From zero to hero

Infrastructure as Code

Daniel Hasselwander

ti&m

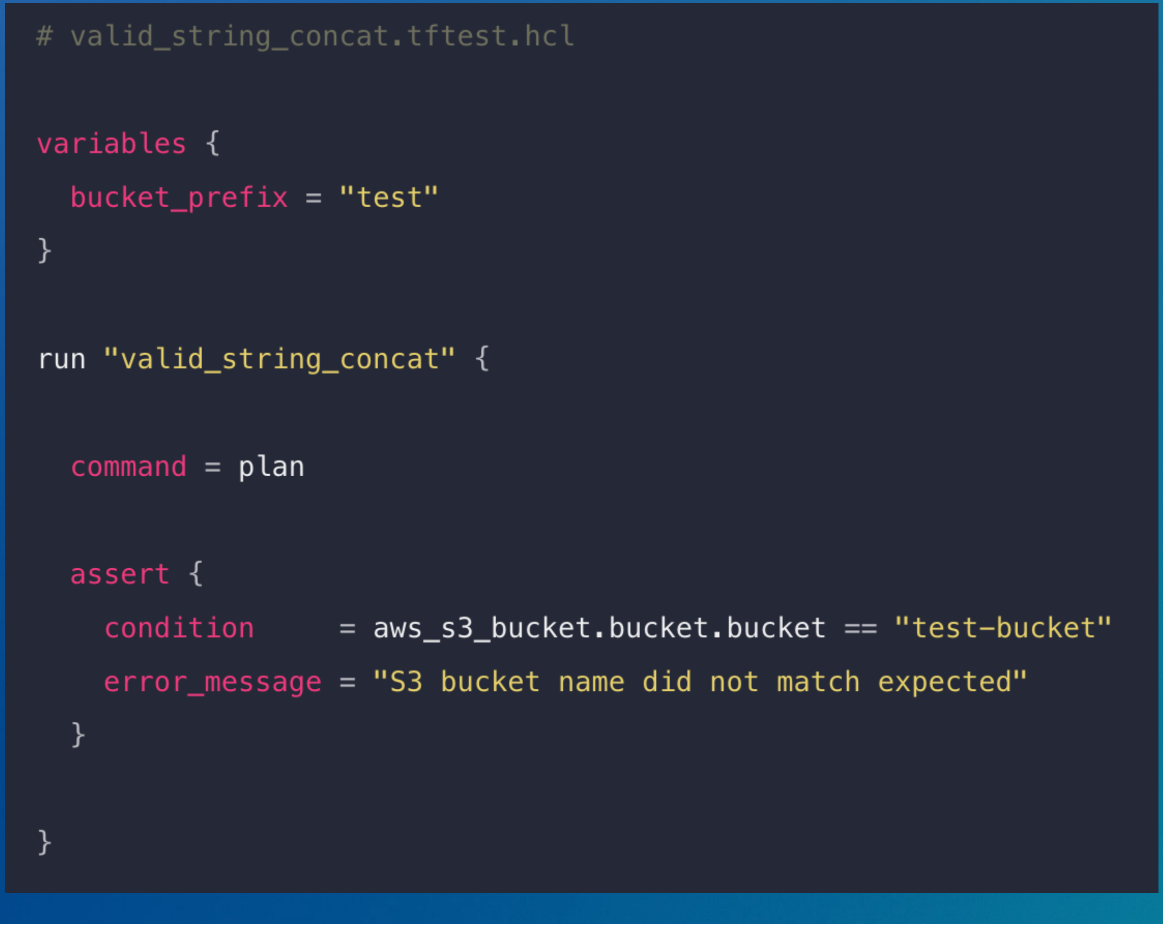
Goals for this session

1. Understand how you can test your terraform code
2. Understand how you can visualize your code
3. Understand how you can check your code for typical problems

How you can test your code

- Like the same than in programming a software tests can bring more stabilty
- Terraform itself build with the newest versions a chance to test your code
- Terraform test
- Store in a .hcl in your module

