

fun and profit and
modules and something
we learned

@mattipaksula

A *wonderful journey* with
a *very special* Terraform
module I've built.

@mattipaksula

everything is on my
github

[Overview](#)[Repositories 254](#)[Stars 86](#)[Followers 51](#)[Following 5](#)

terraform-|

Type: All ▾

Language: All ▾

 New

49 results for repositories matching terraform-

 Clear filter

[terraform-google-kubernetes-engine](#)

 HCL  4 Updated on Feb 6

[terraform-shell-resource](#)

 HCL  19  1 Updated a day ago

[terraform-shell-outputs](#)

 HCL  5 Updated on Mar 9

[terraform-kontena-cli-outputs](#)

 HCL Updated on Feb 11

Overview

Repositories 254

Stars 86

Followers 51

terraform-

49 results for repositories matching **terraform-**

terraform-google-kubernetes-engine

HCL

4

Updated on Feb 6



This repository

Search

Pull requests Issues Marketplace Explore



matti / tf

Watch ▾

0

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

No description, website, or topics provided.

Edit

Add topics

6 commits

1 branch

0 releases

1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

 matti feat (init) Copy workspace related .tf and .tfvars files to dir on init.

Latest commit 2305ba7 on Mar 19

 bin	feat (init) Copy workspace related .tf and .tfvars files to dir on init.	a month ago
 test/root	feat (destroy) Initially working.	a month ago
 .gitignore	feat (recursing) Recurse in sub dirs and apply.	a month ago
 README.md	Update README.md	a month ago

 README.md

tf

```
tf <any terraform command>
tf aa
      - apply -auto-approve
      - destroy -force
      - apply current dir and sub-dirs starting from the current dir.
tf df
```

```
→ meetup-2018-stockholm-hemnet-hashicorp-jättebra git:(master) ✘ tf aa
```

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

Outputs:

```
modulesays = Hejsan, Matti
```

```
→ meetup-2018-stockholm-hemnet-hashicorp-jättebra git:(master) ✘ tf df
```

```
Destroy complete! Resources: 0 destroyed.
```

```
➔ meetup-2018-stockholm-hemnet-hashicorp-jättebra git:(master) ✘ tf plan  
Refreshing Terraform state in-memory prior to plan...  
The refreshed state will be used to calculate this plan, but will not be  
persisted to local or remote state storage.
```

No changes. Infrastructure is up-to-date.

This means that Terraform did not detect any differences between your configuration and real physical resources that exist. As a result, no actions need to be performed.

```
➔ meetup-2018-stockholm-hemnet-hashicorp-jättebra git:(master) ✘ █
```

modules

- re-use, avoid copy-pasta
- alternative for writing *providers* with Go
- Matti's Theorem:
 - All providers and their resources and data sources can be written with modules without Go
 - Proof: we don't have time today

writing modules

- step 1: create a folder mvp_module
- step 2: create main.tf
- step 3: there is no step 3

srsly?

```
1 # variables.tf
2 variable "name" {}
3
4 # main.tf
5 locals {
6   formatted_greeting = "Hejsan, ${var.name}"
7 }
8
9 # outputs.tf
10 output "greeting" {
11   value = "${local.formatted_greeting}"
12 }
```

```
1 module "whateveryouwant" {
2   source = "./mvp_module"
3 }
4
5 output "modulesays" {
6   value = "${module.whateveryouwant.greeting}"
7 }
```

```
→ mvp_module git:(master) ✘ tf aa  
var.name
```

```
Enter a value: Marcus
```

