

for_each, toset, tomap, AZ zone

`for_each` Meta-Argument

`toset` function <https://www.terraform.io/language/functions/toset>

`tomap` function <https://www.terraform.io/language/functions/tomap>

Data Source: `aws_availability_zones`

`aws_availability_zones` 就是region下面有多个数据中心

在一个region下的每一个支持t2.micro的数据中心，都各自创建一个实例

1, 在一个region下的每一个数据中心，都各自创建一个实例可能会产生问题，就是有的数据中心

不支持创建 t2.micro 实例，terminal里先用一下命令查看一下

ap-northeast-1 支持t2.micro 的数据中心

```
aws ec2 describe-instance-type-offerings --location-type availability-zone --filters Name=instance-type,Values=t2.micro --region ap-n
```

```
zhanghaifeng@zhangtekiMacBook-Pro ~ % aws ec2 describe-instance-type-offerings --location-type availability-zone --filters Name=instance-type,Values=t2.micro --region ap-northeast-1 --output table
```

DescribeInstanceTypeOfferings		
InstanceTypeOfferings		
InstanceType	Location	LocationType
t2.micro	ap-northeast-1c	availability-zone
t2.micro	ap-northeast-1a	availability-zone
t2.micro	ap-northeast-1d	availability-zone

```
zhanghaifeng@zhangtekiMacBook-Pro ~ %
```

5, get-instancetype-supported-per-az-in-a-region.tf

获取region里面所有支持t2.micro的数据中心名

```
// 获取某个region下的所有 数据中心的名字，记住就好
data "aws_availability_zones" "my_azones" {
  filter {
    name     = "opt-in-status"
    values   = ["opt-in-not-required"]
  }
}

# 如果 ap-northeast-1 下的数据库中心支持 t2.micro，就返回包含数据中心名的数组
# 如果 ap-northeast-1 下的某个数据库中心不支持 t2.micro，就返回空数组
data "aws_ec2_instance_type_offerings" "my_ins_type" {
  for_each = toset(data.aws_availability_zones.my_azones.names)
  filter {
    name     = "instance-type"
    values   = ["t2.micro"]
  }
  filter {
    name     = "location"
    values   = [each.value]
  }
  location_type = "availability-zone"
}
```

```

# 打印出所有数据中心名-->是否支持t2.micro
# 支持就会显示如 ap-northeast-1a => ["t2.micro"]
# 不支持就会显示如 ap-northeast-1b => []
output "output_v3_1" {
  value = { for az, details in data.aws_ec2_instance_type_offerings.my_ins_type :
    az => details.instance_types }
}

# 打印出去除不支持t2.micro的数据中心
output "output_v3_2" {
  value = { for az, details in data.aws_ec2_instance_type_offerings.my_ins_type :
    az => details.instance_types if length(details.instance_types) != 0 }
}

#获取map的所有key, 也就是获取所有支持t2.micro的数据中心名
output "output_v3_3" {
  value = keys({ for az, details in data.aws_ec2_instance_type_offerings.my_ins_type :
    az => details.instance_types if length(details.instance_types) != 0 })
}

```

2, ec2-setting.tf

在一个region下的每一个支持t2.micro的数据中心, 都各自创建一个实例

```

resource "aws_instance" "myec2vm" {
  ami = data.aws_ami.amzlinux2.id
  instance_type = var.instance_type_map["dev"] //reference map value
  user_data = file("${path.module}/nginx-install.sh")
  key_name = var.instance_keypair
  vpc_security_group_ids = [aws_security_group.vpc-ssh.id, aws_security_group.vpc-web.id]

  # 在一个region下的每一个支持t2.micro的数据中心, 都各自创建一个实例
  for_each = toset( keys({ for az, details in data.aws_ec2_instance_type_offerings.my_ins_type :
    az => details.instance_types if length(details.instance_types) != 0 }) )

  availability_zone = each.key # You can also use each.value because for list items each.key == each.value

  tags = {
    "Name" = "For-Each-Demo-${each.key}"
  }
}

```

3, output-values.tf 文件

所有Instance 被建立后打印出与instance相关的ip, dns信息

由于aws_instance.myec2vm[*].public_dns 不能跟for_each 一起使用

所以output里面要用for...in...

```

# EC2 Instance Public IP with TOSET
output "instance_publicip" {
  description = "EC2 Instance Public IP"
  #value = aws_instance.myec2vm[*].public_ip # Latest Splat
  value = toset([ for myec2vm in aws_instance.myec2vm : myec2vm.public_ip])
}

# EC2 Instance Public DNS with TOSET
output "instance_publicdns" {
  description = "EC2 Instance Public DNS"
  #value = aws_instance.myec2vm[*].public_dns # Latest Splat
  value = toset([for myec2vm in aws_instance.myec2vm : myec2vm.public_dns])
}

# EC2 Instance Public DNS with MAPS
output "instance_publicdns2" {
  value = tomap({
    for az, myec2vm in aws_instance.myec2vm : az => myec2vm.public_dns
    # az intends to be a availability zone
    # for s, myec2vm in aws_instance.myec2vm : s => myec2vm.public_dns
    # s intends to be a subnet ID
  })
}

```

4, 4-for_each, toset,tomap,AZ zone文件夹里执行命令

```
terraform init      // 自动根据terraform-setting.tf 生成 [.terraform]跟[.terraform.lock.hcl] 两个文件

terraform validate // 验证配置是否正确

terraform plan      // 列出将要做的事,配置是什么

terraform apply -auto-approve

// 执行这条命令后, ec2实例建立、控制台会打印出下图的内容, output-values.tf 文件定义的内容
```

5, 验证是否正确建立,查看控制台, 检查网址

```
检查网址
http://18.181.244.242/index.html
http://18.181.244.242/app1/index.html
http://18.181.244.242/app1/metadata.html
```

6,终结ec2实例,aws管理画面的实例跟security groups会被删除

```
# Terraform Destroy
terraform destroy -auto-approve

# Clean-Up Files
rm -rf .terraform*
rm -rf terraform.tfstate*
```

github link

https://github.com/no744634936/terraform/tree/main/4-for_each-to-set-to-map-AZ-zone