# Using a Module for Common Configurations



Ned Bellavance
HashiCorp Ambassador

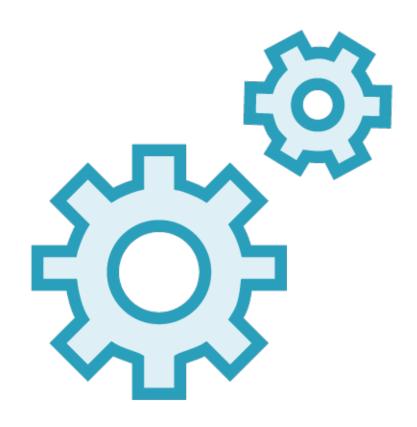
@ned1313 nedinthecloud.com

#### Overview

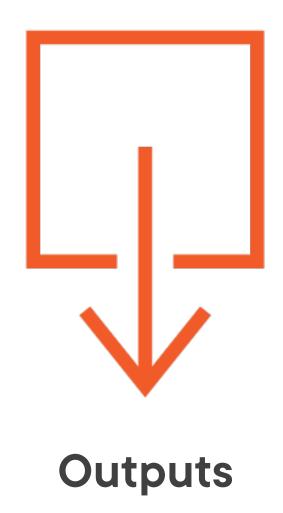


What is a module?
Globomantics updates
Using existing modules
Creating new modules