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

Outputs:

```
greeting = Hejsan, Marcus
```

a module inside a module

- mvp_module/
- main.tf
- **test/**
- **main.tf**

```
1 module "goeran_greeter" {
2   source = ".."
3   name = "Göran"
4 }
5
6 module "anna_greeter" {
7   source = ".."
8   name = "Anna"
9 }
10
```

```
1 module "goeran_greeter" {
2   source = ".."
3   name = "Göran"
4 }
5
6 module "anna_greeter" {
7   source = ".."
8   name = "Anna"
9 }
10
11 output "greetings" {
12   value = [
13     "${module.goeran_greeter.greeting}",
14     "${module.anna_greeter.greeting}",
15   ]
16 }
```

```
→ test git:(master) ✘ tf output -json
{
  "greetings": {
    "sensitive": false,
    "type": "list",
    "value": [
      "Hejsan, Göran",
      "Hejsan, Anna"
    ]
  }
}
→ test git:(master) ✘ tf output -json | jq ".greetings.value[0]"
"Hejsan, Göran"
```

publishing

- naming:
 - `terraform-forwhat-whatdoes`
 - `terraform-google-gce-ipxe-stateful-zonal-instance-groups`
- **1 module = 1 repo** because of [registry.terraform.io](#)

question:

```
1 resource "null_resource" "something" {  
2   provisioner "local-exec" {  
3     command = "somecmd -that -outputs -a-token"  
4   }  
5 }
```

question:

```
1 resource "null_resource" "something" {  
2   provisioner "local-exec" {  
3     command = "somecmd -that -outputs -a-token"  
4   }  
5 }
```

how to get the token (output)
?

like so many things in
life Terraform

you don't



This repository

Search

Pull requests Issues Marketplace Explore

[hashicorp / terraform](#)[Watch](#) ▾

816

[Star](#)

11,758

[Fork](#)

3,800

[Code](#)[Issues 1,148](#)[Pull requests 114](#)[Insights](#)

Request: Custom Outputs from null_resource/local-exec

#6830

[New issue](#)

jen20 commented on May 23, 2016

Hi @qivers! It's not possible to do this using null-resource, but the development branch of master) now has [data driven configuration](#), and a data source for finding AMIs is on the road to release!



jen20 closed this on May 23, 2016

```
resource "null_resource" "debian-ami" {
  provisioner "local-exec" {
    command = "python ./ami-finder.py -o debian"
  }
}
```

Notifications

[Unsubscribe](#)

You're receiving notifications because you commented.



This repository

Search

Pull requests Issues Marketplace Explore



hashicorp / terraform

Watch ▾

816

Star

11,758

Fork

3,800

Code

Issues 1,148

Pull requests 114

Insights

Data-driven Terraform Configuration #6598

Merged jen20 merged 22 commits into master from f-data-sources on May 16, 2016

Conversation 3

Commits 22

Files changed 51

+2,027 -183



apparentlymart commented on May 10, 2016 • edited by jbardin

Contributor



This is where I'm working on the implementation of the proposal from [#4169](#).

This has been re-opened a bunch of times by this point as it's moved from my fork to the main repo, from master to dev-0.7, and now back from dev-0.7 to master again. 😐

Since this change spans multiple Terraform layers, the sections that follow summarize the changes in each layer, in the hope of making this changeset easier to review. The PR is broken into a sequence of commits which, as far as possible, change only one layer at a time so that each change can be understood in isolation.

Configuration (config package)

Reviewers

No reviews

Assignees

No one assigned

Labels

core

enhancement

Projects

None yet



jen20 closed this on May 23, 2016



alexforever86 commented on Aug 10, 2016



Hi, @jen20 I'm not able to see how v0.7.0 can fix this issue. This is required essentially for me as I'm using a script to get the private_ip of resources created by auto_scaling group. I need to pass this information to another TF module. So, I need to output this. Is there any way, I can achieve this?



jen20 closed this on May 23, 2016



alexforever86 commented on Aug 10, 2016



Hi, @jen20 I'm not able to see how v0.7.0 can fix this issue. This is required essentially for me as I'm using a script to get the private_ip of resources created by auto_scaling group. I need to pass this information to another TF module. So, I need to output this. Is there any way, I can achieve this?

wat do?

go provider?

