

Terraform External Datasources



@pedrosaxu
no youtube

Alguns itens contidos aqui

- Introdução aos **External Datasources**
- Cenário com **aplicação simples**
- Cenário com **aplicação avançada**

Introdução

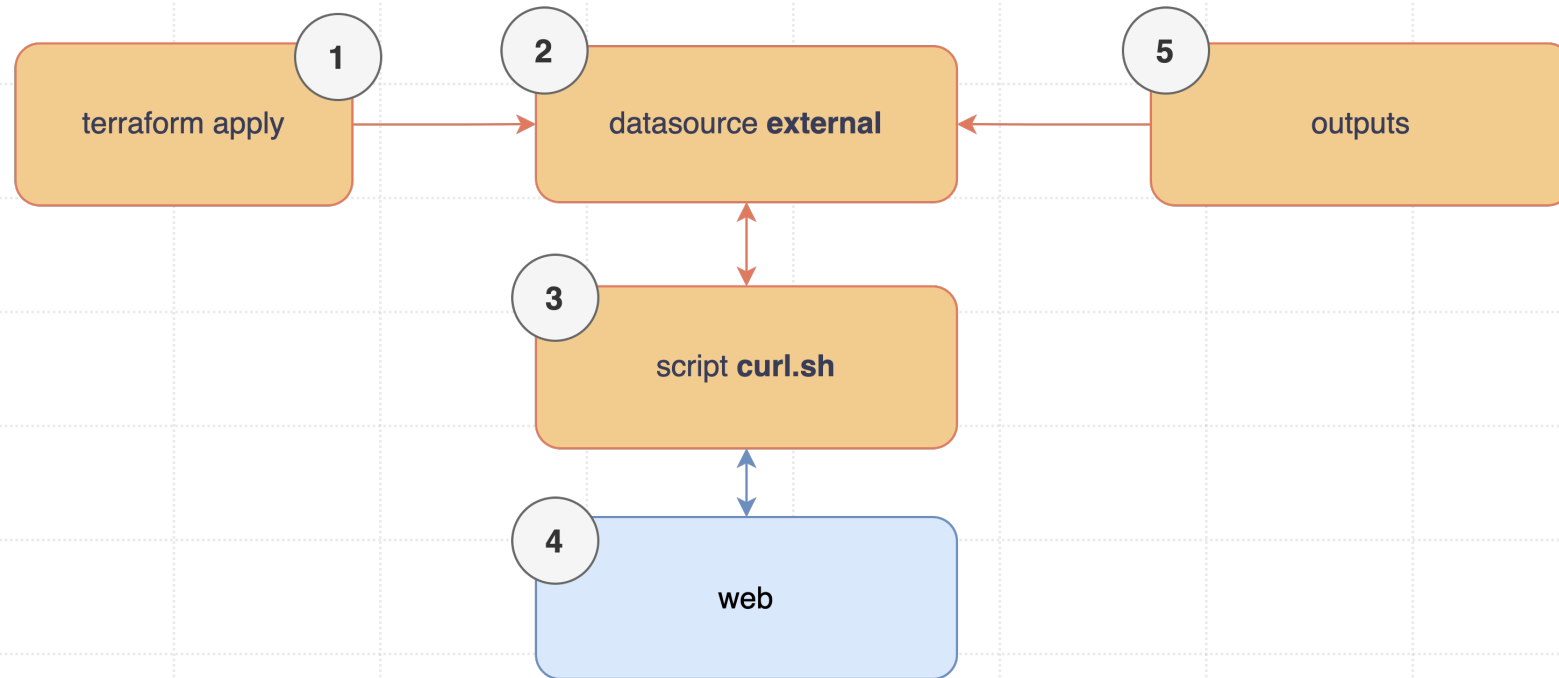
- Uma alternativa **mais indicada aos Provisioners**;
- A maior diferença é que **o resultado do External Datasource pode ser utilizado dentro do Terraform** assim como qualquer outro Datasource;
- Cenário simples e cenário avançado.

⚠ Note que os exemplo com Curl são **exemplos tão simples que não se aplicariam ao dia-a-dia**, no lugar poderíamos **utilizar o próprio provider do curl**, que seria uma opção **menos complexa e mais indicada**, mas a ideia é que você consiga executar scripts mais complexos de forma integrada ao seu Terraform, se necessário.

