HashiCorp

Terraform

Marc Saez

ASIX 2

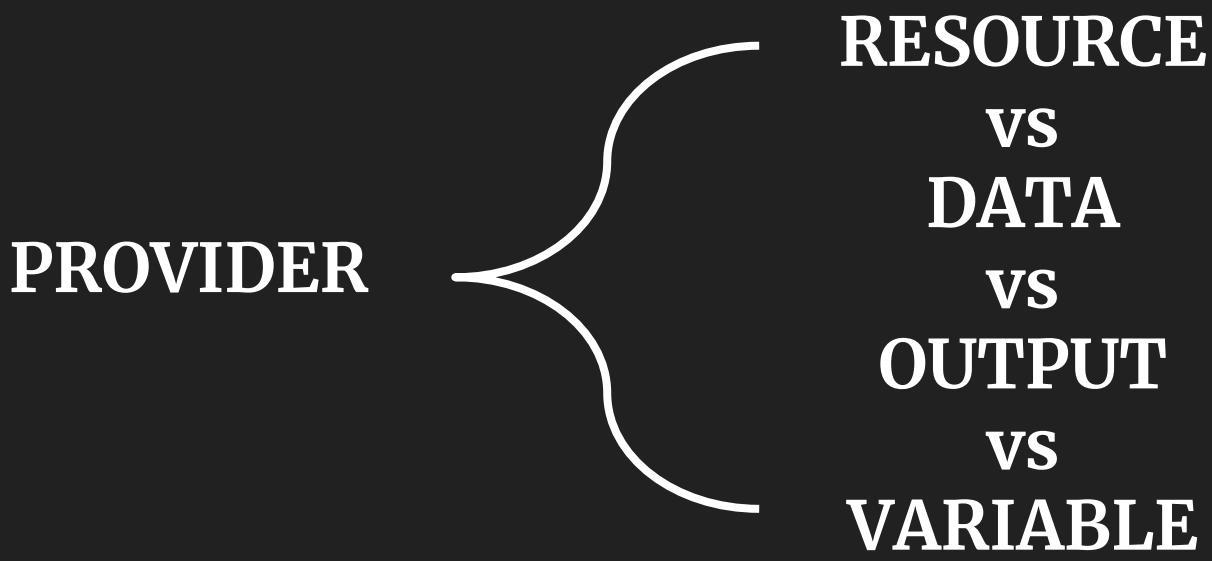
Projecte

Índex

Què és Terraform?

- Open Source
- IaC (Infrastructure as Code)
- CLOUD

Introducció recursos Terraform



PROVIDER

```
provider "aws" {
  region      = "us-east-1"
  profile     = "insti"
  access_key = ""
  secret_key = ""
}
```

RESOURCE

```
resource "aws_instance" "hosting" {
  ami           =  data.aws_ami.ubuntu
  instance_type = var.instance_type
  associate_public_ip_address = true
  tags = {
    "Name" = "msaez"
  }
  vpc_security_group_ids = [aws_security_group.hosting.id]
  key_name = aws_key_pair.marc.key_name
}
```

