terraform win-instance

de interés

- uso **key** para recuperar contraseña admin windows
- asignación IP privada a dedo
- IP pública variable

código

main.tf

```
resource "aws_vpc" "my_vpc" {
  cidr_block = "172.16.0.0/16"

  tags = {
    Name = "tf-example"
  }
}

resource "aws_subnet" "my_subnet" {
  vpc_id            = "${aws_vpc.my_vpc.id}"
  cidr_block        = "172.16.10.0/24"
  availability_zone = "${var.az1}"
  map_public_ip_on_launch = true

  tags = {
    Name = "tf-example"
  }
}

resource "aws_internet_gateway" "igw_main" {
  vpc_id = "${aws_vpc.my_vpc.id}"

  tags {
    Name = "IGW-MYAPP"
  }

  depends_on = ["aws_vpc.my_vpc"]
}

# resource "aws_nat_gateway" "natgw_az1" {
#   allocation_id = "${aws_eip.eip_natgw_az1.id}"
#   subnet_id     = "${aws_subnet.my_subnet.id}"

#   depends_on = ["aws_internet_gateway.igw_main"]
# }

# resource "aws_eip" "eip_natgw_az1" {
#   vpc = true
# }
```

```
# resource "aws_network_interface" "foo" {
#   subnet_id   = "${aws_subnet.my_subnet.id}"
#   private_ips = ["172.16.10.100"]

#   tags = {
#     Name = "primary_network_interface"
#   }
# }

resource "aws_instance" "foo" {
  disable_api_termination = "${var.vm_adwriter_disable_api_termination}"
  instance_type           = "${var.vm_adwriter_instance_type}"
  ami                     = "${var.vm_adwriter_image}"
  subnet_id               = "${aws_subnet.my_subnet.id}"
  key_name                = "${aws_key_pair.foo.key_name}"
  get_password_data       = true

  # network_interface {
  #   network_interface_id = "${aws_network_interface.foo.id}"
  #   device_index         = 0
  # }

}

resource "tls_private_key" "foo" {
  algorithm = "RSA"
  rsa_bits  = 4096
}

resource "aws_key_pair" "foo" {
  key_name   = "foo-kp"
  public_key = "${tls_private_key.foo.public_key_openssh}"
}

resource "aws_iam_instance_profile" "instance_profile_adwriter" {
  name = "INSTANCE_PROFILE_ADWRITER"
  role = "${aws_iam_role.iam_role_adwriter.name}"
}

resource "aws_iam_role" "iam_role_adwriter" {
  name = "IAM_ROLE_ADWRITER"
  path = "/"

  assume_role_policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Principal": {
               "Service": "ec2.amazonaws.com"
            },
            "Effect": "Allow",
            "Sid": ""
```

```
        }
    ]
}
EOF
}

resource "aws_security_group" "secgroup_foo" {
  name = "SECGROUP-FOO"

  vpc_id = "${aws_vpc.my_vpc.id}"

  ingress {
    from_port = 1
    to_port   = 65535
    protocol  = "tcp"

    cidr_blocks = [
      "${var.trusted_ip_address}",
    ]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags {
    Name = "SECGROUP-ADWRITER"
  }
}
```

output.tf

```
output "ec2_public_dns" {
  value = "${aws_instance.foo.public_ip}"
}
output "ec2_password" {
  value = "${rsadecrypt(aws_instance.foo.password_data,
file("${path.cwd}/ec2_foo.key"))}"
}

output "Private KEY" {
  value = "${tls_private_key.foo.private_key_pem}"
}

resource "local_file" "foo" {
    content  = "${tls_private_key.foo.private_key_pem}"
    filename = "${path.cwd}/ec2_foo.key"
}
```

variables.tf

```
# AWS credentials
# variable "provider_aws_access_key" { }
# variable "provider_aws_secret_key" { }
variable "provider_aws_zone" {}

# Availability zones
variable "az1" {}

variable "az2" {}

# VPC
variable "vpc_cidr" {}

# Access
variable "trusted_ip_address" {}

# Directory Service
variable "dir_domain_name" {}

variable "dir_admin_password" {}
variable "dir_type" {}
variable "dir_computer_ou" {}

# AD Writer machine
variable "vm_adwriter_disable_api_termination" {}

variable "vm_adwriter_instance_type" {}
variable "vm_adwriter_image" {}
```

terraform.tfvars

```
# AWS credentials
provider_aws_zone = "us-east-1"

# Availability zones
az1 = "us-east-1a"
az2 = "us-east-1b"

# VPC
vpc_cidr = "10.1.0.0/16"

trusted_ip_address  = "213.151.119.65/32"

# Directory Service
dir_domain_name     = "myapp.com"
dir_admin_password  = "Sup3rS3cret"
dir_type            = "MicrosoftAD"
dir_computer_ou     = "OU=myapp,DC=myapp,DC=com"

# AD Writer machine
vm_adwriter_disable_api_termination = false
vm_adwriter_instance_type           = "t2.medium"
vm_adwriter_image                   = "ami-0bf148826ef491d16"
```