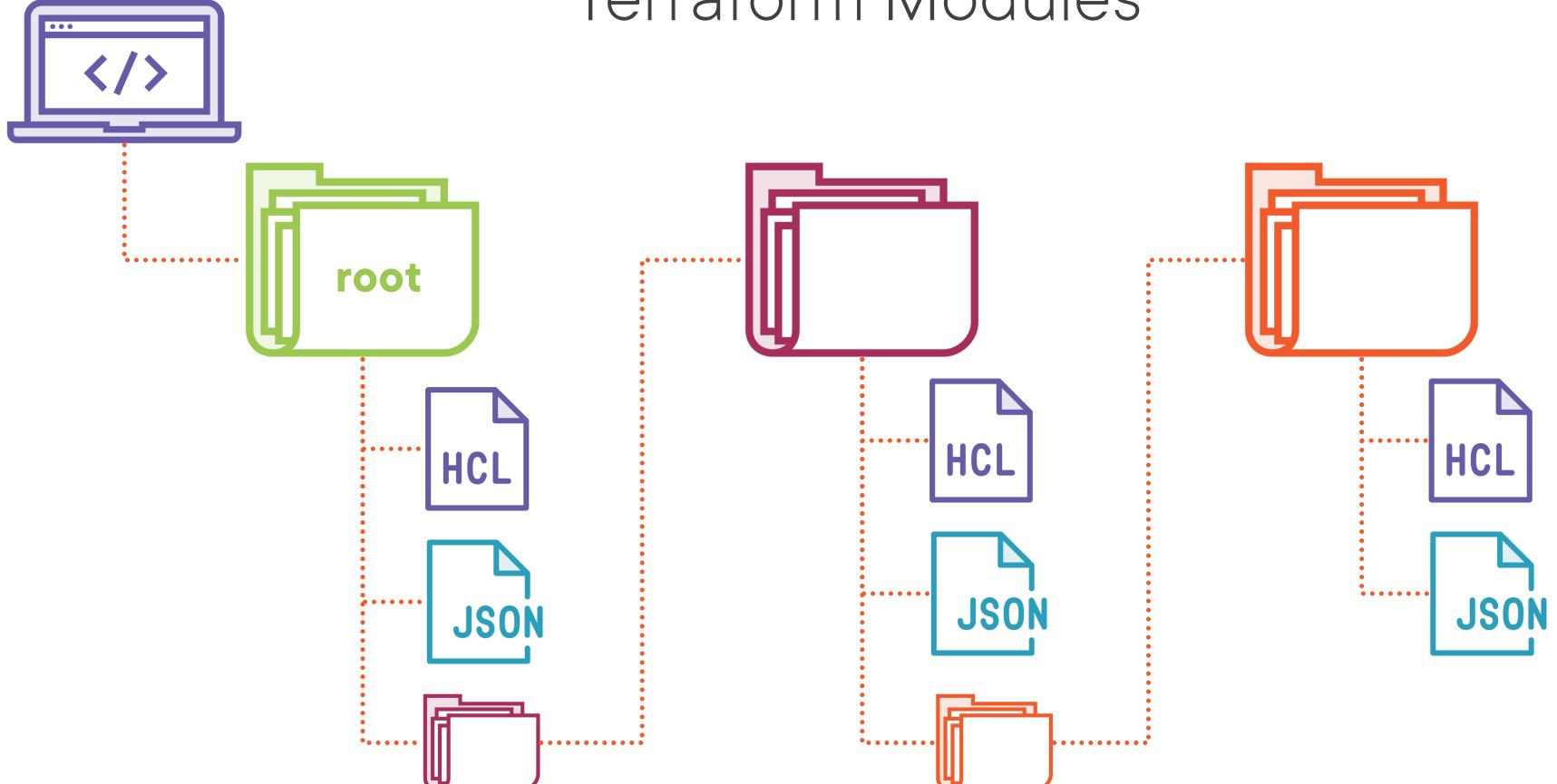


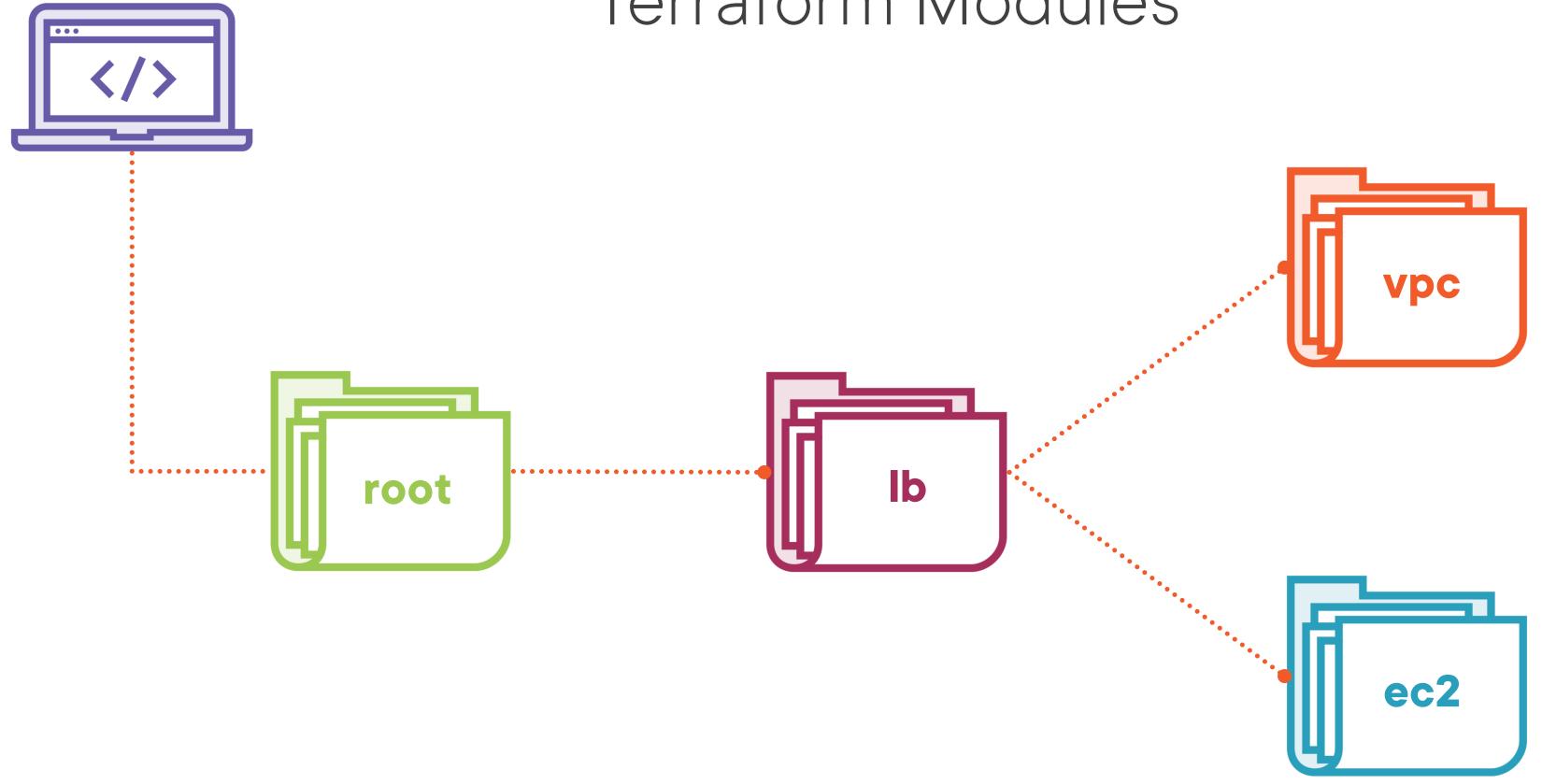














**Code reuse** 

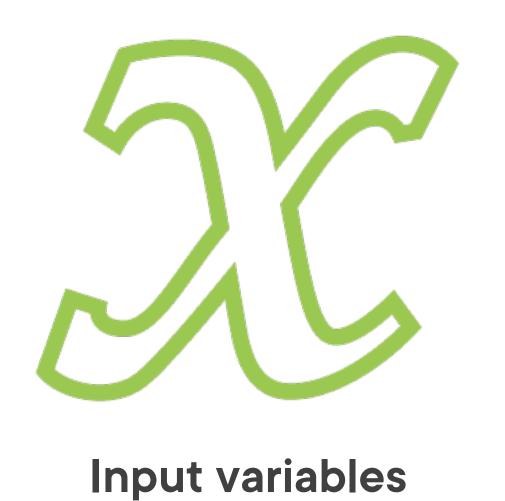
Remote or local source

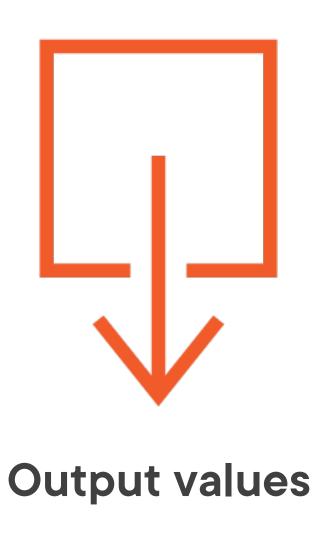
Versioning

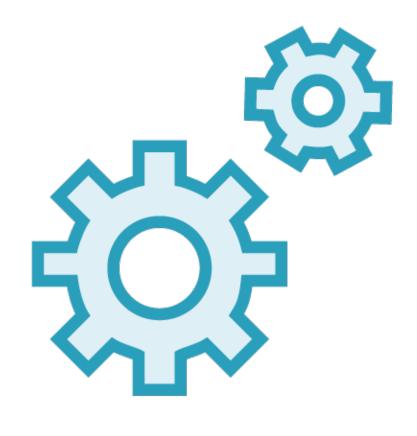
**Terraform init** 

Multiple instances

# Module Components



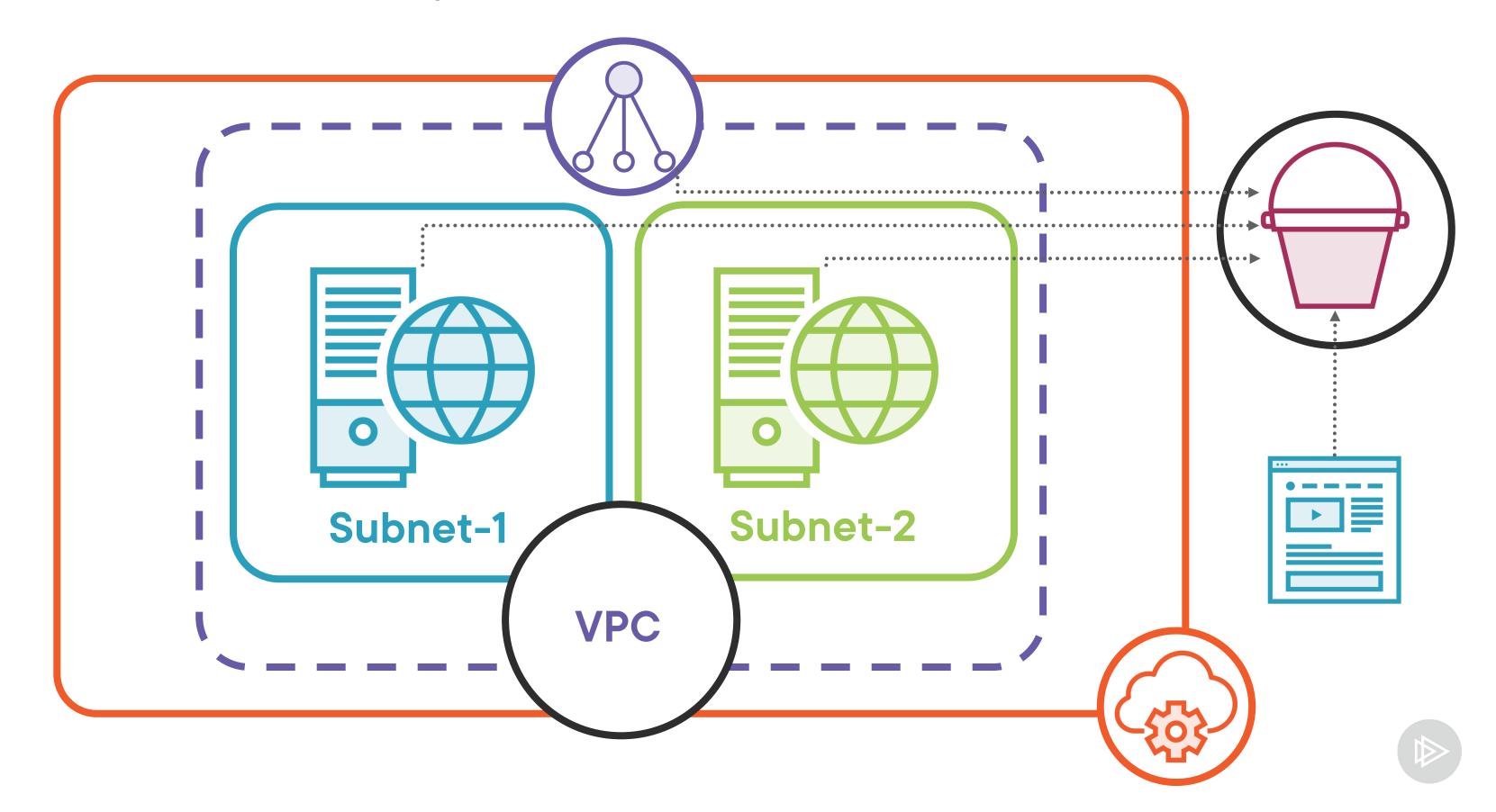




Resources
Data Sources

# Globomantics Scenario

# Deployment Architecture



# Potential Improvements



Leverage the VPC module for networking

Create a module for S3 buckets

- Include load balancer permissions
- Include instance profile permissions

