# Using Input Variables and Outputs



Ned Bellavance
HashiCorp Ambassador

@ned1313 nedinthecloud.com

### Overview



Supplying inputs

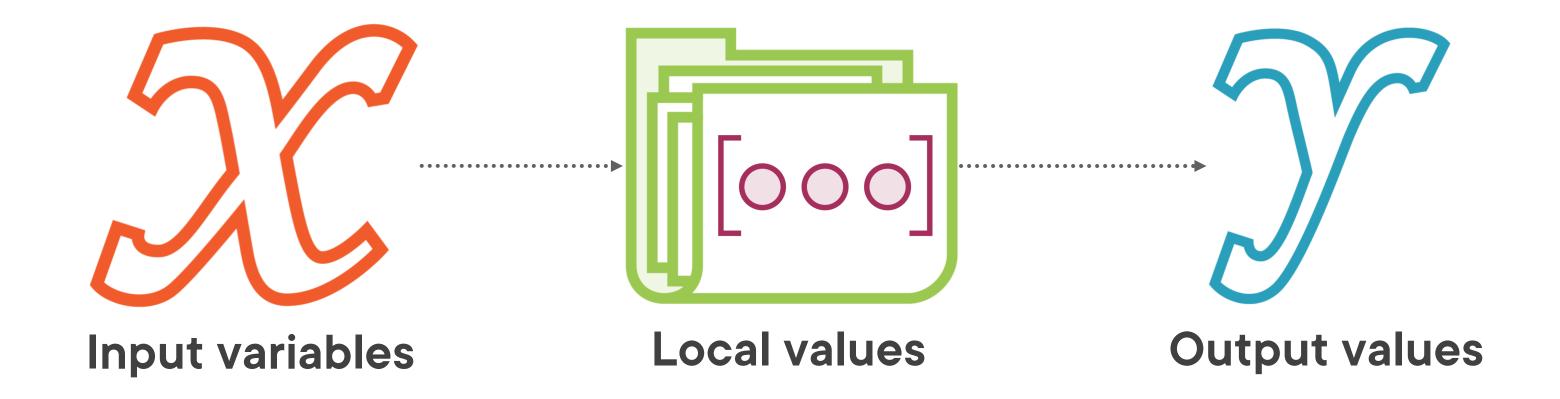
Constructing values

Specifying outputs

Validate configurations

## Working with Data in Terraform

## Variables and Outputs



#### main.tf

## Variable Syntax

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

main.tf

## Variable Syntax

```
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"
}
```

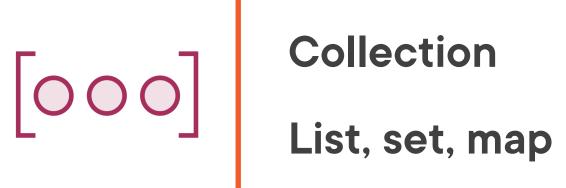
#### Terraform Variable Reference

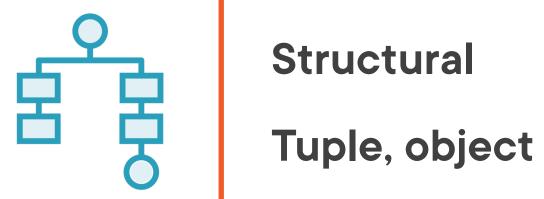
```
var.<name_label>
var.aws_region
```

## Terraform Data Types



Primitive
String, number, boolean





## 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 = "Region 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]
```

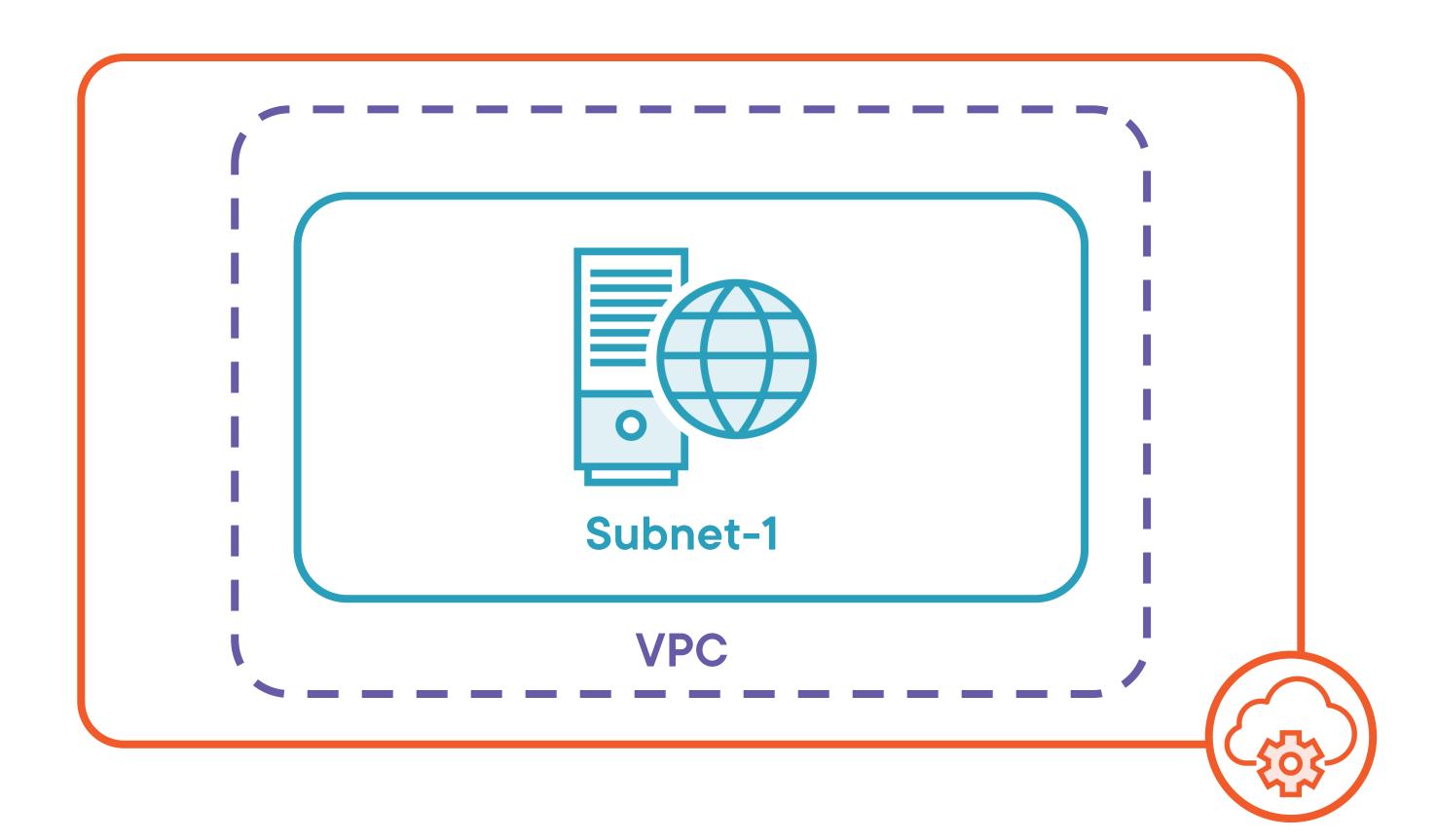
```
variable "aws_instance_sizes" {
