Using Functions and Looping in Your Configuration

Ned Bellavance

HashiCorp Ambassador

@ned1313 nedinthecloud.com

Overview

Globomantics requests

Loops and dynamic blocks

Using functions

Terraform console

Globomantics Scenario

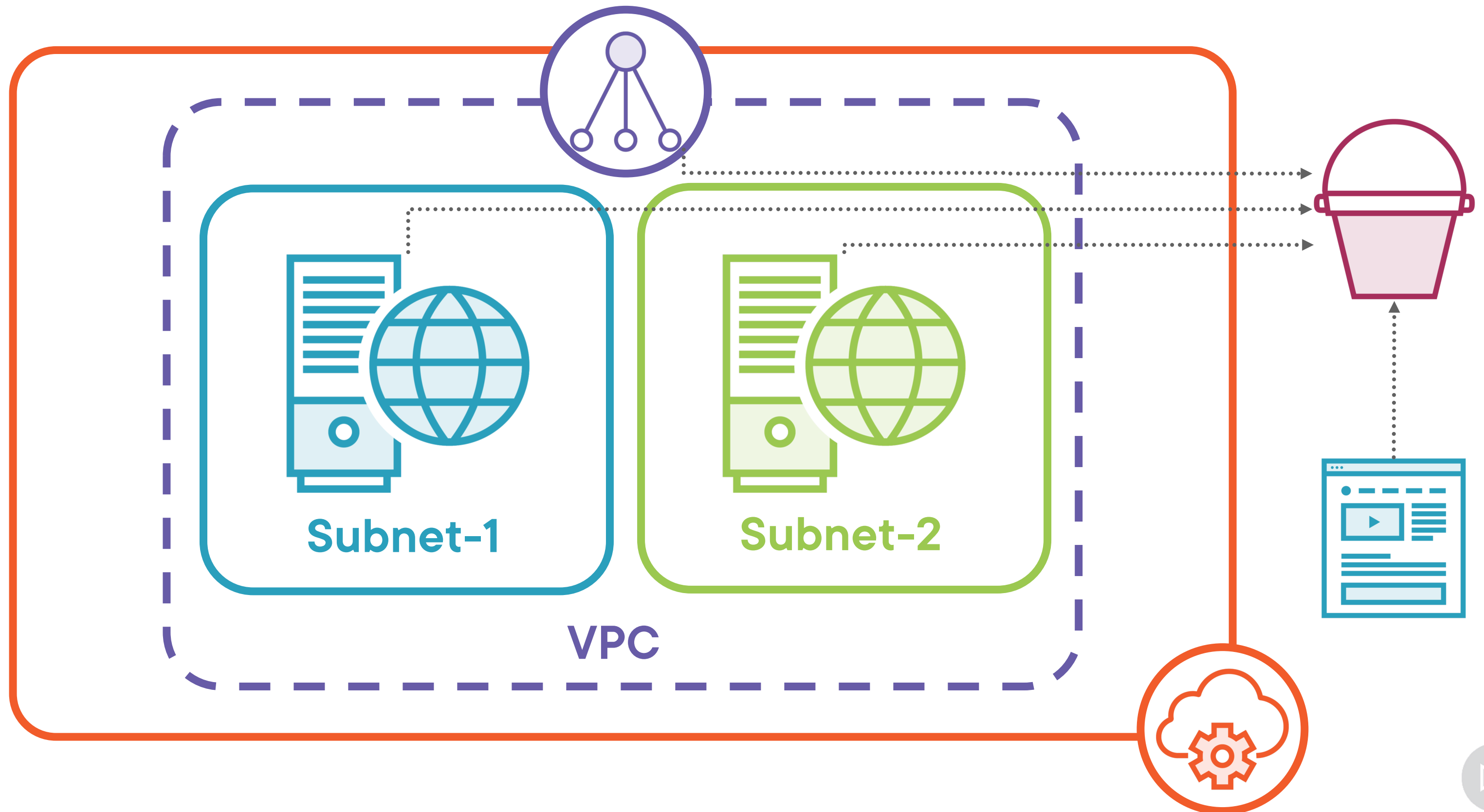# Potential Improvements

Dynamically increase instances

Use a template for startup script

Simplify networking input

Add consistent naming prefix
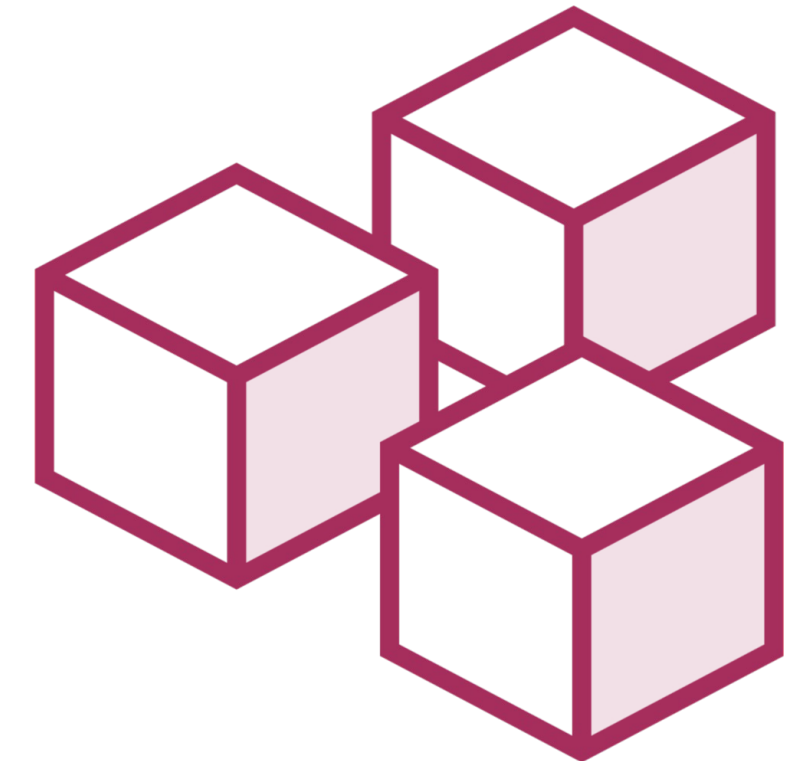
Deployment Architecture

Loops in Terraform

Looping Constructs

Count

Integer

For_each

Map or set

Dynamic blocks

Map or set

Count Syntax

```
resource "aws_instance" "web_servers" {

  count = 3

  tags = {

    Name = "globo-web-${count.index}"

  }

}
```

```
resource "aws_instance" "web_servers" {

  count = 3

  tags = {

    Name = "globo-web-${count.index}"

  }

}
```

Count References

<resource_type>.<name_label>[element].<attribute>

aws_instance.web_servers[0].name # Single instance

aws_instance.web_servers[*].name # All instances

For_each Syntax

```
resource "aws_s3_bucket_object" "taco_toppings" {

  for_each = {

    cheese = "cheese.png"

    lettuce = "lettuce.png"

  }

  key    = each.value

  source = ".${each.value}"

  tags = {

    Name = each.key

  }

}
```

```
resource "aws_s3_bucket_object" "taco_toppings" {

  for_each = {

    cheese = "cheese.png"

    lettuce = "lettuce.png"

  }

}
```

For_each References

<resource_type>.<name_label>[key].<attribute>

aws_s3_bucket_object.taco_toppings["cheese"].id # Single instance

aws_s3_bucket_object.taco_toppings[*].id # All instances

Looping Targets

```
# Primary resources

"aws_subnets" # Count loop

"aws_instance" # Count loop

"aws_s3_bucket_object" # For_each loop

# Impacted resources

"aws_route_table_association"

"aws_lb_target_group_attachment"
```

