KPLABS Course

HashiCorp Certified: Terraform Associate

Domain 3 - Terraform Provisioners

ISSUED BY

Zeal

REPRESENTATIVE

instructors@kplabs.in

Module 1: Understanding Provisioners in Terraform

1.1 Understanding the Challenge

Till now we have been working only on the creation and destruction of infrastructure scenarios.

Let's take an example:

We created a web-server EC2 instance with Terraform.

Problem: It is only an EC2 instance, it does not have any software installed.

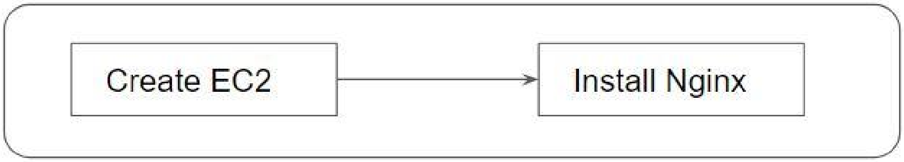
What if we want a complete end to end solution?

1.2 Introducing Terraform Provisioners

Provisioners are used to execute scripts on a local or remote machine as part of resource creation or destruction.

Let's take an example:

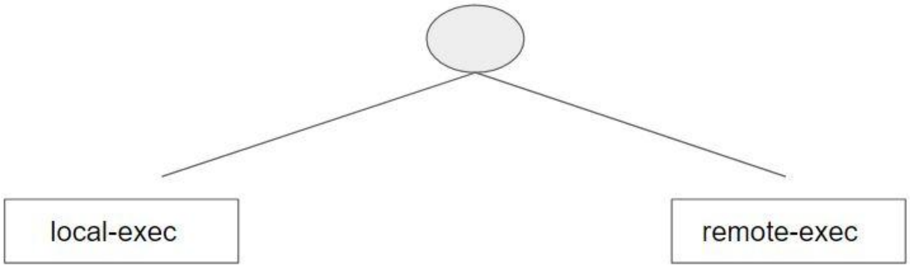
On creation of Web-Server, execute a script which installs Nginx web-server.



Module 2: Types of Provisioners

Terraform has the capability to turn provisioners both at the time of resource creation as well as destruction.

There are two main types of provisioners:



2.1 Local Exec Provisioners

local-exec provisioners allow us to invoke a local executable after the resource is created.

One of the most used approaches of local-exec is to run ansible-playbooks on the created server after the resource is created.

Let’s take an example:

```
provisioner "local-exec" {
   command = "echo ${aws_instance.web.private_ip} >> private_ips.txt"
 }
```

2.2 Remote Exec Provisioners

Remote-exec provisioners allow invoking scripts directly on the remote server.

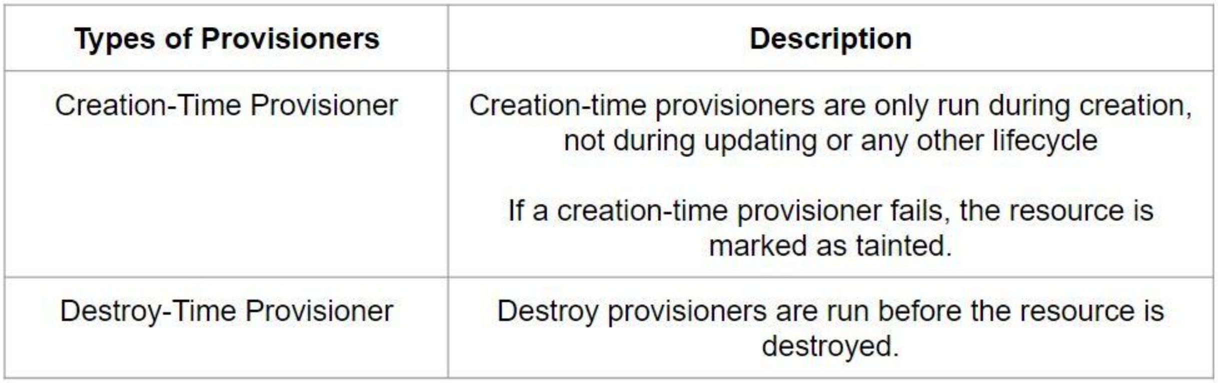
Let’s take an example:

```
resource "aws_instance" "web" {
  # …

  provisioner "remote-exec" {
                          ……………………………...
  }
}
```