https://registry.terraform.io/providers/hashicorp/aws/latest/docs

DATA

```
data "aws_ami" "ubuntu" {
  most_recent = true
  owners      = ["099720109477"] # Canonical

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-*"]
  }

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

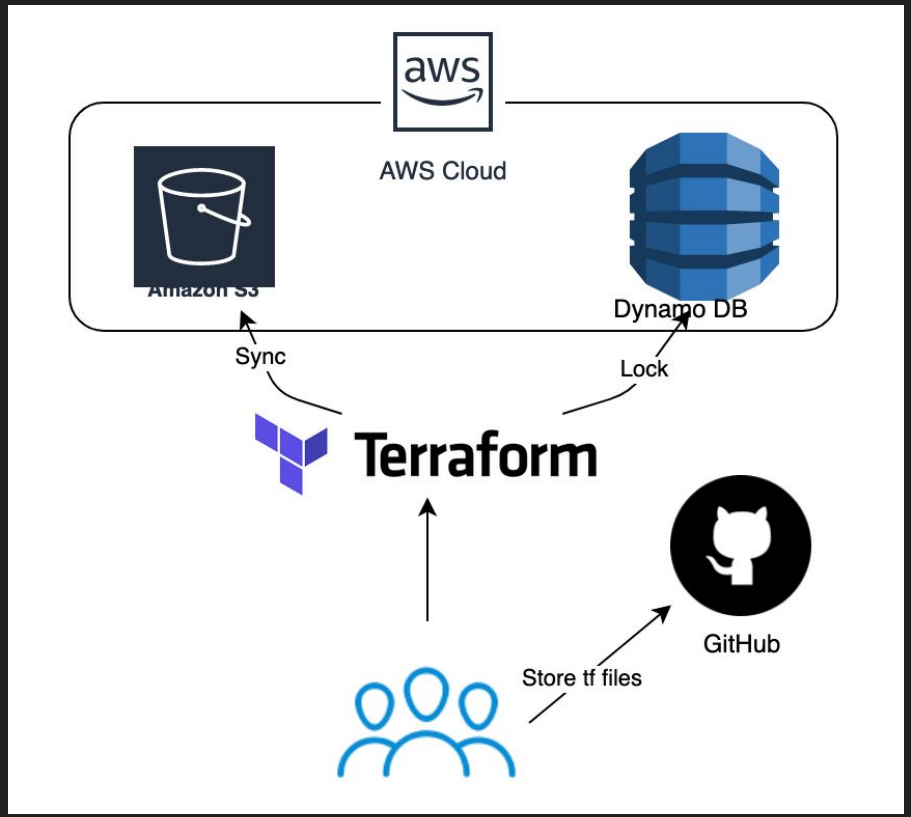
OUTPUT

```
output "ip_addr" {
  value       = aws_instance.rancher.public_ip
}

output "http_link" {
  value       = "http://${aws_instance.rancher.public_ip}"
  description = "HTTP Link Address"
}
```

VARIABLE

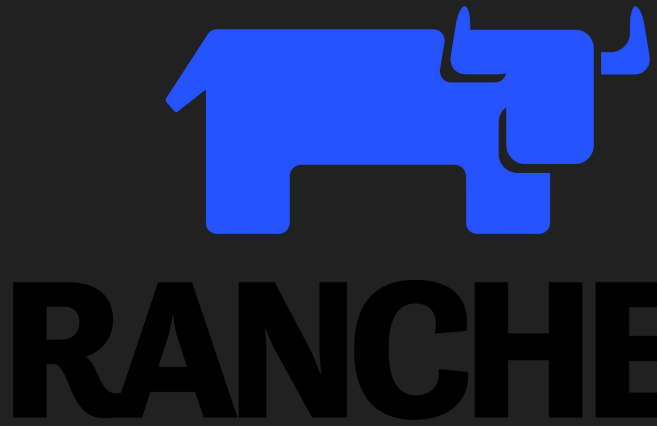
```
variable "example" {
  type = string
  description = "Variable d'exemple"
  default = ""
}
```

Terraform com equip




```
terraform {
  backend "s3" {
    bucket         = "terraform-state-msaez"
    key            = "global/s3/terraform.tfstate"
    region         = "us-east-1"
    profile        = "insti"
    dynamodb_table = "terraform-locks"
    encrypt        = true
  }
```

Objectius



APACHE
HTTP SERVER PROJECT

Rancher

EC2

```
resource "aws_instance" "rancher" {
  ami = data.aws_ami.ubuntu.id
  instance_type = var.instance_type
  key_name = aws_key_pair.rancher_key_pair.key_name
  subnet_id = "subnet-00f67cae2874a3204"
  security_groups = [aws_security_group.rancher_sg_allowall.id]
  user_data_base64 = base64encode(data.template_file.cloud-init-cc
  ebs_optimized = true
  iam_instance_profile = "LabInstanceProfile"
  root_block_device {
    volume_type = "gp2"
    volume_size = 16
    encrypted = true
  }
```

Security Group

