Provisioners

Provisioners can be used to model specific actions on the local machine or on a remote machine in order to prepare servers or other infrastructure objects for service. Provisioners are used for executing scripts or shell commands on a local or remote machine as part of resource creation/deletion. They are similar to “EC2 instance user data” scripts that only run once on the creation and if it fails terraform marks it tainted... Terraform doesn't run these scripts multiple times.Most provisioners require access to the remote resource via SSH or WinRM, and expect a nested connection block with details about how to connect.

*These are the types of Generic provisioners (file, local exec, remote exec).

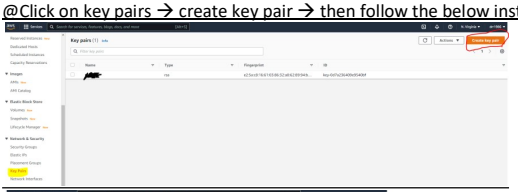
file provisioner is used to copy files or directories to a remote resource. We can’t provide any arguments to script in remote-exec provisioner. We can achieve this by copying script from file provisioner and then execute a script using a list of commands.

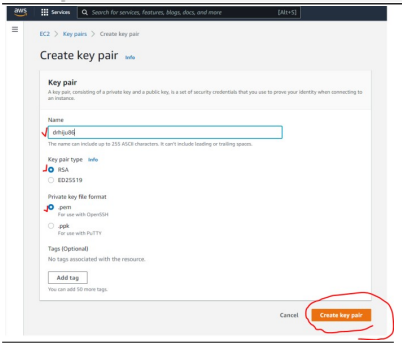
local-exec provisioner helps run a script on instance where we are running our terraform code, not on the resource we are creating. For example, if we want to write EC2 instance IP address to a file, then we can use below local-exec provisioner with our EC2 resource and save it locally in a file.

remote-exec provisioner helps invoke a script on the remote resource once it is created. We can provide a list of command strings which are executed in the order they are provided. We can also provide scripts with a local path which is copied remotely and then executed on the remote resource

Step 1:- Make sure to create key pairs on AWS console.

@Click on key pairs → create key pair → then follow the below instructions.





Step 2:- example with Remote-exec

#mkdir remote_exec

#cat providers.tf

```
[root@porali remote-exec]# ca
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aw
      version = "~>3.0"
    }
  }
}
```

#cat variable.tf

```
[root@porali remote-exec]# ca
variable "region" {
  type    = string
  default = "us-east-1"
```

#cat resource.tf

```
[root@porali remote-exec]# ca
data "aws_ami" "centos" {
  most_recent = true
  owners      = ["12552308842

  filter {
    name   = "name"
    values = ["CentOS 7.9.200
  }

  filter {
    name   = "architecture"
    values = ["x86_64"]
  }
  filter {
    name   = "virtualization-
    values = ["hvm"]
  }
}

resource "aws_instance" "web"
  ami           = data.aws_am
  instance_type = "t2.micro"
  key_name      = "drhiju86"

  provisioner "remote-exec" {
    inline = [
      "sudo yum update -y",
      "sudo yum install -y ep
```

#ls –lrt

```
[root@porali remote-exec]# ls -lrt
total 16
-rw-r--r-- 1 root          root          204 Oct
-rw-r--r-- 1 root          root           65 Nov
-rw-rw-r-- 1 sarapettap67 sarapettap67 1674 Dec
```

#terraform init && terraform fmt && terraform validate && terraform plan
#terraform apply

```
Do you want to perform these actions?
  Terraform will perform the actions o
  Only 'yes' will be accepted to appro

  Enter a value: yes
```

@Connection output from terraform apply

```
aws_instance.web: Provisioning with 'remote-exec'...
aws_instance.web (remote-exec): Connecting to remote host via SSH...
aws_instance.web (remote-exec):   Host: 174.129.167.53
aws_instance.web (remote-exec):   User: centos
aws_instance.web (remote-exec):   Password: false
aws_instance.web (remote-exec):   Private key: true
aws_instance.web (remote-exec):   Certificate: false
aws_instance.web (remote-exec):   SSH Agent: false
aws_instance.web (remote-exec):   Checking Host Key: false
aws_instance.web (remote-exec):   Target Platform: unix
```

```
aws_instance.web (remote-exec):   Target Platform: unix
aws_instance.web (remote-exec): Connected!
aws_instance.web: Still creating... [2m10s elapsed]
aws_instance.web (remote-exec): Loaded plugins: fastestmirror
aws_instance.web (remote-exec): Determining fastest mirrors
aws_instance.web (remote-exec):  * base: download.cf.centos.org
aws_instance.web (remote-exec):  * extras: download.cf.centos.org
aws_instance.web (remote-exec):  * updates: download.cf.centos.org
aws_instance.web (remote-exec): base                     | 3.6 kB     00:00
aws_instance.web (remote-exec): extras                   | 2.9 kB     00:00
aws_instance.web (remote-exec): updates                  | 2.9 kB     00:00
aws_instance.web (remote-exec): (2/4): base/7/x 0%       |    0 B     --:-- ETA
aws_instance.web (remote-exec): (1/4): base/7/x86        | 153 kB     00:00
aws_instance.web (remote-exec): (2/4): extras/7/x8       | 243 kB     00:00
aws_instance.web (remote-exec): (3/4): updates/7/x       |  13 MB     00:00
aws_instance.web (remote-exec): (4/4): base/7/ 70%       |  13 MB     --:-- ETA
aws_instance.web (remote-exec): (4/4): base/7/ 92%       |  18 MB     00:00 ETA
aws_instance.web (remote-exec): (4/4): base/7/x86        | 6.1 MB     00:01
aws_instance.web (remote-exec): Resolving Dependencies
aws_instance.web (remote-exec): --> Running transaction check
```

Step 3:- example with File-exec

@Create shell script with following commands.