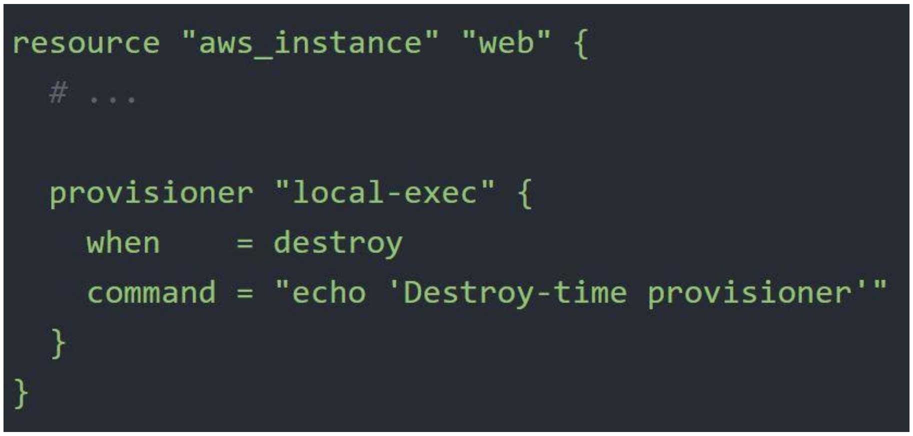
Module 3: Provisioner Types

There are two primary types of provisioners:



Types of Provisioners	Description
Creation-Time Provisioner	Creation-time provisioners are only run during creation, not during updating or any other lifecycle If a creation-time provisioner fails, the resource is marked as tainted.
Destroy-Time Provisioner	Destroy provisioners are run before the resource is destroyed.

If when = destroy is specified, the provisioner will run when the resource it is defined within is destroyed.



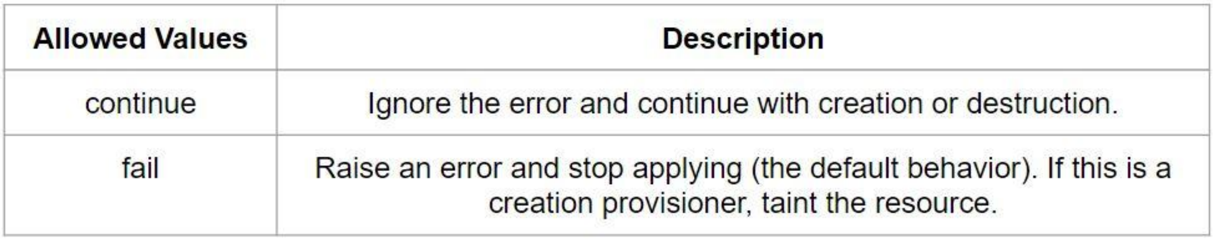
```
resource "aws_instance" "web" {
  # ...

  provisioner "local-exec" {
    when    = destroy
    command = "echo 'Destroy-time provisioner'"
  }
}
```

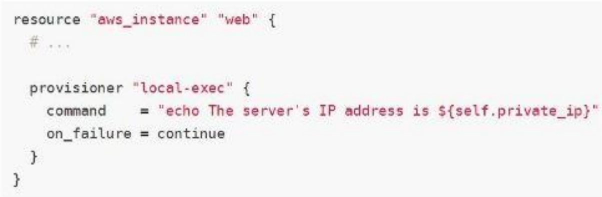
Module 4: Failure Behavior

By default, provisioners that fail will also cause the terraform apply itself to fail.

The on_failure setting can be used to change this. The allowed values are:



Allowed Values	Description
continue	Ignore the error and continue with creation or destruction.
fail	Raise an error and stop applying (the default behavior). If this is a creation provisioner, taint the resource.