[toddnni / terraform-provider-shell](#)[Watch](#) 3[Star](#) 6[Fork](#) 1[Code](#)[Issues 0](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Insights](#)

No description, website, or topics provided.

[17 commits](#)[1 branch](#)[0 releases](#)[1 contributor](#)[MIT](#)

Known Problems

- [REDACTED]
- The provider won't print output of the commands.
- [REDACTED]
- [REDACTED]

There is also [exec](#) provider, but it only implements `Create` CRUD operation.

remember my
hypothesis?

module time!

aka

a wonderful journey
with



This repository

Search



[matti / terraform-shell-resource](#)

```
1 resource "null_resource" "something" {  
2   provisioner "local-exec" {  
3     command = "somecmd -that -outputs -a-token > output.txt"  
4   }  
5 }  
6
```

```
1  resource "null_resource" "something" {  
2    provisioner "local-exec" {  
3      command = "somecmd -that -outputs -a-token > output.txt"  
4    }  
5  }  
6  
7  data "local_file" "output" {  
8    depends_on = ["null_resource.something"]  
9    filename  = "output.txt"  
10 }  
11
```

```
1  resource "null_resource" "something" {
2    provisioner "local-exec" {
3      command = "somecmd -that -outputs -a-token > output.txt"
4    }
5  }
6
7  data "local_file" "output" {
8    depends_on = ["null_resource.something"]
9    filename  = "output.txt"
10 }
11
```

what if output.txt is
removed?



Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

Support for the second apply (when output files generated and not there anymore) #1

Closed

JPLachance opened this issue on Mar 15 · 7 comments



JPLachance commented on Mar 15

+

First of all, thanks for sharing that module! It's helping me a lot to understand how I could run a shell command and get its output. 😊

I am trying to generate a key using the `openssl rand -hex 16`. I found your module and I tried it:

```
module "openssl_rand_key" {
  source  = "matti/resource/shell"
  version = "0.3.2"

  command = "openssl rand -hex 16"
}
```

Assigned to

No one

Labels

None yet

Projects

None yet

Milestones

No milestones

hmm..

[Configuration](#)[Commands \(CLI\)](#)[Import](#)[State](#)[Providers](#)[Provisioners](#)[chef](#)[connection](#)[file](#)[habitat](#)[local-exec](#)[null_resource](#)[remote-exec](#)[salt-masterless](#)[Modules](#)

null_resource

The `null_resource` is a resource that allows you to configure provisioners that are not directly associated with a single existing resource.

A `null_resource` behaves exactly like any other resource, so you configure [provisioners](#), [connection details](#), and other meta-parameters in the same way you would on any other resource.

This allows fine-grained control over when provisioners run in the dependency graph.

Example usage

```
resource "aws_instance" "cluster" {
  count = 3

  # ...
}

resource "null_resource" "cluster" {
  # Changes to any instance of the cluster requires re-provisioning
  triggers {
```

Argument Reference

In addition to all the resource configuration available, `null_resource` supports the following specific configuration options:

- `triggers` - A mapping of values which should trigger a rerun of this set of provisioners. Values are meant to be interpolated references to variables or attributes of other resources.

Argument Reference

In addition to all the resource configuration available, `null_resource` supports the following specific configuration options:

- `triggers` - A mapping of values which should trigger a rerun of this set of provisioners. Values are meant to be interpolated references to variables or attributes of other resources.

I get a "a *mapping of values*"

...and also

Argument Reference

In addition to all the resource configuration available, `null_resource` supports the following specific configuration options:

- `triggers` - A mapping of values which should trigger a rerun of this set of provisioners. Values are meant to be interpolated references to variables or attributes of other resources.

Values are meant to be interpolated references to variables or attributes of other resources.

Argument Reference

In addition to all the resource configuration available, `null_resource` supports the following specific configuration options:

- `triggers` - A mapping of values which should trigger a rerun of this set of provisioners. Values are meant to be interpolated references to variables or attributes of other resources.

Values are meant to be interpolated references to variables or attributes of other resources.

wat



matti commented 27 days ago

Owner + 😊 ...