#cat nginx-install.sh

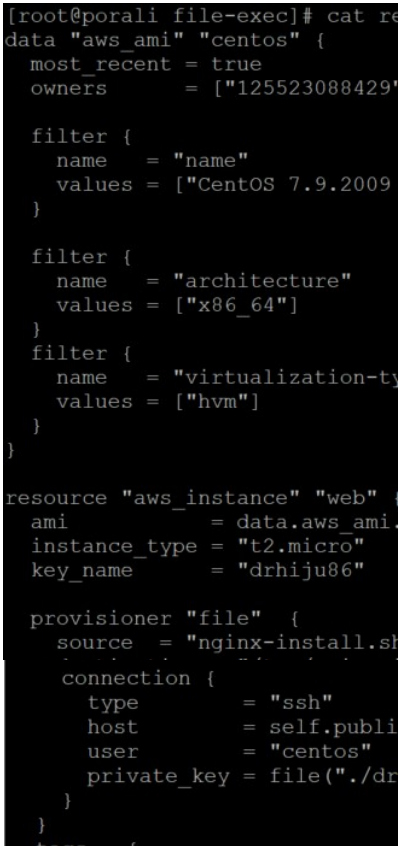
```
[root@porali file-exec]# cat nc
#!/bin/bash
sudo yum update -y
sudo yum install -y epel-releas
```

#cat providers.tf

```
[root@porali remote-exec]# ca
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aw
      version = "~>3.0"
    }
  }
}
```

#cat variable.tf

#cat resource.tf



#terraform init && terraform fmt && terraform validate && terraform plan
#terraform apply

Step 3:- example with Local-exec
@Create a ansible playbook for httpd service

@make sure to add local-exec with ansible entries on main.tf or resource.tf file.

#cat resource.tf

```
[cicd_deepu@karl-marx local-exec]$ cat resource.tf
data "aws_ami" "centos" {
  most_recent = true
  owners      = ["125523088429"]

  filter {
    name   = "name"
    values = ["CentOS 7.9.2009 x86_64"]
  }

  filter {
    name   = "architecture"
    values = ["x86_64"]
  }
  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }
}
resource "aws_instance" "web" {
  ami           = data.aws_ami.centos.id
  instance_type = "t2.micro"
  key_name      = "drhiju86"

  provisioner "remote-exec" {
    inline = [
      "sudo yum update -y",
      "sudo yum install python -y"
    ]


    connection {
      type        = "ssh"
      host        = self.public_ip
      user        = "centos"
      private_key = file("./drhiju86.pem")
    }
    }
    provisioner "local-exec"  {
        command = "ansible-playbook -u centos -i '${self.public_ip},' --private-key drhiju86.pem httpd.yml"
    }
  tags = {
    Name = "robo"
  }
}
[cicd_deepu@karl-marx local-exec]$
```

cat providers.tf

```
[cicd_deepu@karl-marx local-exec]$ cat providers.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~>3.0"
    }
  }
}

#Configure tje AWS Provider
provider "aws" {
  region  = var.region
  profile = "dev"
}
[cicd_deepu@karl-marx local-exec]$
```

#cat variables

#terraform init && terraform fmt && terraform validate && terraform plan

#terraform apply

```
aws_instance.web (remote-exec): Nothing to do
aws_instance.web: Provisioning with 'local-exec'...
aws_instance.web (local-exec): Executing: ["/bin/sh" "-c" "ansible-playbook -u centos -i '3.93.152.124,' --private-key drhiju86.pem http
pd.yml"]

aws_instance.web (local-exec): PLAY [all] ***************************************************************

aws_instance.web (local-exec): TASK [Gathering Facts] ***************************************************
aws_instance.web (local-exec): ok: [3.93.152.124]

aws_instance.web (local-exec): TASK [yum] ***************************************************************
aws_instance.web: Still creating... [5m50s elapsed]
aws_instance.web (local-exec): changed: [3.93.152.124]

aws_instance.web (local-exec): TASK [service] ***********************************************************
aws_instance.web (local-exec): changed: [3.93.152.124]

aws_instance.web (local-exec): PLAY RECAP *********************************************************************
aws_instance.web (local-exec): 3.93.152.124              : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0

aws_instance.web: Creation complete after 5m56s [id=i-00deaef64c274ec45]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

@Verify whether httpd is installed & service is running on instance or not.

@Note:- Local-exec is working perfect, please make sure to have the steps followed properly.