# Module Structure and Syntax



#### s3/main.tf

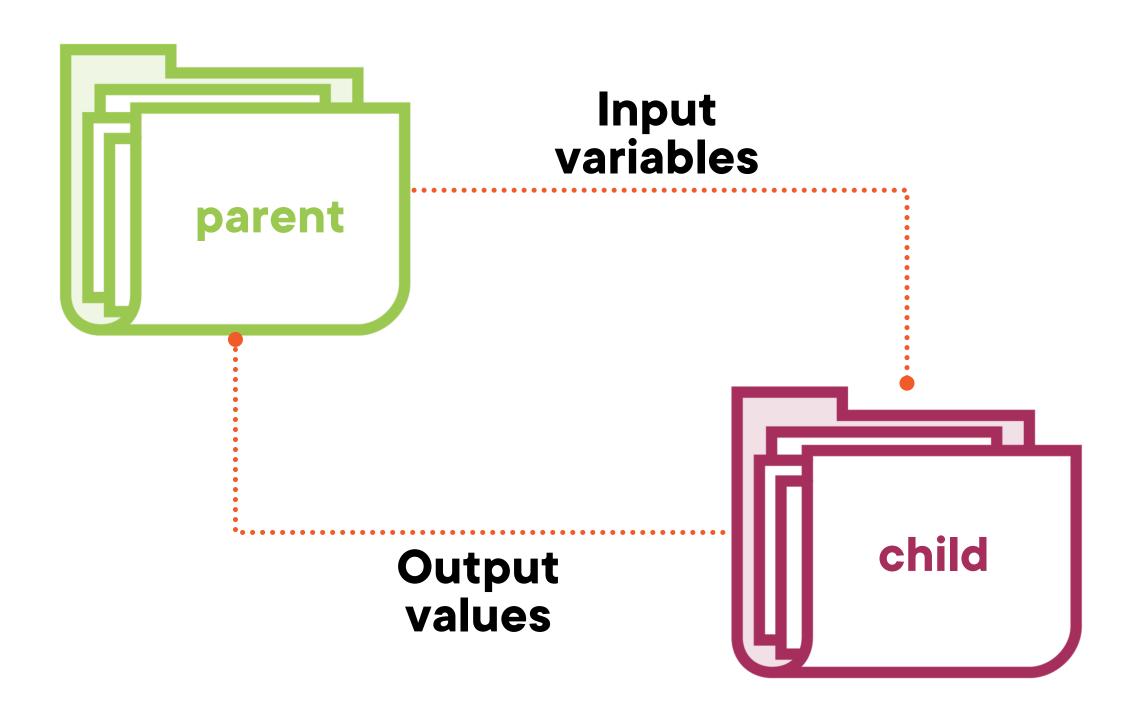
### Module Structure

```
variable "bucket_name" {}

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

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

# Scoping



# Module Syntax

```
module "name_label" {
  source = "local_or_remote_source"
  version = "version_expression"
  providers = {
    module_provider = parent_provider
  }

# Input variable values...
}
```

# Module Syntax

```
module "taco_bucket" {
  source = "./s3"

# Input variable values...
  bucket_name = "mah_bucket"
}
```

```
module "taco_bucket" {
  source = "./s3"

# Input variable values...
  bucket_name = "mah_bucket"
}
```

