DRY coding with Terraform, CSVs, ForEach

Jesse Loudon
Principal Consultant

coder_au
jloudon.com
jesseloudon

What problems are we solving?

- Large-scale inputs and data
- Frequent changes (adds/removes/updates)
- Maximum code reusability
- Minimum coding effort
- Time sensitive delivery

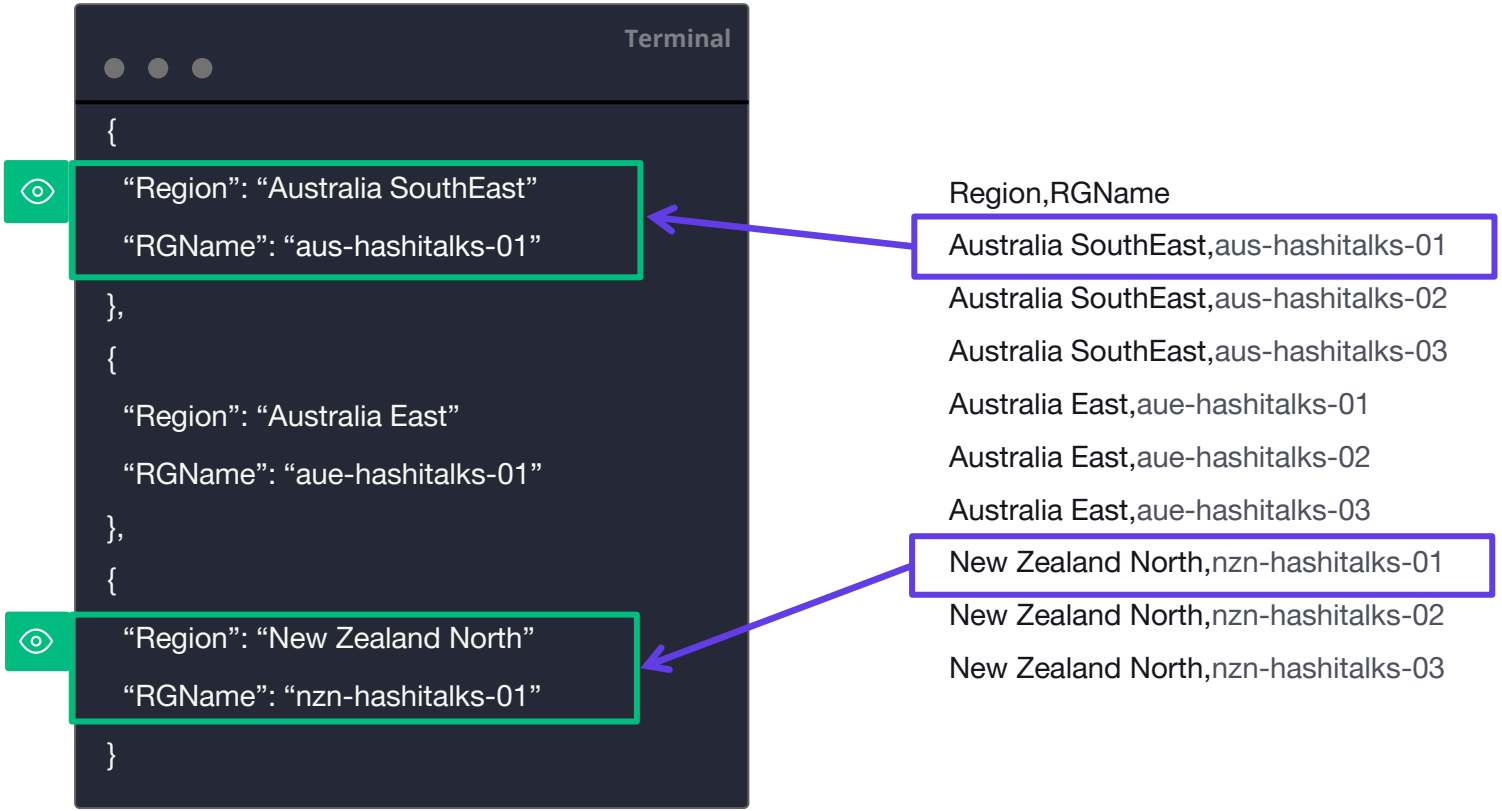
Let’s talk data

CSV dataset

- **Header row**
- **Key/value pairs**
- **Resource per row**

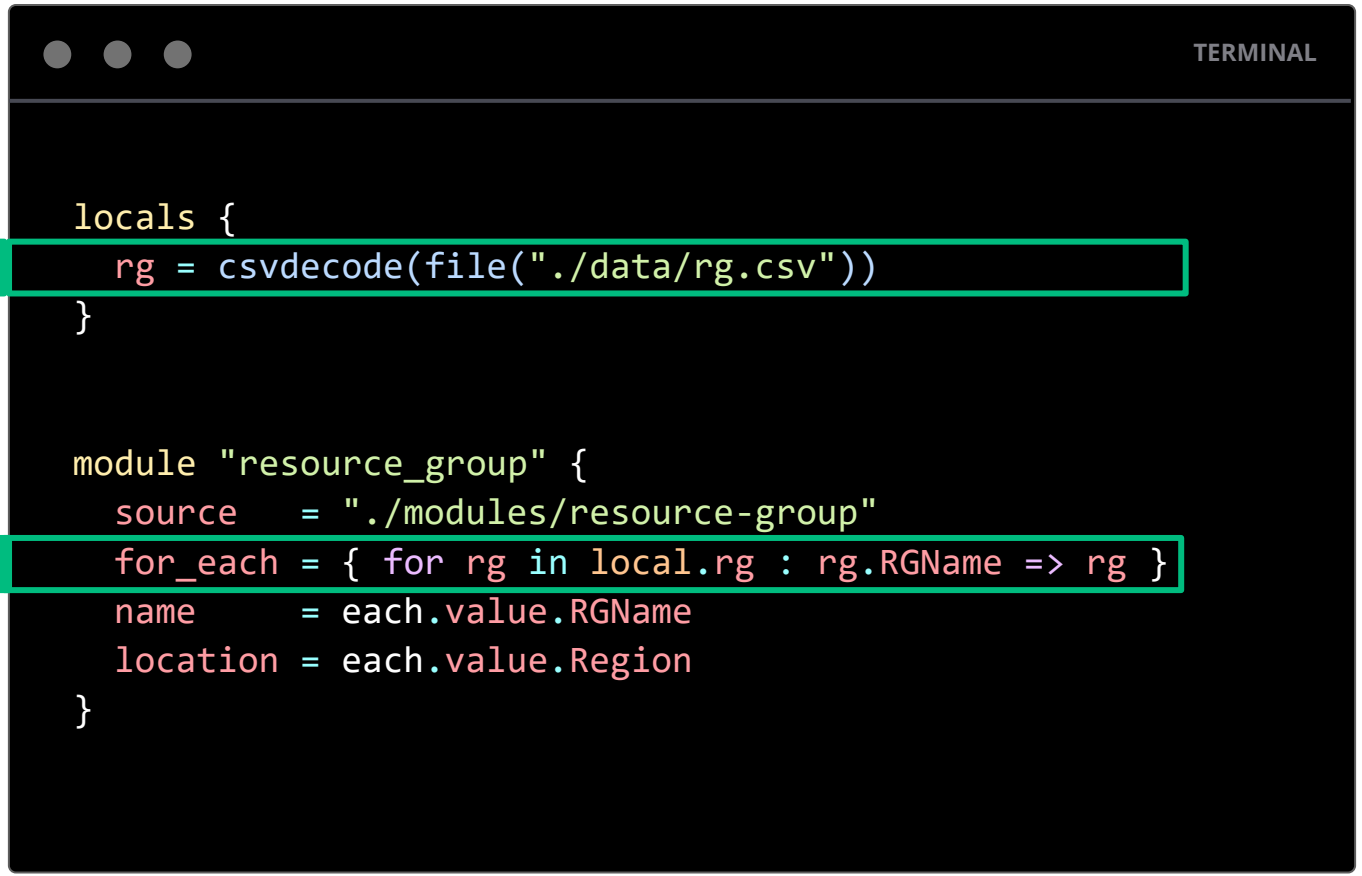
Region	RGName
Australia SouthEast	aus-hashitalks-01
Australia SouthEast	aus-hashitalks-02
Australia SouthEast	aus-hashitalks-03
Australia East	aue-hashitalks-01
Australia East	aue-hashitalks-02
Australia East	aue-hashitalks-03
New Zealand North	nzn-hashitalks-01
New Zealand North	nzn-hashitalks-02
New Zealand North	nzn-hashitalks-03

csvdecode under the hood



csvdecode & for_each

Better together


TERMINAL
locals {
rg = csvdecode(file("./data/rg.csv"))
}
module "resource_group" {
source = "./modules/resource-group"
for_each = { for rg in local.rg : rg.RGName => rg }
name = each.value.RGName
location = each.value.Region
}

Don’t Repeat Yourself (DRY)

CSV datasets can be called by various TF modules and resources to maximise code reusability and minimise coding effort.

TERMINAL

```
locals {
  vm = csvdecode(file("./data/vm.csv"))
}


module “virtual_machine" {