I might have solved this with a horrible hack - getting too drunk to verify it fully, let's see tomorrow if I need to drink more or not.

😊 1



```
1 resource "null_resource" "something" {
2   provisioner "local-exec" {
3     command = "somecmd -that -outputs -a-token > output.txt"
4   }
5 }
6
7 data "local_file" "output" {
8   depends_on = ["null_resource.something"]
9   filename   = "output.txt"
10 }
11
12 resource "null_resource" "stash" {
13   triggers = {
14     output = "${data.local_file.output.content}"
15   }
16 }
```

```
1 resource "null_resource" "something" {
2   provisioner "local-exec" {
3     command = "somecmd -that -outputs -a-token > output.txt"
4   }
5 }
6
7 data "local_file" "output" {
8   depends_on = ["null_resource.something"]
9   filename   = "output.txt"
10 }
11
12 resource "null_resource" "stash" {
13   triggers = {
14     output = "${data.local_file.output.content}"
15   }
16 }
17
18 output "stashed_output" {
19   value = "${null_resource.stash.triggers["output"]}"
20 }
```

```
→ shell_resource git:(master) ✘ tf aa
null_resource.something: Creating...
null_resource.something: Provisioning with 'local-exec'...
null_resource.something (local-exec): Executing: ["./bin/sh" "-c" "somecmd -that
en > output.txt"]
null_resource.something: Creation complete after 0s (ID: 5308348052053635468)
data.local_file.output: Refreshing state...
null_resource.stash: Creating...
  triggers.%:      "" => "1"
  triggers.output: "" => "abbacdabbacdacdc\n"
null_resource.stash: Creation complete after 0s (ID: 9143888003722038711)
```

```
→ shell_resource git:(master) ✘ tf aa
null_resource.something: Creating...
null_resource.something: Provisioning with 'local-exec'...
null_resource.something (local-exec): Executing: ["./bin/sh" "-c" "somecmd -that
en > output.txt"]
null_resource.something: Creation complete after 0s (ID: 5308348052053635468)
data.local_file.output: Refreshing state...
null_resource.stash: Creating...
  triggers.%:      "" => "1"
  triggers.output: "" => "abbacdabbacdacdc\n"
null_resource.stash: Creation complete after 0s (ID: 9143888003722038711)
```

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

```
stashed_output = abbacdabbacdacdc
→ shell_resource git:(master) ✘ █
```

but

```
→ shell_resource git:(master) ✘ rm output.txt  
→ shell_resource git:(master) ✘ tf aa
```

```
→ shell_resource git:(master) ✘ rm output.txt
→ shell_resource git:(master) ✘ tf aa
null_resource.something: Refreshing state... (ID: 4003726849089400476)
null_resource.stash: Refreshing state... (ID: 1730357087954157984)
null_resource.stash: Destroying... (ID: 1730357087954157984)
data.local_file.output: Refreshing state...
null_resource.stash: Destruction complete after 0s
```