How a test looks like



```
# valid_string_concat.tftest.hcl

variables {
  bucket_prefix = "test"
}

run "valid_string_concat" {

  command = plan

  assert {
    condition     = aws_s3_bucket.bucket.bucket == "test-bucket"
    error_message = "S3 bucket name did not match expected"
  }

}
```

Terraform

Visualize your code

- Use the dependency graph
- Use **Terraform Visual**

ti&m

Terraform

Visualize your code - Dot

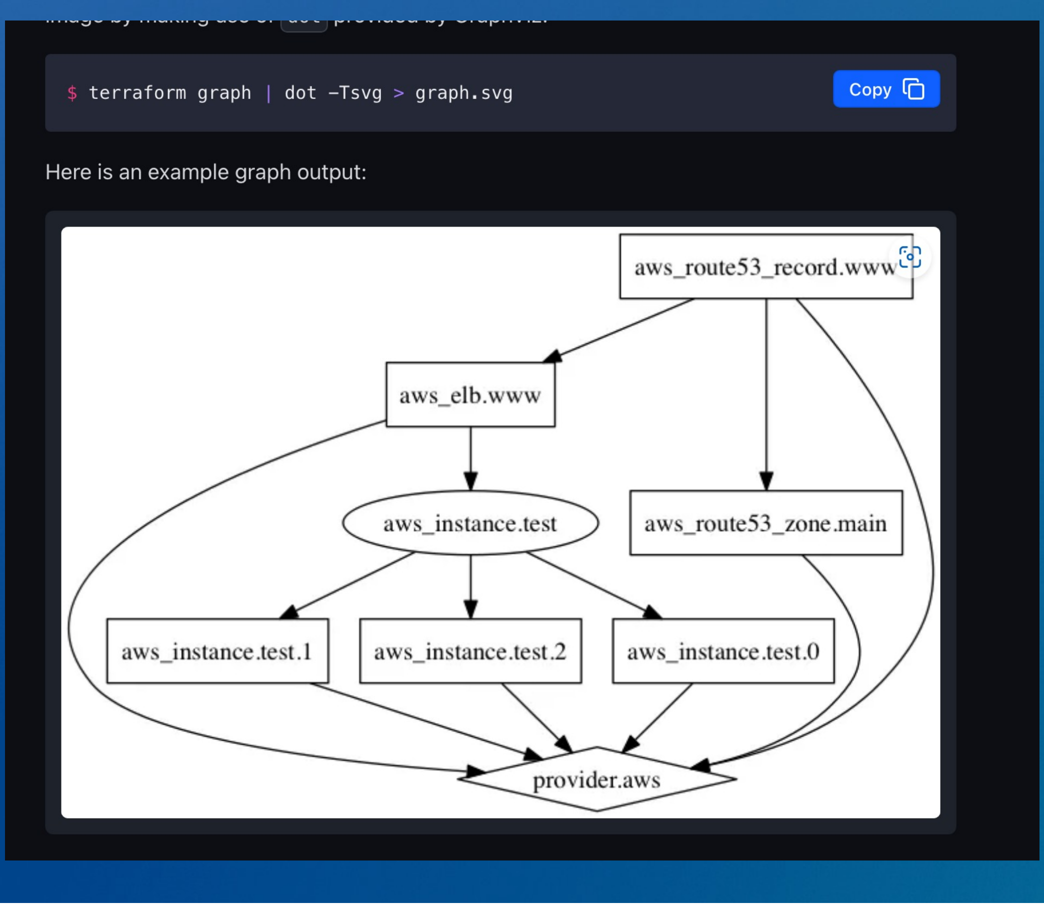


image by making use of `dot` provided by GraphViz:

