

Updating Your Configuration with More Resources



Ned Bellavance

MICROSOFT MVP, CLOUD AND DATACENTER MANAGEMENT

@ned1313 www.nedinthecloud.com



Overview



Consistent and predictable updates

Our scenario evolves

Let's get balanced and secure



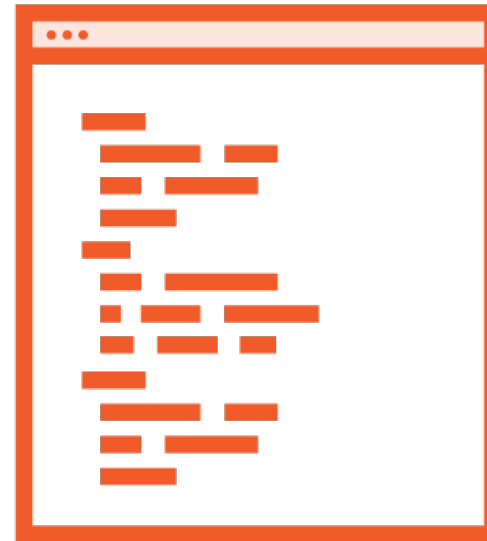
Automating Infrastructure Deployment



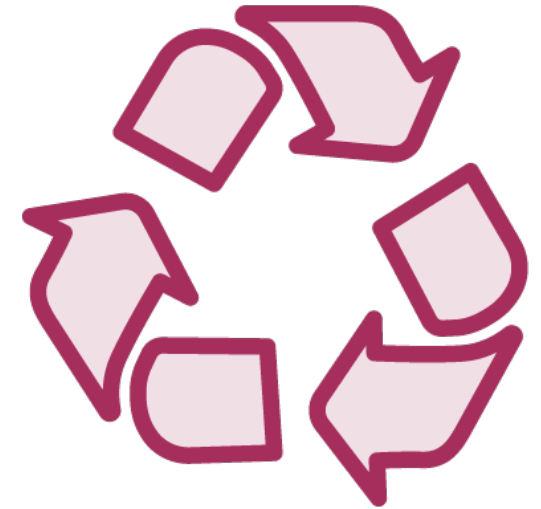
Provisioning
resources



Planning
updates

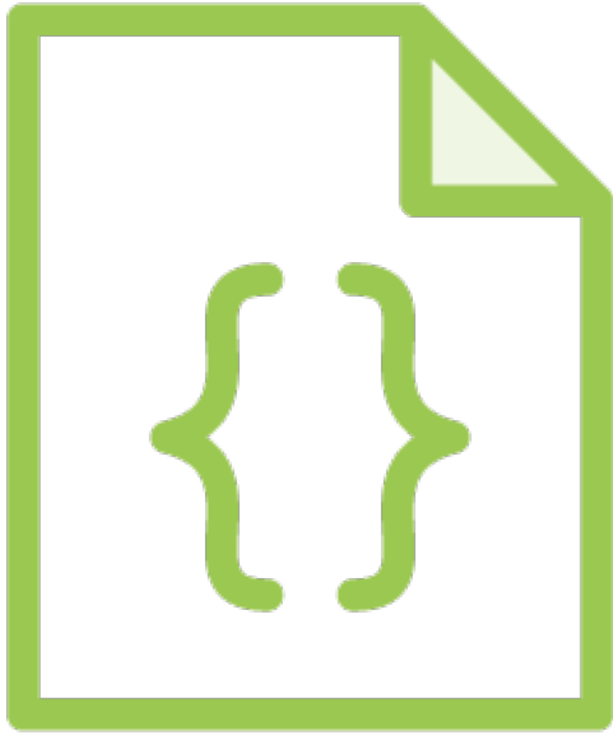


Using source
control



Reusing
templates

Terraform State



JSON format (Do not touch!)