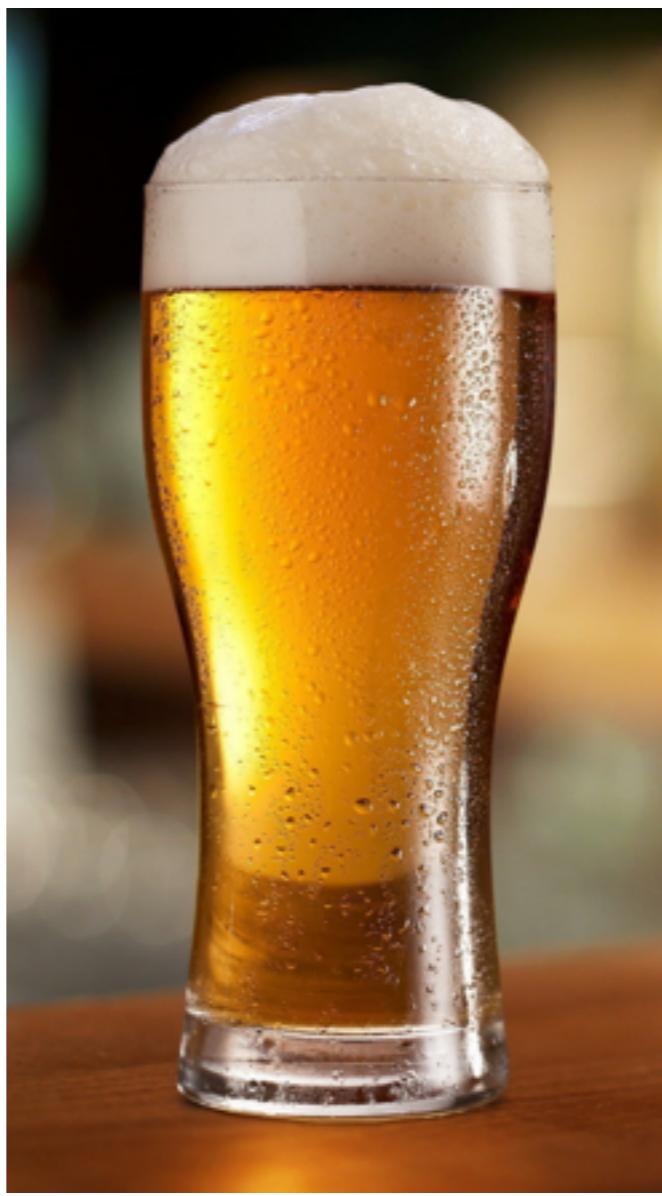
Error: Error applying plan:

1 error(s) occurred:

```
* data.local_file.output: data.local_file.output: open output.txt: no such file or directory
```

Terraform does not automatically rollback in the face of errors.
Instead, your Terraform state file has been partially updated with
any resources that successfully completed. Please address the error
above and apply again to incrementally change your infrastructure.

wat do?



- [All Providers](#)
- [External Provider](#)
- [Data Source](#)

External Data Source

The external data source allows an external program implementing a specific protocol (defined below) to act as a data source, exposing arbitrary data for use elsewhere in the Terraform configuration.

Warning This mechanism is provided as an "escape hatch" for exceptional situations where a first-class Terraform provider is not more appropriate.

Plan for attack

- Write an external program that
 - if the file exists
 - return the contents
 - else
 - return ""

```
1 require 'json'  
2  
3 params = JSON.parse STDIN.read  
4 if File.exist? params["filename"]  
5   contents = File.read params["filename"]  
6 end  
7  
8 obj = {  
9   contents: contents  
10 }  
11  
12 puts obj.to_json  
13
```

```
1   resource "null_resource" "something" {
2     provisioner "local-exec" {
3       command = "somecmd -that -outputs -a-token > output.txt"
4     }
5   }
6
7   data "external" "reader" {
8     program     = ["ruby", "reader.rb"]
9     depends_on = ["null_resource.something"]
10
11    query = {
12      filename = "output.txt"
13    }
14  }
15
```

```
1   resource "null_resource" "something" {
2     provisioner "local-exec" {
3       command = "somecmd -that -outputs -a-token > output.txt"
4     }
5   }
6
7   data "external" "reader" {
8     program     = ["ruby", "reader.rb"]
9     depends_on = ["null_resource.something"]
10
11    query = {
12      filename = "output.txt"
13    }
14  }
15
16  resource "null_resource" "stash" {
17    triggers = {
18      output = "${data.external.reader.result["contents"]}"
19    }
20  }
21
22  output "stashed_output" {
23    value = "${null_resource.stash.triggers["output"]}"
24  }
```

```
→ shell_resource git:(master) ✘ rm output.txt  
→ shell_resource git:(master) ✘ tf aa
```

```
→ shell_resource git:(master) ✘ rm output.txt
→ shell_resource git:(master) ✘ tf aa
null_resource.something: Refreshing state... (ID: 2561217176410429585)
null_resource.stash: Refreshing state... (ID: 2057884690421461927)
null_resource.stash: Destroying... (ID: 2057884690421461927)
null_resource.stash: Destruction complete after 0s
data.external.reader: Refreshing state...
null_resource.stash: Creating...
triggers.%:      "" => "1"
triggers.output: "" => ""
null_resource.stash: Creation complete after 0s (ID: 8183876098735153056)
```