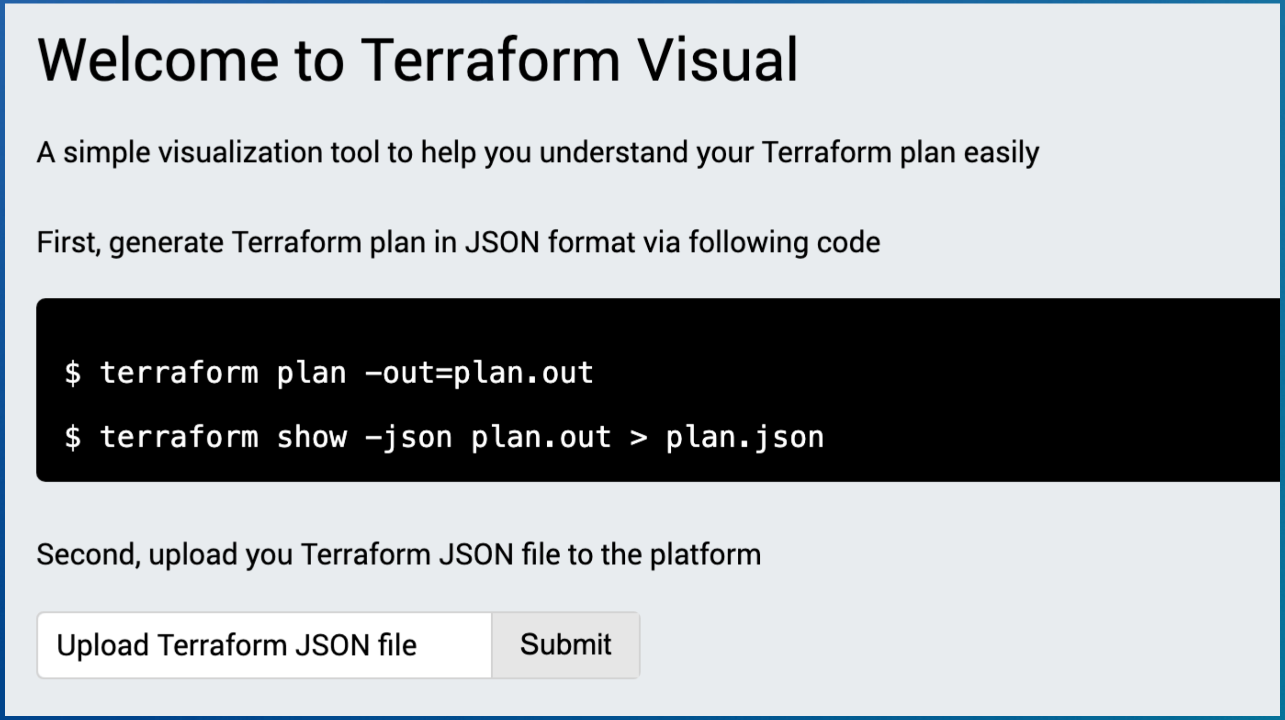
```
$ terraform graph | dot -Tsvg > graph.svg
```

Here is an example graph output:

- Must install GraphViz

Terraform

Visualize your code Visual



Welcome to Terraform Visual

A simple visualization tool to help you understand your Terraform plan easily

First, generate Terraform plan in JSON format via following code

```
$ terraform plan -out=plan.out

$ terraform show -json plan.out > plan.json
```