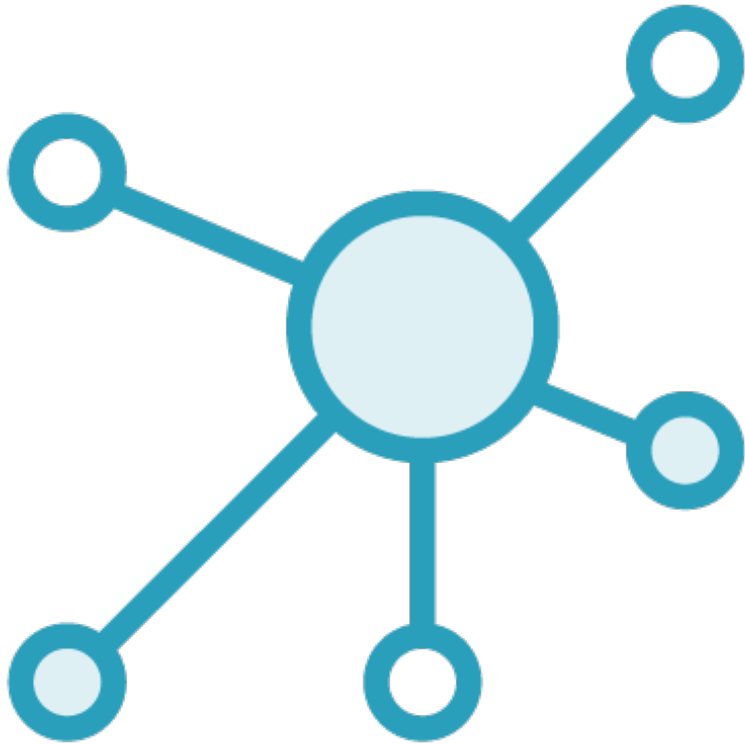
Resources mappings and metadata

Locking

Local / remote

Environments

Terraform Planning



Inspect state

Dependency graph

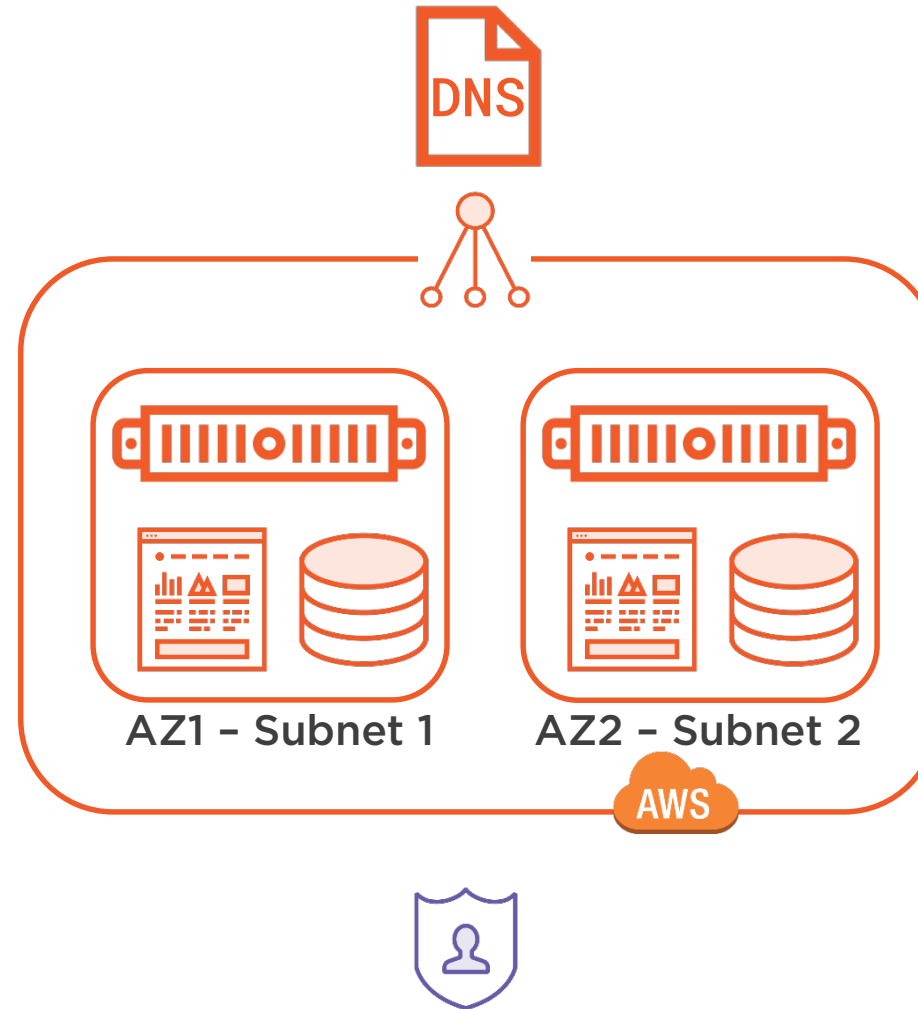
Additions and deletions

Walk the line

The Scenario



The Scenario



```
data "aws_availability_zones"  
  "available" {}
```

```
resource "aws_subnet" "subnet1"  
{  
  cidr_block =  
    "${var.subnet1_address_space}"  
  vpc_id = "${aws_vpc.vpc.id}"  
  availability_zone =  
    "${data.aws_availability_zones.a  
available.names[0]}"  
}
```

◀ Data source

◀ Select AZ




```
resource "aws_security_group"
"elb-sg" {
  name = "nginx_elb_sg"
  vpc_id = "${aws_vpc.vpc.id}"

  ingress {}
  egress {}
}

resource "aws_elb" "web" {
name = "nginx-elb"

  security_groups =
[ "${aws_security_group.elb-
sg.id}" ]
}
```

◀ Security group

◀ Load balancer



Demo



Examine the Terraform file

Deploy the configuration

Review the results

Play along!

- AWS account
- Terraform software (terraform.io)
- Demo files



Some of the resources
deployed in AWS may cost
money. You've been
warned.



Summary



Terraform updates and state file

Data sources

Load balancer and security

Coming up

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