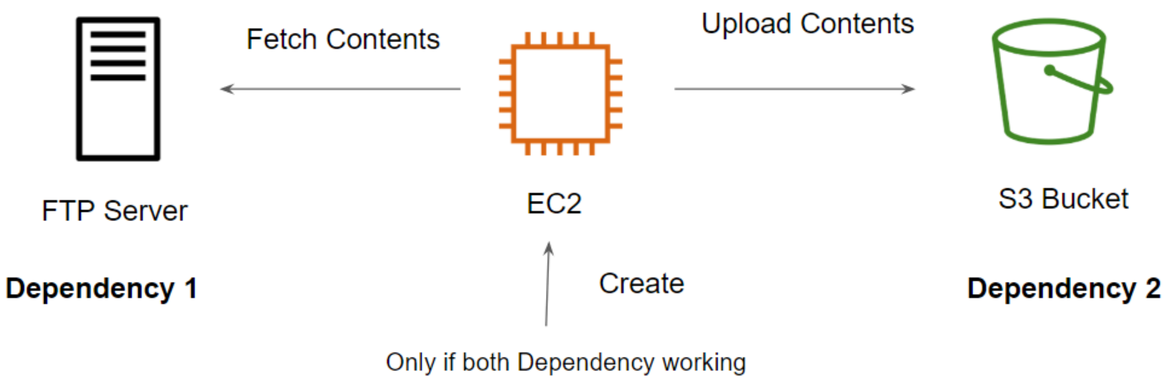
```
resource "aws_instance" "web" {
  # ...

  provisioner "local-exec" {
    command    = "echo The server's IP address is ${self.private_ip}"
    on_failure = continue
  }
}
```

.

Module 5: Null Resource

The null_resource implements the standard resource lifecycle but takes no further action.

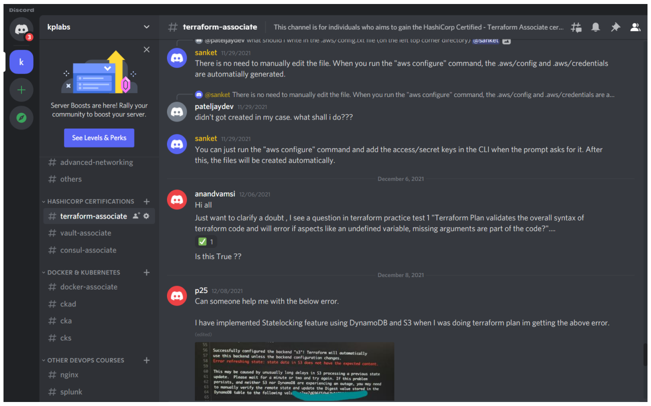


Join Our Discord Community

We invite you to join our Discord community, where you can interact with our support team for any course-based technical queries and connect with other students who are doing the same course.

Joining URL:

http://kplabs.in/chat



@pateljaydev what should i write in the .aws/config.txt file (on the left top corner directory) @sanket

sanket 11/29/2021
There is no need to manually edit the file. When you run the "aws configure" command, the .aws/config and .aws/credentials are automatially generated.

@sanket There is no need to manually edit the file. When you run the "aws configure" command, the .aws/config and .aws/credentials are a...

pateljaydev 11/29/2021
didn't got created in my case. what shall i do???

sanket 11/29/2021
You can just run the "aws configure" command and add the access/secret keys in the CLI when the prompt asks for it. After this, the files will be created automatically.

December 6, 2021

anandvamsi 12/06/2021
Hi all

Just want to clarify a doubt , I see a question in terraform practice test 1 "Terraform Plan validates the overall syntax of terraform code and will error if aspects like an undefined variable, missing arguments are part of the code?"....

✅ 1

Is this True ??

December 8, 2021

p25 12/08/2021
Can someone help me with the below error.

I have implemented Statelocking feature using DynamoDB and S3 when I was doing terraform plan im getting the above error.
(edited)

```
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Error refreshing state: state data in S3 does not have the expected content.

This may be caused by unusually long delays in S3 processing a previous state
update.  Please wait for a minute or two and try again. If this problem
persists, and neither S3 nor DynamoDB are experiencing an outage, you may need
to manually verify the remote state and update the Digest value stored in the
DynamoDB table to the following valu[redacted]
```