Updating Your Configuration with More Resources

Ned Bellavance
MICROSOFT AZURE MVP
@ned1313 | nedinthecloud.com

Overview

Terraform state

Planning updates

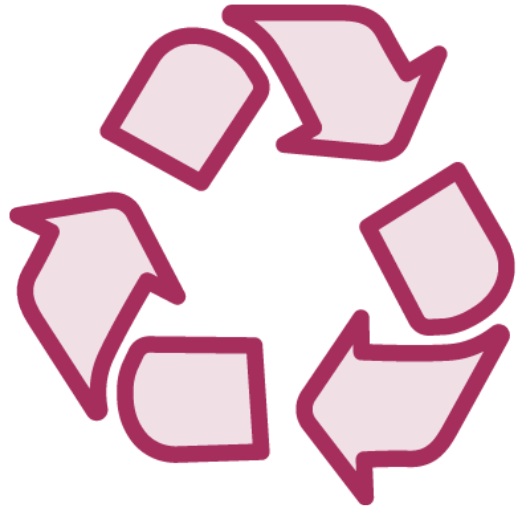
Our scenario evolves

Automating Infrastructure Deployment

Provisioning resources

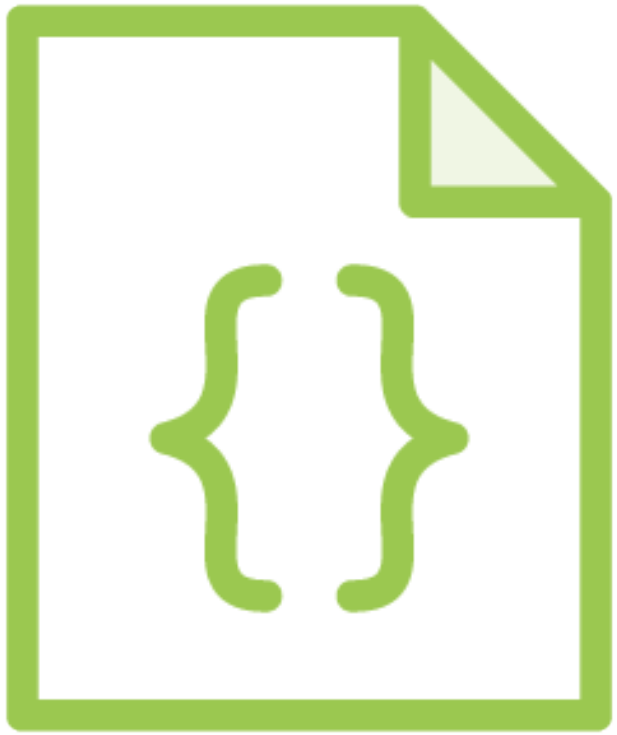
Planning updates

Using source control



Reusing templates

Terraform State



JSON format (Do not touch!)

Resources mappings and metadata

Locking

Location

- Local
- Remote: AWS, Azure, NFS, Terraform Cloud

Workspaces

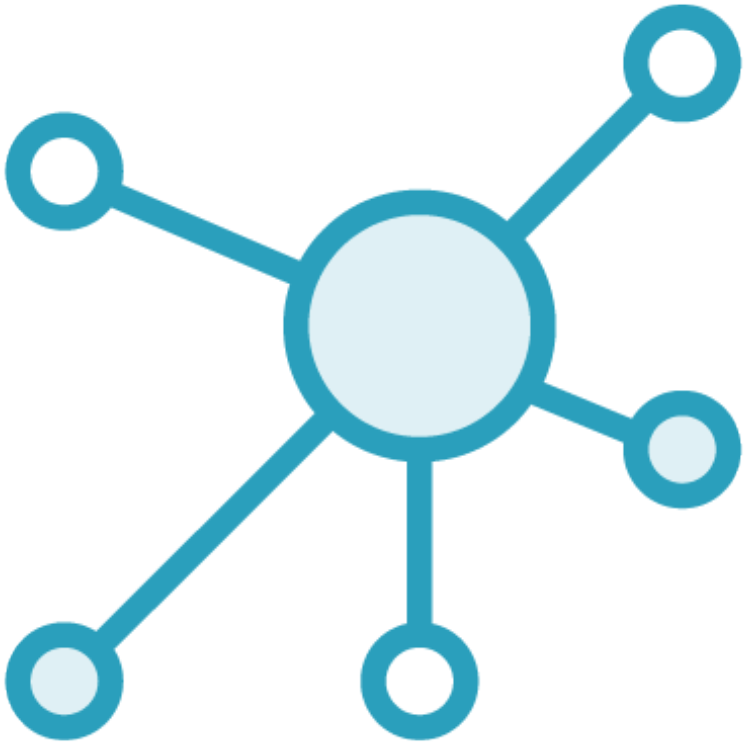
State File


```
{
  "version": 4,
  "terraform_version": "0.12.5",
  "serial": 30,
  "lineage": "",
  "outputs": {},
  "resources": []
}
```