Terraform Functions and Expressions
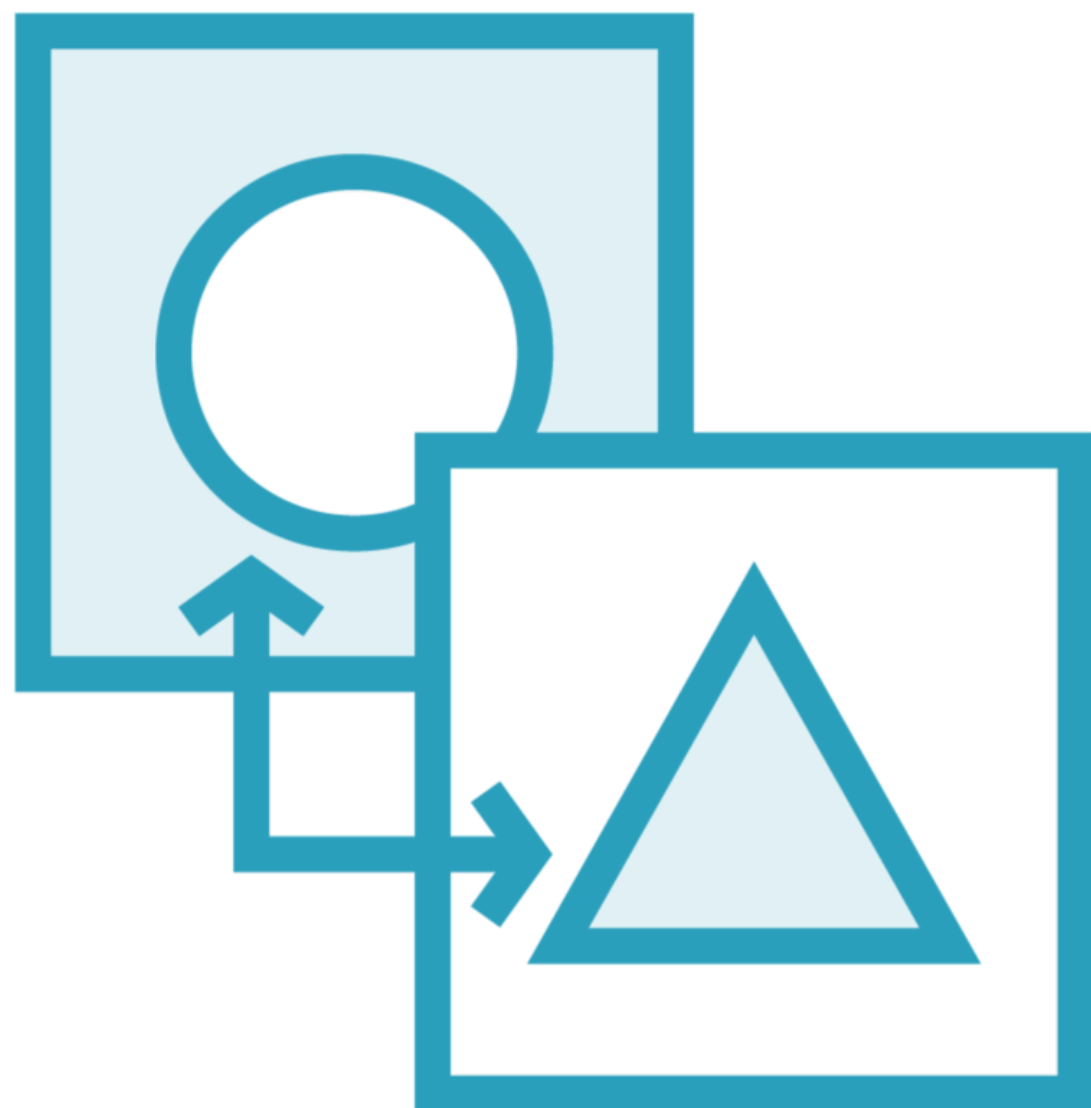
Terraform Expressions

Interpolation and heredoc

Arithmetic and logical operators

Conditional expressions

For expressions

Terraform Functions

Built-in Terraform

Func_name(arg1, arg2, arg3, ...)

Test in terraform console

Several broad categories

Common Function Categories

Numeric

min(42, 13, 7)

String

lower("TACOS")

Collection

merge(map1, map2)

IP network

cidrsubnet()

Filesystem

file(path)

Type Conversion

toset()

Functions to Use

```
# Startup script

templatefile(file_location, { map of variables })

# Extract subnet address from VPC CIDR

cidrsubnet(cidr_range, subnet bits to add, network number)

# Add tags to common tags

merge(common_tags, { map of additional tags })

# S3 bucket name

lower("bucket name")
```

Summary

Looping for dynamic configurations

Applying functions for transformation

Up Next:
Using a Module for Common Configurations