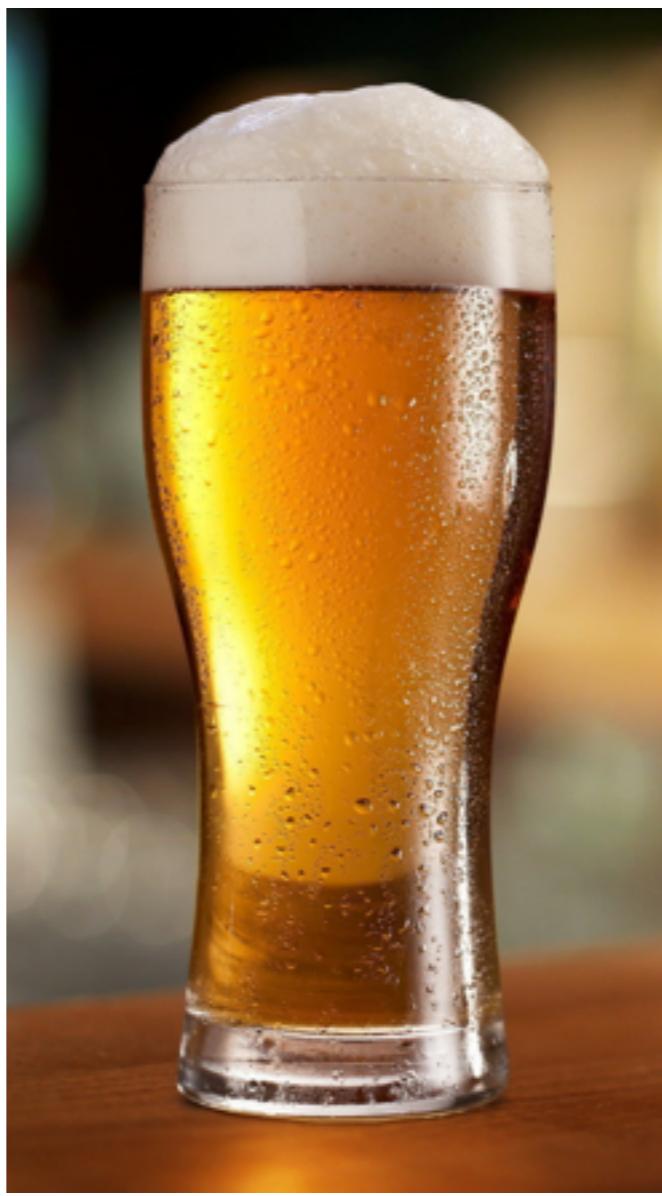
```
→ shell_resource git:(master) ✘ rm output.txt
→ shell_resource git:(master) ✘ tf aa
null_resource.something: Refreshing state... (ID: 2561217176410429585)
null_resource.stash: Refreshing state... (ID: 2057884690421461927)
null_resource.stash: Destroying... (ID: 2057884690421461927)
null_resource.stash: Destruction complete after 0s
data.external.reader: Refreshing state...
null_resource.stash: Creating...
triggers.%:      "" => "1"
triggers.output: "" => ""
null_resource.stash: Creation complete after 0s (ID: 8183876098735153056)
```

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

Outputs:

```
stashed_output =
```

wat do?



meta-parameters

Meta-parameters

There are **meta-parameters** available to all resources:

- [count](#) (int) - The number of identical resources to create. This doesn't apply to all resources. For details on using variables in conjunction with count, see [Using Variables with count](#) below.