#### Module References

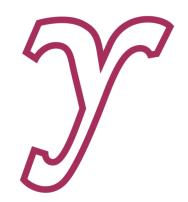
```
module.<name_label>.<output_name>
module.taco_bucket.bucket_id
```

# For Expressions



Input types

List, set, tuple, map, or object



**Result types** 

**Tuple or object** 



Filtering with if statement

# For Tuple Syntax

```
main.tf
```

```
# Create a tuple
[for item in items : tuple_element]
# Example
locals {
 toppings = ["cheese", "lettuce", "salsa"]
[for t in local.toppings : "Globo ${t}"]
# Result
["Globo cheese", "Globo lettuce", "Globo salsa"]
```

# For Object Syntax

#### main.tf

```
# Create an object
{ for key, value in map : obj_key => obj_value }
# Example
locals {
 prices = {
  taco = "5.99"
  burrito = "9.99"
  enchilada = "7.99"
{ for i, p in local.prices : i => ceil(p) }
# Result
{ taco = "6", burrito = "10", enchilada = "8"}
```

#### S3 Module

```
# Input variables – variables.tf
"bucket_name" # Name of bucket
"elb_service_account_arn" # ARN of ELB service account
"common_tags" # Tags to apply to resources
# Resources – main.tf
"aws_s3_bucket"
"aws_iam_role"
"aws_iam_role_policy"
"aws_iam_instance_profile"
# Outputs – outputs.tf
"web_bucket" # Full bucket object
"instance_profile" # Full instance profile object
```



## Summary



Modules enable code reuse

**Common configurations** 

**Root module** 



# Up Next: Using Workspaces for Multiple Environments