Cenário simples

- Com curl, **coleta uma frase aleatória** em uma API Pública;
- **Exibe o texto na tela;**
- Utiliza **o texto como um output;**

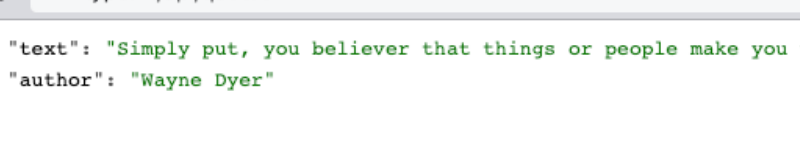
Visão Geral do Processo



```
> terraform state list
data.external.curl_tests
```

```
main.tf U X
Docs > Canal > devops-templates > Youtube > terraform_external_data_endp
1  provider "external" {
2      # Configuration options
3  }
4
5  data "external" "curl_tests" {
6      program = ["bash", "${path.root}/endpoint_curl.sh"]
7  }
8
9  output "curl_tests_result" {
10     value = data.external.curl_tests.result
11 }
```

```
$ endpoint_curl.sh U X
Docs > Canal > devops-templates > Youtube > terraform_external_data_endpoint_curl > 01_cenario_simples > $ en...
1  #!/bin/bash
2
3  set -e
4
5  RESULT=$(curl --silent https://type.fit/api/quotes | jq -r '[] | .text' | shuf -n 1)
6
7  jq -n --arg result "$RESULT" '{"result":$result}'
8
```



The screenshot shows a web browser window with the address bar displaying "type.fit/api/quotes". The page content is a JSON array of three quote objects, rendered in a light green monospace font. The first quote is by Wayne Dyer, the second by Maya Angelou, and the third by Catherine Pulsifer. The browser's developer tools are open, showing the JSON data.

```
[{"text": "Simply put, you believe that things or people make you unhappy", "author": "Wayne Dyer"}, {"text": "Nothing will work unless you do.", "author": "Maya Angelou"}, {"text": "Our ability to achieve happiness and success depends on the strength of our character.", "author": "Catherine Pulsifer"}]
```

```
> curl --silent https://type.fit/api/quotes | jq -r '.[].text' | shuf -n 1
```

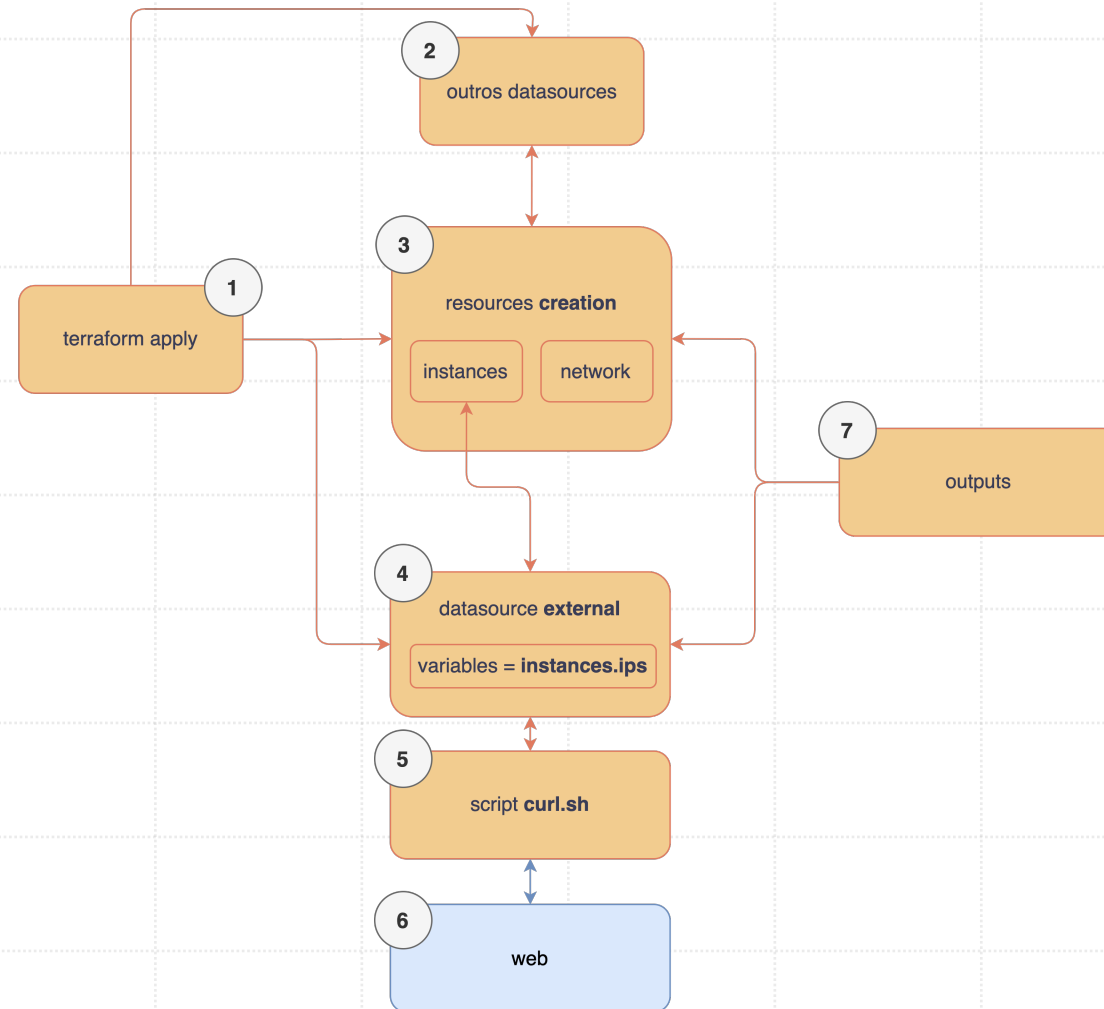
When one door of happiness closes, another opens; but often we look so long at the closed door that we do not see the one which has been opened for us.