```
resource "aws_security_group" "rancher_sg_allowall" {
  name        = "rancher_sg"
  description = "Rancher"

  ingress {
    from_port   = "80"
    to_port     = "80"
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
    description = ""
  }
  egress {
    from_port   = "0"
    to_port     = "0"
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

CLOUD-INIT

```
#cloud-config
ssh_authorized_keys:
  - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDpJGWF4vUpSyZgsT6/uylFwecdaL4six8YOgy2PmPnJCmwwbI6jUuPy8LsWfc7paRsc976/
  8IxUGOxlxQ7EucMrz5NrfPhjUv6DDztEKNTFg2moSCdReNuBNqjmdSd1Z68uVmMFqSMfbLfds7c+Kib3UF/
  VreKDS5nKs6gjLlQhjwbWtBW8WAAX2c0kqr6UbiGhYPDF2eiGLjPa6donhYxdaFZwN0cIVMWY48WX51Ptd9qJTKvtE10ZOgkmKy3LZK56+V0IJ81UC/
  DWEmrS0nHU2lolY8vsYrDSftf+krpvhKbx76FulPV8ZGRyxbOaNU/OhifyjZ6WTpAOmdOyy9AgjTNvx8/UIdOQkKeaHMuxwRWT7FIECziM+TqCLChjrv
  +RpZvIPuLKLLoX/AS1FsZbd6zNN9sHfNTQMTRqfEjE+5OIlteK8VuMMM5t1QyZeusb8/Ouz0YDA5vMuxTUFxucwP7jm+FFcU4jZjgPsxz2WUPedqJgeUluuXmh/
  xpFFE= austria@austria-Lenovo-V14-ADA

# Esta parte escribe el archivo rancher.service el el siguiente path
write_files:
  - path: /etc/systemd/system/rancher.service
    permissions: 0644
    owner: root
    content: |
      [Unit]
      Description=rancher-server como servicio
      Requires=docker.service
      After=docker.service

      [Service]
      Restart=on-failure
      RestartSec=10
      ExecStart=/usr/bin/docker run --name %p --rm --privileged -p 80:80 -p 443:443 -e CATTLE_BOOTSTRAP_PASSWORD=admin -v /opt/
      rancher:/var/lib/rancher rancher/rancher:stable
      ExecStop=-/usr/bin/docker stop -t 2 %p

      [Install]
      WantedBy=multi-user.target
runcmd:
  - export DEBIAN_FRONTEND=noninteractive
  - curl -sL https://releases.rancher.com/install-docker/${docker_version}.sh | sh
  - sudo usermod -aG docker ${username}
  - sudo systemctl start rancher
```