Modules don't currently support the count parameter.

- [depends_on](#) (list of strings) - Explicit dependencies that this resource has. These dependencies will be created before this resource. For syntax and other details, see the section below on [explicit dependencies](#).
- [provider](#) (string) - The name of a specific provider to use for this resource. The name is in the format of TYPE.ALIAS, for example, aws.west. Where west is set using the alias attribute in a provider. See [multiple provider instances](#).
- [lifecycle](#) (configuration block) - Customizes the lifecycle behavior of the resource. The specific options are documented below.

The `lifecycle` block allows the following keys to be set:

- [create_before_destroy](#) (bool) - This flag is used to ensure the replacement of a resource is created before the original instance is destroyed. As an example, this can be used to create a new DNS record before removing an old record.
- [prevent_destroy](#) (bool) - This flag provides extra protection against the destruction of a given resource. When this is set to true, any plan that includes a destroy of this resource will return an error message.
- [ignore_changes](#) (list of strings) - Customizes how diffs are evaluated for resources, allowing individual attributes to be ignored through changes. As an example, this can be used to ignore dynamic changes to the resource from external resources. Other meta-parameters cannot be ignored.

Meta-parameters

There are **meta-parameters** available to all resources:

- `count` (int) - The number of identical resources to create. This doesn't apply to all resources. For details on using variables in conjunction with count, see [Using Variables with count](#) below.

Modules don't currently support the `count` parameter.

- `depends_on` (list of strings) - Explicit dependencies that this resource has. These dependencies will be created

- **`ignore_changes`** (list of strings) - Customizes how attributes to be ignored through changes. As an resource from external resources. Other meta-p

- `create_before_destroy` (bool) - This flag is used to ensure the replacement of a resource is created before the original instance is destroyed. As an example, this can be used to create a new DNS record before removing an old record.

- `prevent_destroy` (bool) - This flag provides extra protection against the destruction of a given resource. When this is set to true, any plan that includes a destroy of this resource will return an error message.

- `ignore_changes` (list of strings) - Customizes how diffs are evaluated for resources, allowing individual attributes to be ignored through changes. As an example, this can be used to ignore dynamic changes to the resource from external resources. Other meta-parameters cannot be ignored.

```
resource "null_resource" "stash" {
  triggers = {
    output = "${data.external.reader.result["contents"]}"
  }

  lifecycle {
    ignore_changes = [
      "triggers",
    ]
  }
}
```

→ shell_resource git:(master) ✘ tf aa

```
→ shell_resource git:(master) ✘ tf aa  
null_resource.something: Refreshing state... (ID: 6058126027141669987)  
null_resource.stash: Refreshing state... (ID: 8535799801043622399)  
data.external.reader: Refreshing state... this returns 
```

```
→ shell_resource git:(master) ✘ tf aa  
null_resource.something: Refreshing state... (ID: 6058126027141669987)  
null_resource.stash: Refreshing state... (ID: 8535799801043622399)  
data.external.reader: Refreshing state... this returns !!!  
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

Outputs:

```
stashed_output = abbacdabbacdacdc
```

but it is ignored

how to remove ruby / external data source?



This repository

Search

Pull requests Issues Marketplace Explore



matti / terraform-shell-resource

Unwatch

3

Star

19

Fork

1

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

bourne shell #2

[Edit](#)[New issue](#)[Closed](#)

krzysztofantczak opened this issue 2 days ago · 5 comments



krzysztofantczak commented 2 days ago • edited



Hey,

Its really naive implementation which uses bourne shell, but it works for simple use cases - as a replacement for read.rb.