Documentação oficial
sobre External Datasource

Cenário avançado

- Criação de **infraestrutura de Rede**;
- Criação de um **pool de servidores Web**;
- **Script bem simples**;
- Faz uma **chamada curl** para o **IP Público dos servidores no Pool de Webservers**;
- Utilização do **resultado destes Curls** como um **output**.

Visão Geral do Processo



Visão Geral dos Recursos

```
data.tf U X
Docs > Canal > devops-templates > Youtube > terraform_external_data_endpoint_curl > data.tf > ...
20 state = available
21 }
22
23
24 # Get script overwriting variables
25 data "external" "curl_tests" {
26   for_each = { for k, v in aws_instance.webserver_pool1 : k => v.public_ip }
27   program = ["bash", "${path.root}/endpoint_curl.sh"]
28   query = {
29     ip = each.value
30   }
31 }
```

```
$ endpoint_curl.sh U X
Docs > Canal > devops-templates > Youtube > terraform_external_data_endpoint_curl > $ endpoint_curl.sh
1 #!/bin/bash
2
3 set -e
4
5 eval "$(jq -r '@sh "ip=\(.ip)"')
6
7 sleep 300
8
9 IP="$ip"
10 RESULT=$(curl $IP)
11
12 jq -n --arg ip "$IP" --arg result "$RESULT" '{"ip":$ip, "result":$result}'
13
```

```
output.tf U X
Docs > Canal > devops-templates > Youtube > terraform_external_data_endpoint_curl > output.tf > ...
1 output "curl_tests_result" {
2   value = values(data.external.curl_tests).*.result
3 }
4
Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
Outputs:
curl_tests_result = [
  tomap({
    "ip" = "54.196.108.114"
    "result" = "<h1><center>Pool 1: Webserver 0</center></h1>"
  }),
]
```

```
> terraform state list
data.aws_availability_zones.azs
data.aws_route_table.main_route_table
data.aws_ssm_parameter.webserver-ami
data.external.curl_tests["0"]
data.external.curl_tests["1"]
data.external.curl_tests["2"]
data.external.curl_tests["3"]
aws_default_route_table.internet_route
aws_instance.webserver_pool1[0]
aws_instance.webserver_pool1[1]
aws_instance.webserver_pool1[2]
aws_instance.webserver_pool1[3]
aws_internet_gateway.igw
aws_key_pair.webserver-key
aws_security_group.sg
aws_subnet.subnet
aws_vpc.vpc
```

```
ec2.tf U X
Docs > Canal > devops-templates > Youtube > terraform_external_data_endpoint_curl > ec2.tf > ...
6 resource "aws_instance" "webserver_pool1" {
7   count = 4
8   ami = data.aws_ssm_parameter.webserver-ami.value
9   instance_type = "t3.micro"
10  key_name = aws_key_pair.webserver-key.key_name
11  associate_public_ip_address = true
12  vpc_security_group_ids = [aws_security_group.sg.id]
13  subnet_id = aws_subnet.subnet.id
14  user_data = <<EOF
15  #!/bin/bash
16  yum -y install httpd && systemctl start httpd
17  echo '<h1><center>Pool 1: Webserver ${count.index}</center></h1>' > index.html
18  mv index.html /var/www/html/
19  EOF
20
21
22
23
24
25
```

54.196.108.114

Pool 1: Webserver 0

```
> curl 54.196.108.114
<h1><center>Pool 1: Webserver 0</center></h1>
```

[Documentação oficial sobre External Datasource](#)

Terraform External Datasources



@pedrosaxu
no youtube