First rule of Terraform?

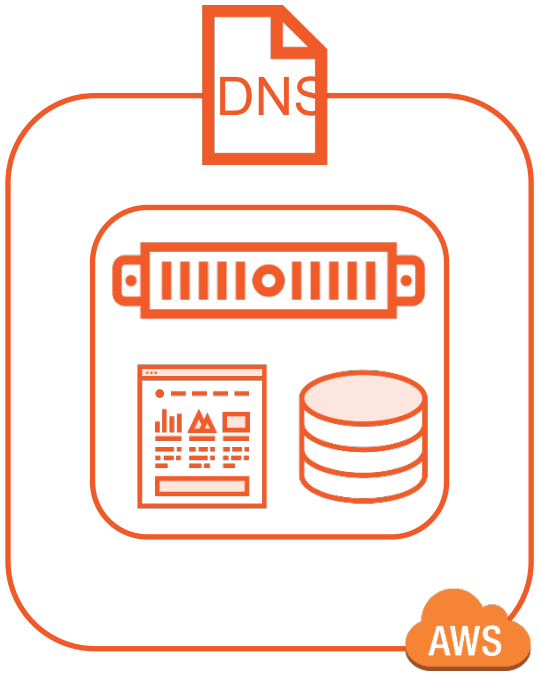
Make all changes in Terraform.

Terraform Planning

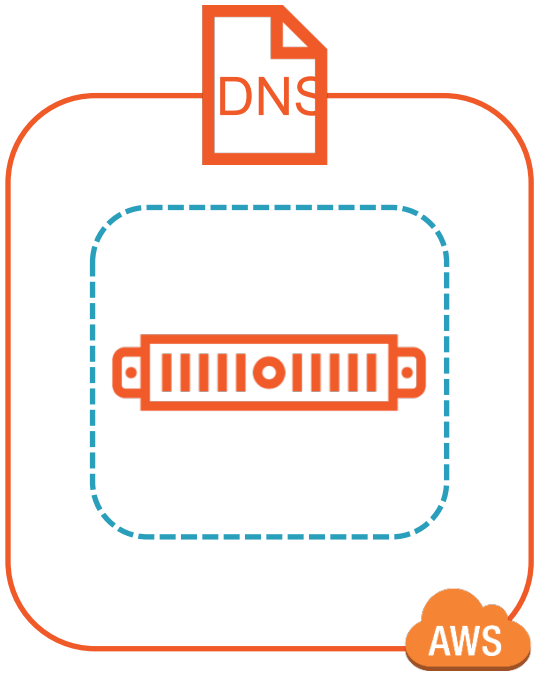


- Inspect state
- Dependency graph
- Additions, updates, and deletions
- Parallel execution
- Save the plan

