Using Input Variables and Outputs

Ned Bellavance

HashiCorp Ambassador

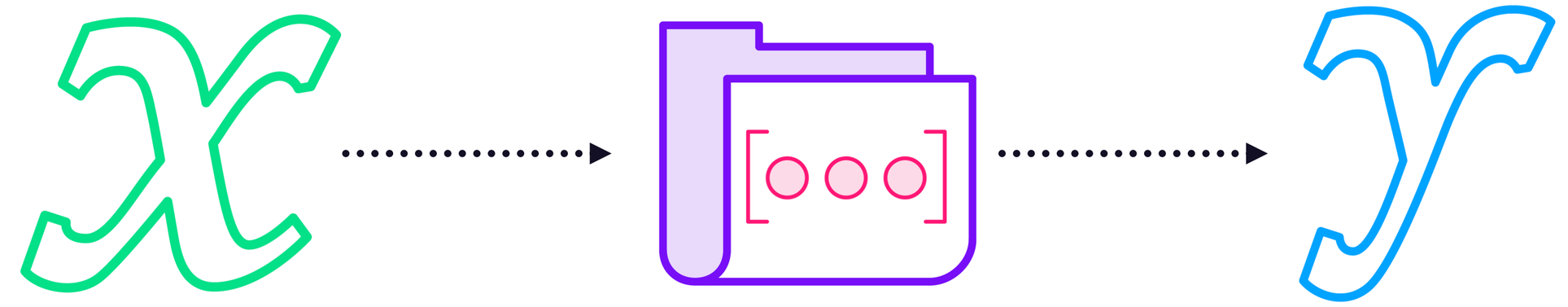
@ned1313 | nedinthecloud.com

Module Overview

- Supplying inputs
- Constructing values
- Specifying outputs
- Validate configurations

Working with Data in Terraform

Input Variables and Output Values



Variable Syntax

main.tf

```
variable "name_label" {}

variable "name_label" {
  type        = value
  description = "string"
  default     = value
  sensitive   = true | false
}
```

Variable Syntax

main.tf

```
variable "billing_tag" {}

variable "aws_region" {
  type        = string
  description = "Region to use for AWS resources"
  default     = "us-east-1"
  sensitive   = false
}
```

```
variable "aws_region" {
  type        = string
  description = "Region to use for AWS resources"
  default     = "us-east-1"
}
```

Terraform Variable Reference

`var.<name_label>`

`var.aws_region`

Terraform Data Types

Primitive

String, number, bool

Collection

List, set, map

Structural

Tuple, object

Any
list(any)

Null

Data Type Examples

```
# List
[1, 2, 3, 4]
["us-east-1", "us-east-2", "us-west-1", "us-west-2"]
[1, "us-east-2", true] # INVALID LIST!

# Map
{
  small  = "t2.micro"
  medium = "t2.small"
  large  = "t2.large"
}
```

```
variable "aws_regions" {
  type        = list(string)
  description = "Regions to use for AWS resources"
  default     = ["us-east-1", "us-east-2", "us-west-1", "us-west-2"]
}
```

Referencing Collection Values

`var.<name_label>[<element_number>]`

`var.aws_regions[0]`

```
variable "aws_instance_sizes" {
  type = map(string)
  description = "Instance sizes to use in AWS"
  default = {
    small  = "t3.micro"
    medium = "m4.large"
    large  = "m4.xlarge"
  }
}
```