```
#!/bin/sh

files=`cat $1:-/dev/stdin` | grep -Eo '"[^"]+"' | grep -v '"[a-z]"' | sed s/\\"//g`
exitcode=$((cat `echo "$files" | head -n1` | sed -e ':a' -e 'N' -e '$!ba' -e 's/\n/\\n\\n/g')
stderr=$(cat `echo "$files" | head -n2` | tail -n1 | sed -e ':a' -e 'N' -e '$!ba' -e 's/\n/
stdout=$(cat `echo "$files" | tail -n1` | sed -e ':a' -e 'N' -e '$!ba' -e 's/\n/\\n\\n/g')

echo "{\"stdout\":\"$stdout\", \"stderr\":\"$stderr\", \"exitstatus\": \"$exitcode\"}"
```

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Plan for attack II

- create an empty file (always)
- overwrite that file (on the first apply)
- read that file with vanilla terraform

```
1 resource "local_file" "output_dummy" {  
2   content = ""  
3   filename = "output.txt"  
4 }  
5
```

```
1 resource "local_file" "output_dummy" {
2   content = ""
3   filename = "output.txt"
4 }
5
6 resource "null_resource" "something" {
7   depends_on = ["local_file.output_dummy"]
8
9   provisioner "local-exec" {
10    command = "somecmd -that -outputs -a-token > output.txt"
11  }
12 }
```

```
6   resource "null_resource" "something" {
7     depends_on = ["local_file.output_dummy"]
8
9     provisioner "local-exec" {
10       command = "somecmd -that -outputs -a-token > output.txt"
11     }
12   }
13 }
```

```
6   resource "null_resource" "something" {
7     depends_on = ["local_file.output_dummy"]
8
9     provisioner "local-exec" {
10       command = "somecmd -that -outputs -a-token > output.txt"
11     }
12   }
13
14   data "local_file" "output" {
15     depends_on = [
16       "null_resource.something",
17       "local_file.output_dummy",
18     ]
19     =
20     filename = "output.txt"
21   }
```

```
14  data "local_file" "output" {
15    depends_on = [
16      "null_resource.something",
17      "local_file.output_dummy",
18    ]
19    =
20    filename = "output.txt"
21  }
22  resource "null_resource" "stash" {
23    triggers = {
24      output = "${data.local_file.output.content}"
25    }
  }
```

```
→ shell_resource git:(master) ✘ rm output.txt  
→ shell_resource git:(master) ✘ tf aa
```

```
→ shell_resource git:(master) ✘ rm output.txt
→ shell_resource git:(master) ✘ tf aa
local_file.output_dummy: Refreshing state... (ID: da39a3ee5e6b4b0d3255bfe)
null_resource.something: Refreshing state... (ID: 8481132068867294232)
null_resource.stash: Refreshing state... (ID: 2071642807951430475)
local_file.output_dummy: Creating...
  filename: "" => "output.txt"
local_file.output_dummy: Creation complete after 0s (ID: da39a3ee5e6b4b0d
09)
data.local_file.output: Refreshing state...
```

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

Outputs:

```
stashed_output = abbacdabbacdacdc_
```

terraform-shell-resource

- stdout, stderr, exitcode
- great for implementing almost anything without go

```
module "files" {  
  source = "matti/resource/shell"  
  command = "ls -l"  
}  
  
output "my_files" {  
  value = "${module.files.stdout}"  
}
```

some additional modz

- execers
 - <https://github.com/matti/terraform-shell-resource>
 - <https://github.com/matti/terraform-shell-outputs>
- utils
 - <https://github.com/matti/terraform-http-download>
 - <https://github.com/matti/terraform-ssh-remove-known-hosts>
 - <https://github.com/matti/terraform-dir-glob-outputs>

cont'd

- blocking wait:
 - <https://github.com/matti/terraform-http-until>
 - <https://github.com/matti/terraform-tcp-until>
 - <https://github.com/matti/terraform-shell-until>
- transformers
 - <https://github.com/matti/terraform-map-yaml>
 - <https://github.com/matti/terraform-json-map>
 - <https://github.com/matti/terraform-json-yaml>

tack, tack

github/matti/
meetup-2018-stockholm-hemnet-hashicorp-jättebra

@mattipaksula