Using Modules for Common Configurations

Ned Bellavance
MICROSOFT AZURE MVP
@ned1313 | nedinthecloud.com

Overview

Another day, another request

Modules make code reuse easier

Outputs and process flow

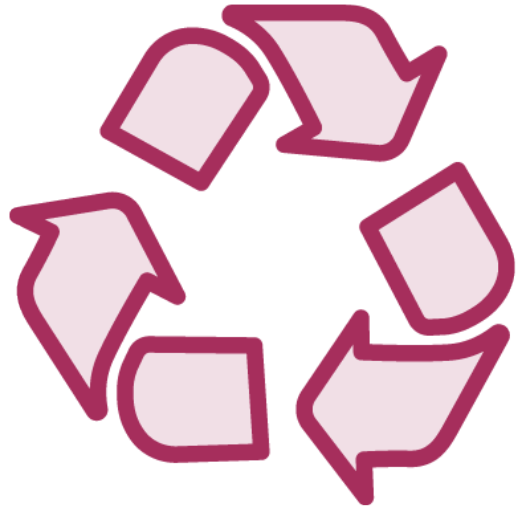
Putting it all together

Automating Infrastructure Deployment

Provisioning resources

Planning updates

Using source control



Reusing templates

The Scenario

App1

App2

App3

App4

AWS

VPC
Subnets
Internet gateway
Routing

VPC module

S3 module

Terraform Modules

Code reuse

Remote or local source

- Terraform Registry

Root module

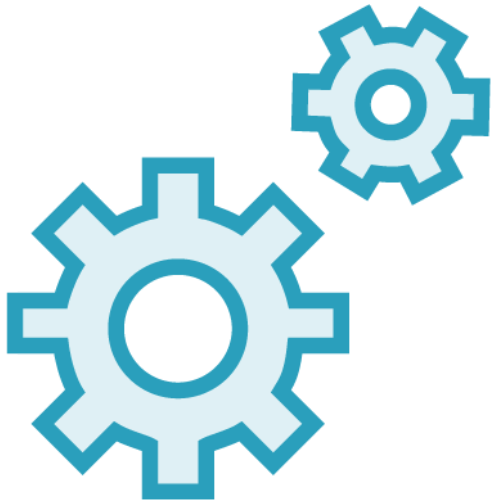
Versioning

Provider inheritance

Multiple instances (no count)

Module Components

Input variables



Resources

Output values

Terraform Module

```
variable "name" {}

resource "aws_s3_bucket" "bucket"{
  name = var.name
  […]
}

output "bucket_id" {
  value = aws_s3_bucket.bucket.id
}
```

Terraform Module

```
#Create module bucket
module "bucket" {

  name = "taco-bucket"
  source = ".\\Modules\\s3"

}

#Use taco-bucket
resource "aws_s3_bucket_object" {

  bucket = module.bucket.bucket_id
  […]

}
```

Summary

Terraform modules

Code reuse

Abstraction

What's next?

Course Summary

Build infrastructure automagically

Ensure consistent repeatable deployment

Reuse existing configurations

Increase your productivity

Make your job better or find a better job!

Go Build Something Great!