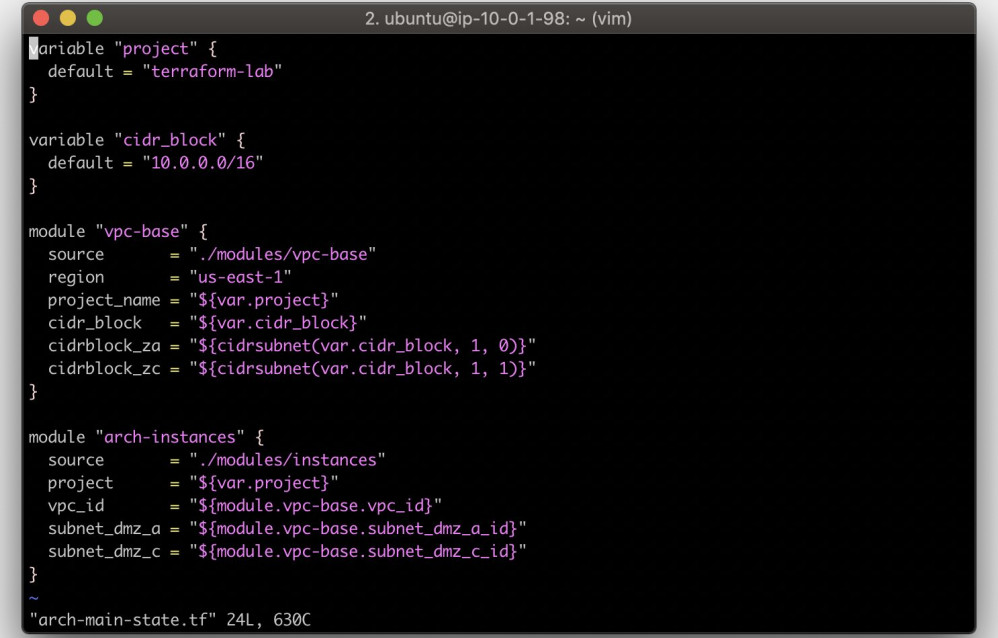
2. ubuntu@ip-10-0-1-98: ~ (vim)

```
resource "aws_instance" "ec2-a" {
  ami                         = "ami-0f9cf087c1f27d9b1"
  instance_type               = "t3.nano"
  vpc_security_group_ids      = ["${aws_security_group.sg_ec2_a.id}"]
  subnet_id                   = "${var.subnet_dmz_a}"
  associate_public_ip_address = true
  user_data                   = "${data.template_file.ec2_a.rendered}"

  tags {
    Name = "ec2-a"
  }
}

data "template_file" "ec2_a" {
  template = "${file("${path.module}/files/ec2-a.sh")}"
}

resource "aws_security_group" "sg_ec2_a" {
  name        = "sg_ec2_a"
  description = "Permite trafego para ssh na instancia"
  vpc_id      = "${var.vpc_id}"

  ingress {
    from_port   = 22
    to_port     = 22
```

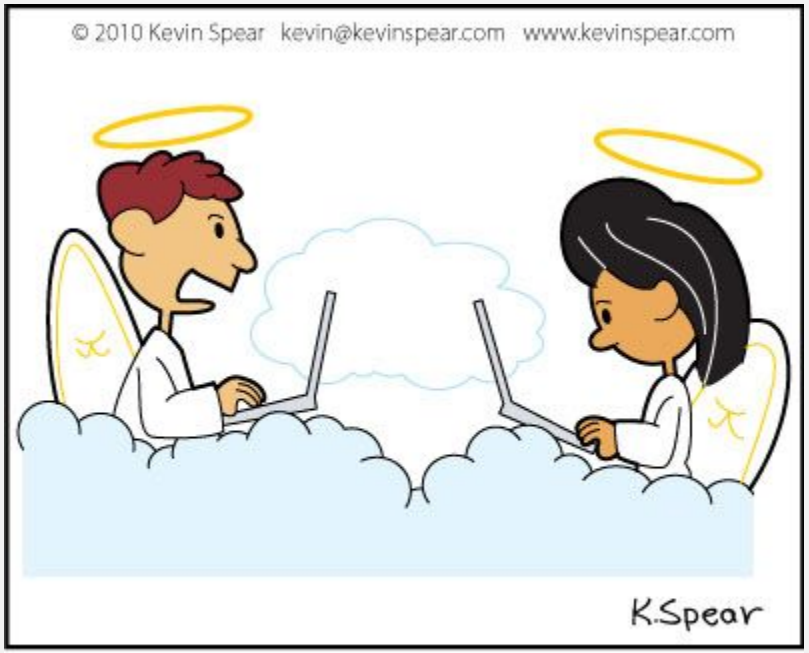
Código: ec2-userdata



Código: terraform modules



Que os Deuses das apresentações nos abençoem



"I'm glad they introduced cloud computing up here."

Quero Educação

- *Marketplace Universidades*
- *Desenvolve o site* ***querobolsa.com.br***
- ***+1200*** *universidades parceiras*
- ***+300 mil*** *vagas ativas no site*
- ***Desde 2012 +420 mil*** *alunos matriculados*
- ***+500mil*** *acessos em 1 dia*

Obrigado!

que.bo/vagas-dev

- Fernando Gonçalves Pereira

- Head of Infrastructure@***QueroEducação***

LinkedIn: *fernando-pereira-br* | **Github**: *pexaorj*