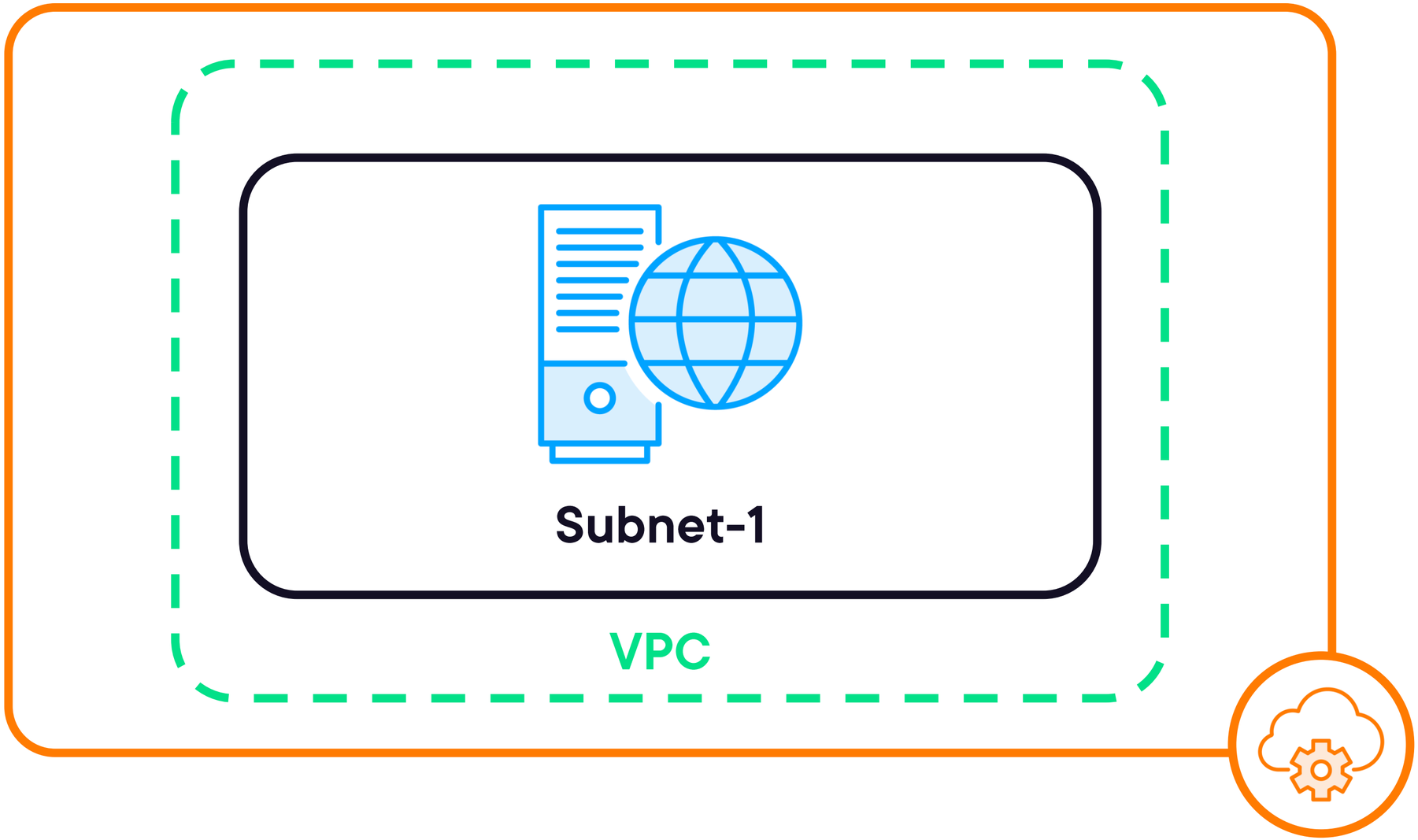
Referencing Collection Values

`var.<name_label>.<key_name> or var.<name_label>["key_name"]`

`var.aws_instance_sizes.small or var.aws_instance_sizes["small"]`

Globomantics Scenario

Deployment Architecture



Potential Improvements

Remove AWS credentials

Replace hard coded values

Tags for company, project, and billing

Generate output of public DNS hostname

Local Values

Internal to the configurations

Values with multiple references

Data transformation

Locals Syntax

main.tf

```
locals {
  key = value
}
```

Locals Syntax

main.tf

```
locals {
  instance_prefix = "globo"
  common_tags = {
    company      = "Globomantics"
    project      = var.project
    billing_code = var.billing_code
  }
}
```

```
locals {
  instance_prefix = "globo"
  common_tags = {
    company      = "Globomantics"
    project      = var.project
    billing_code = var.billing_code
  }
}
```

Terraform Locals Reference

local.<key>

local.instance_prefix

local.common_tags.company

Common Tags

Company tag

Default by Globomantics

Project tag

Company-project

No default

Billing code tag

No default

Outputs Syntax

main.tf

```
output "name_label" {
  value       = value
  description = "string"
  sensitive   = true | false
}
```

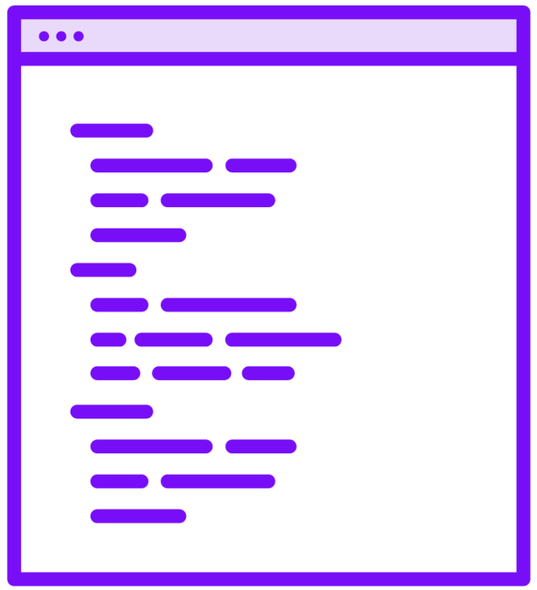
Outputs Syntax

main.tf

```
output "public_dns_hostname" {
  value       = aws_instance.web_server.public_dns
  description = "Public DNS hostname of web server"
}
```

Validate and Update Your Configuration

Formatting with terraform fmt



Formats code in working directory

Follows HashiCorp's standard

Command flags

- check
- recursive

Does not check validity

Validation with terraform validate

Initialize before running

Checks syntax and logic

Does not check state

No guarantee of success

Supply Variable Values

Default value

-var flag

-var-file flag

terraform.tfvars

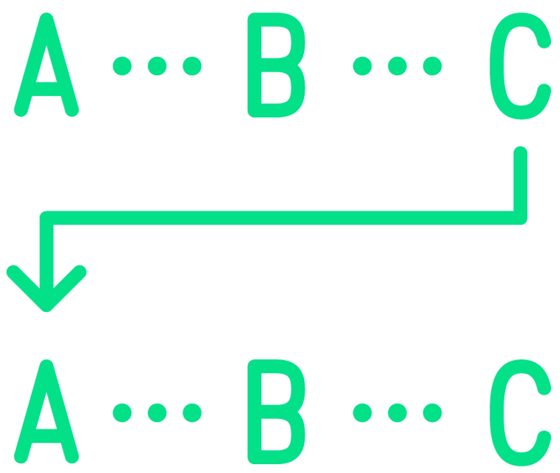
terraform.tfvars.json

.auto.tfvars

.auto.tfvars.json

TF_VAR_

Order of Evaluation



TF_VAR_ environment variable

terraform.tfvars or terraform.tfvar.json

.auto.tfvars or .auto.tfvars.json

-var-file flag

-var flag

Command line prompt

Module Summary

Supply data through input variables

Many ways to submit values

Receive data through outputs

Validate your configurations

Up Next:

Updating Your Configuration with More Resources