  depends_on     = [module.resource_group]
  source         = "./modules/virtual-machine"
  for_each       = { for vm in local.vm : vm.VMName => vm }
  name           = each.value.VMName
  resource_group = each.value.RGName
}

resource “azurerm_backup_protected_vm" {
  depends_on     = [module.virtual_machine]
  for_each       = { for vm in local.vm : vm.VMName => vm }
  resource_group = each.value.RGName
  source_vm_id   =
format("/subscriptions/xxxx/resourceGroups/%s/providers/Microsoft.Compute/v
irtualMachines/%s", each.value.RGName, each.value.VMName)
}
```

RGName	VMName
aus-hashitalks-01	aushashtalk01
aue-hashitalks-01	auehashitalk01
nzn-hashitalks-01	nznhashitalk01

Lessons learnt

Costly mistakes

Notes from the field using csvdecode and for_each

- Space(s) within a String {“foo ”}
- Empty/extra cells {“”}
- List values seen as a String {“foo,bar,baz”}
- UniqueID column changes {“foo”} → {“bar”}

Quick demo

Where to from here?

- Test out **csvdecode** and **for_each** yourself
- Don’t Repeat Yourself (**DRY**) strategy
- Look for **economies of scale** with datasets
- Beware of dataset **gotchas**
- Plan for **changes** (adds,removes,updates)
- **Terraform + CSVs = Netflix** | Lucian Franghiu

Thank you
& take care!

DRY coding with Terraform, CSVs, ForEach

Jesse Loudon
Principal Consultant

coder_au

jloudon.com
jesseloudon