Second, upload you Terraform JSON file to the platform

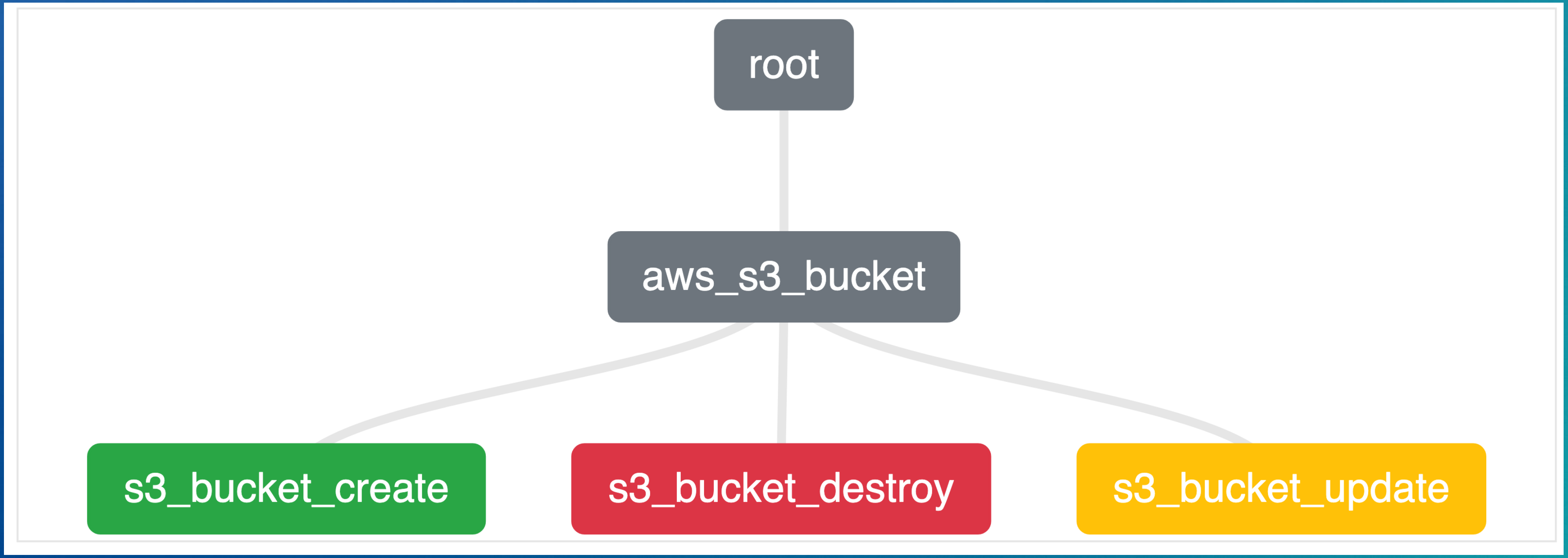
- Use GithubPage
- [Terraform Visual (hieven.github.io)](hieven.github.io)