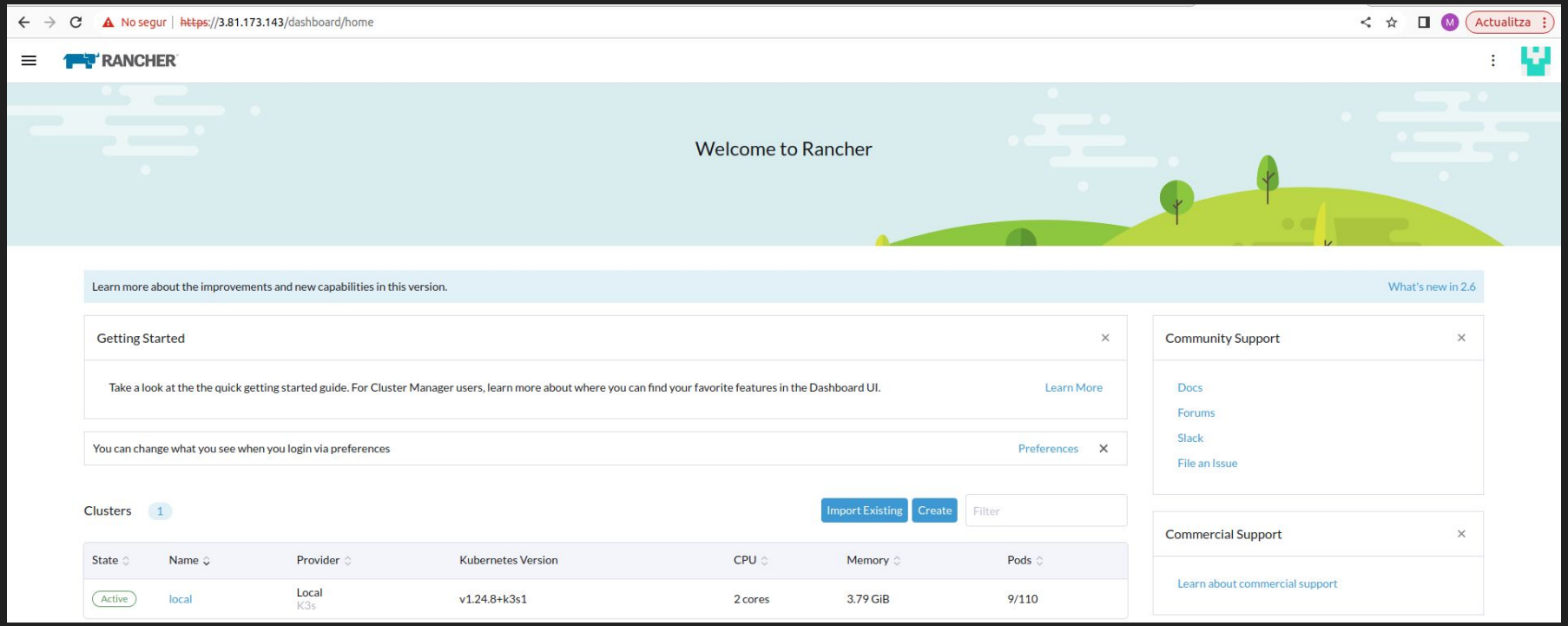
You have Docker installed on your system. Do you

OUTPUTS

```
output "ip_addr" {
  value       = aws_instance.rancher.public_ip
}

output "http_link" {
  value       = "http://${aws_instance.rancher.public_ip}"
  description = "HTTP Link Address"
}
```