The Scenario



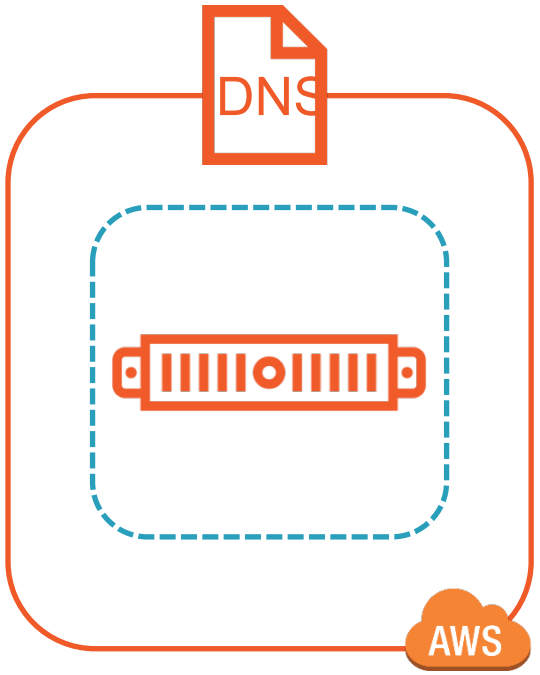
The Scenario



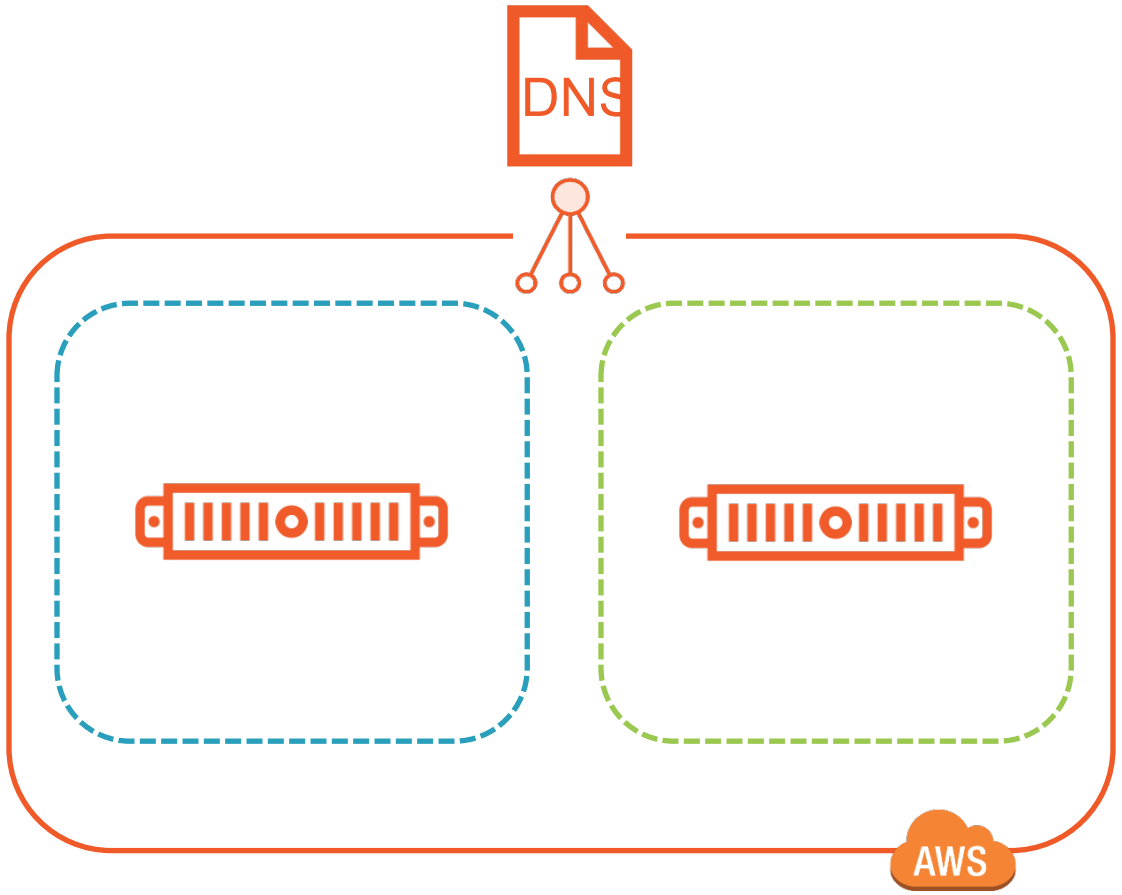
Adding a VPC

```
resource "aws_vpc" "vpc" {}

resource "aws_internet_gateway" "igw" {}

resource "aws_subnet" "subnet1" {}

resource "aws_route_table" "rtb" {}

resource "aws_route_table_association" "rta-subnet1" {}
```

The Scenario



The Scenario



Summary

Terraform updates and state file

Data sources

Load balancer and security

Coming up

- Provisioners
- Tagging
- Syntax (Woo-hoo!)