- Local Install with npm

- Docker Container

Terraform

Visualize your code Visual



root

aws_s3_bucket

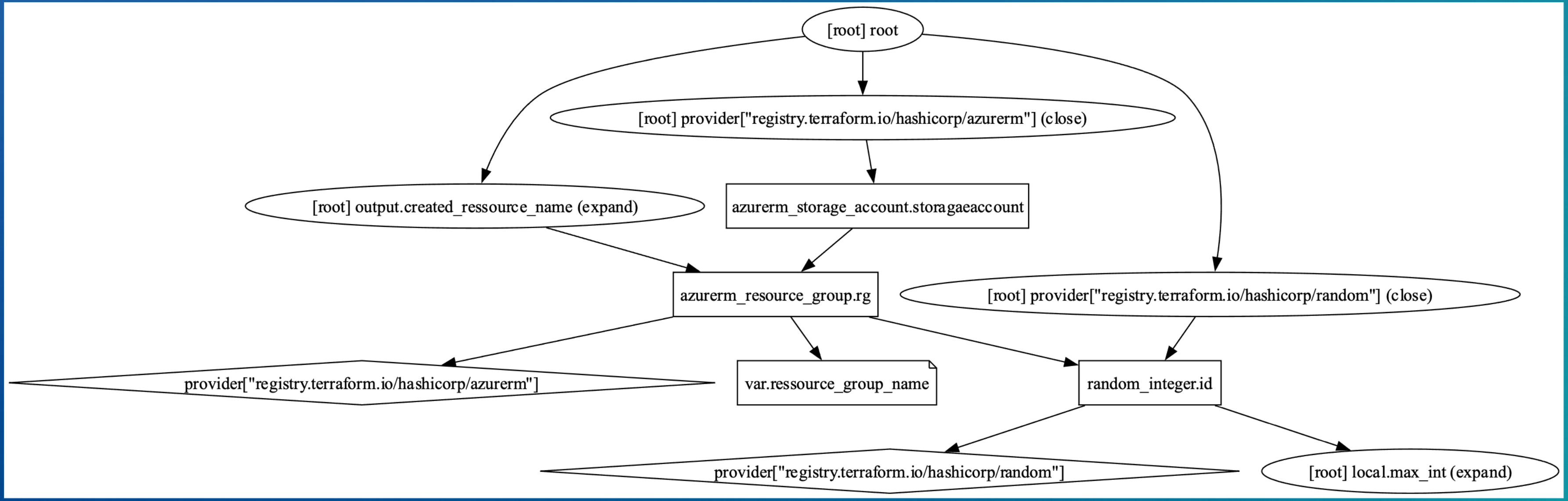
s3_bucket_create

s3_bucket_destroy

s3_bucket_update

Terraform

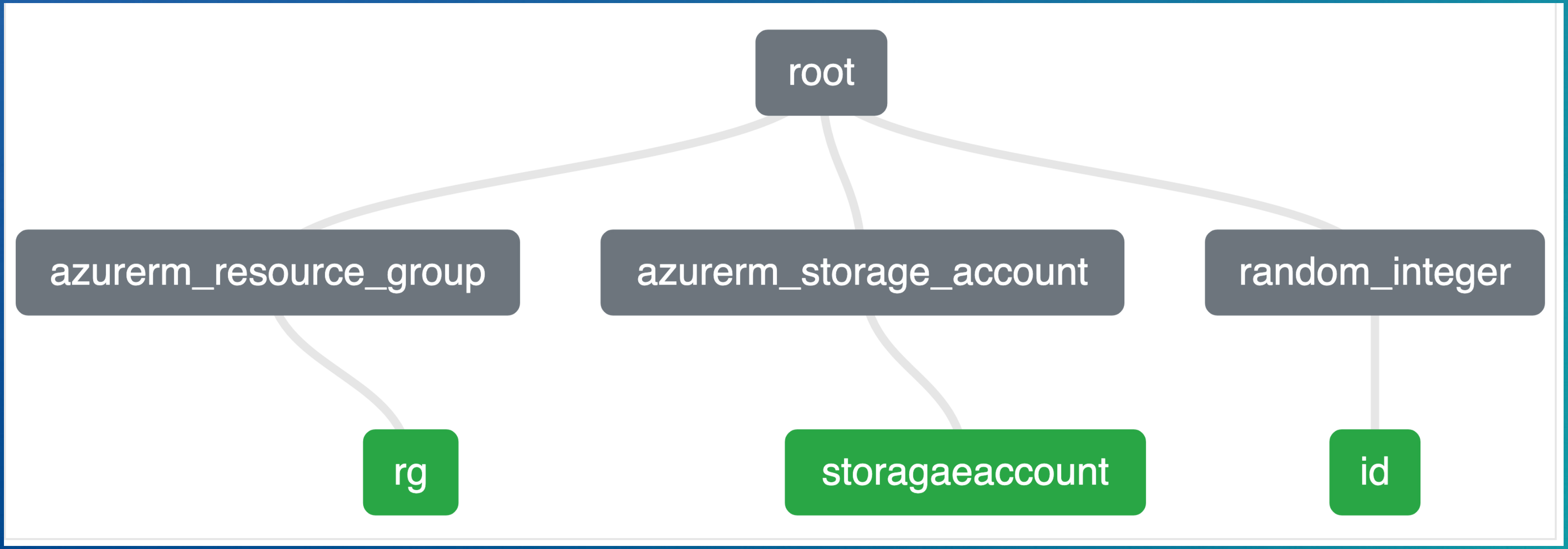
Results


[root] root
[root] provider["registry.terraform.io/hashicorp/azurerm"] (close)
[root] output.created_ressource_name (expand)
azurerm_storage_account.storagaeaccount
azurerm_resource_group.rg
[root] provider["registry.terraform.io/hashicorp/random"] (close)
provider["registry.terraform.io/hashicorp/azurerm"]
var.ressource_group_name
random_integer.id
provider["registry.terraform.io/hashicorp/random"]
[root] local.max_int (expand)

ti&m

Terraform

Results



root

azurerm_resource_group

azurerm_storage_account

random_integer

rg

storagaeaccount

id

Terraform

TFlint

- Linter for Terraform for typical wrong configuration
- terraform-linters/tflint: A Pluggable Terraform Linter (github.com)
- Can be installed or use with docker
- Use .tflint.hcl

Terraform

TFlint

Assets

ti&m