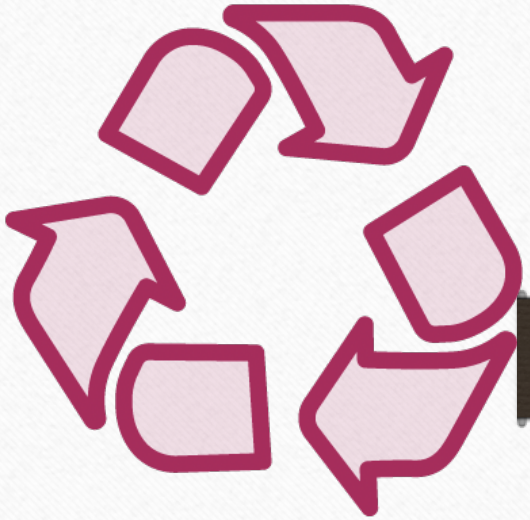
Automating Infrastructure Deployment

Provisioning resources

Planning Updates

Using Source Control

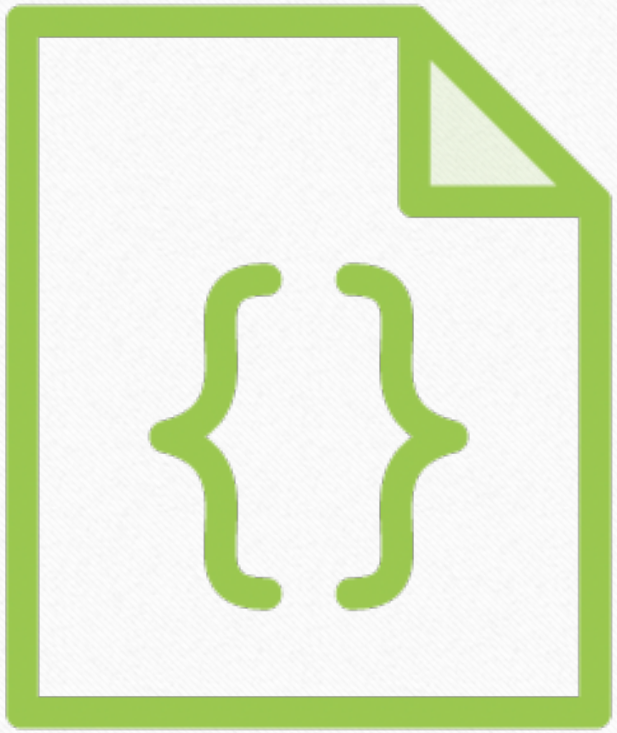


Reusing templates

Terraform State

- Terraform must store state about your managed infrastructure and configuration.
- This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures

Terraform State



JSON format (Do not touch!)

Resources mappings and metadata Locking

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

Workspaces

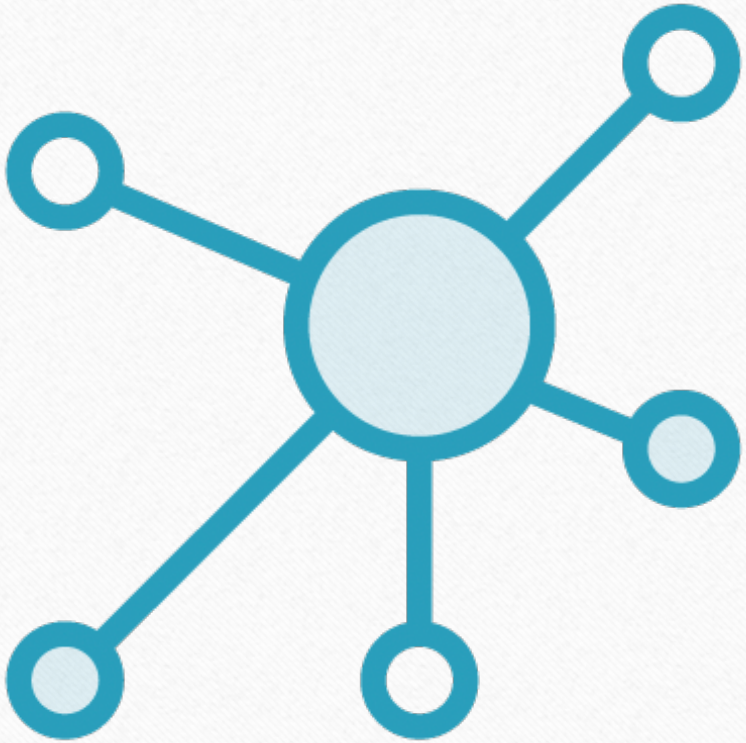
State file


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

Terraform Plan

The terraform plan command is used to create an execution plan. Terraform performs a refresh, unless explicitly disabled, and then determines what actions are necessary to achieve the desired state specified in the configuration files.

Terraform Planning



Inspect state

Dependency graph

Additions, updates, and deletions

Parallel execution

Save the plan

Scenario Part1

- Launch the instances in customized VPC(previous one is launched the default one)

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" {}
```

Scenario Part2

- Launch the instances in multiple available zones to increase availability of the service
- Launch load balancer to balance the load between the instances with public dns name

Summary

Terraform updates and state file

Data sources

Load balancer and security