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

http_link = "http://3.81.173.143"
ip_addr = "3.81.173.143"
```

DEMO


No segur | https://3.81.173.143/dashboard/home
Actualitza
RANCHER
Welcome to Rancher
Learn more about the improvements and new capabilities in this version.
What's new in 2.6
Getting Started
Take a look at the the quick getting started guide. For Cluster Manager users, learn more about where you can find your favorite features in the Dashboard UI.
Learn More
You can change what you see when you login via preferences
Preferences
Clusters 1
Import Existing
Create
Filter
State
Name
Provider
Kubernetes Version
CPU
Memory
Pods
Active
local
Local
K3s
v1.24.8+k3s1
2 cores
3.79 GiB
9/110
Community Support
Docs
Forums
Slack
File an Issue
Commercial Support
Learn about commercial support

APACHE

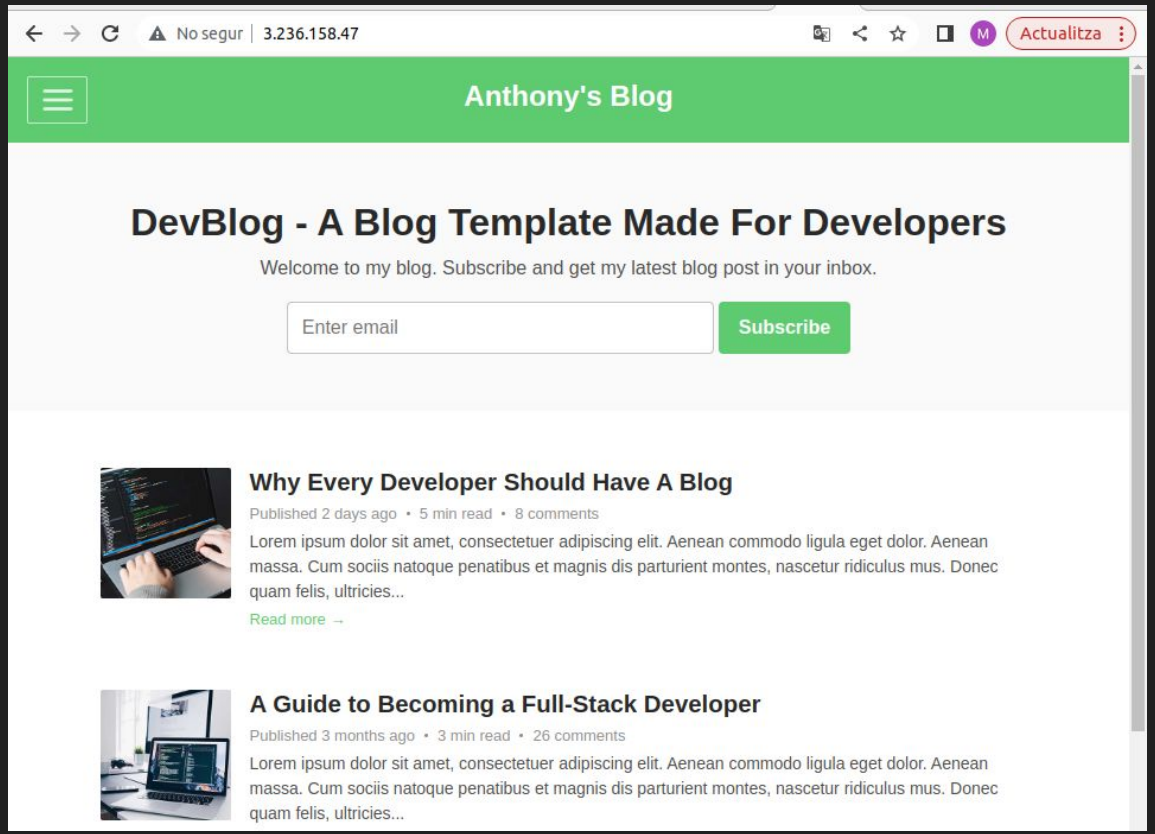
EC2

```
resource "aws_instance" "hosting" {
  ami           = "ami-007855ac798b5175e"#data.aws_ami.ubuntu.id
  instance_type = var.ec2_type
  #key_name      = aws_key_pair.marc.key_name
  vpc_security_group_ids = [aws_security_group.hosting.id]
  associate_public_ip_address = true
  user_data_base64 = base64encode(data.template_file.cloud-init-config.rendered)
  tags = {
    "Name" = "${var.name}hosting"
  }
  provisioner "remote-exec" {
    inline = [
      "echo 'Waiting for cloud-init to complete...'",
      "cloud-init status --wait > /dev/null",
      "echo 'Completed cloud-init!'",
      "echo 'Apache Server ready!'",
    ]
    connection {
      type = "ssh"
      host = self.public_ip
      user = "ubuntu"
      private_key = "${file("~/.ssh/id_rsa")}"
    }
  }
}
```

NULL RESOURCE

```
resource "null_resource" "web1" {
  triggers = {
    instance_id = data.terraform_remote_state.main.outputs.ip_ec2
  }
  connection {
    type     = "ssh"
    user     = "ubuntu"
    host     = "${data.terraform_remote_state.main.outputs.ip_ec2}"
    private_key = "${file("~/.ssh/id_rsa")}"
  }
  provisioner "file" {
    source = "./webs/DevBlog-Theme-master"
    destination = "/home/ubuntu/"

  }
  provisioner "remote-exec" {
    inline = [
      "ls -l ~/DevBlog-Theme-master",
      "sudo cp -r ~/DevBlog-Theme-master/* /var/www/web2/"

    ]
  }
}
```

DEMO




```
Outputs:

ip_ec2 = "3.236.158.47"
```