type = map(string)
 description = "Region to use for AWS resources"
 default = {
  small = "t2.micro"
  medium = "t2.small"
  large = "t2.large"
```

## 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

#### main.tf

```
Locals Syntax
```

```
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.<name_label>
local.instance_prefix
local.common_tags.company
```

Outputs Syntax

#### main.tf

```
output "name_label" {
  value = output_value
  description = "Description of output"
  sensitive = true | false
}
```

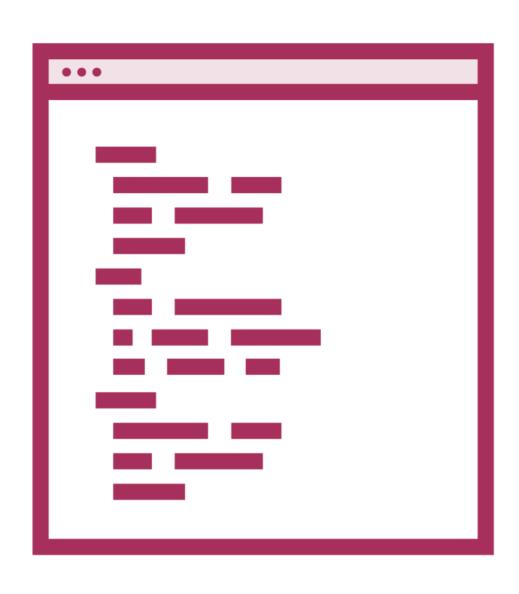
#### main.tf

## Outputs Syntax

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

## Validate and Update Your Configuration

## Syntax Validation



**Terraform init first** 

**Checks syntax and logic** 

Does not check state

No guarantee of deployment

## Supply Variable Values

**Default value** 

-var flag

-var-file flag

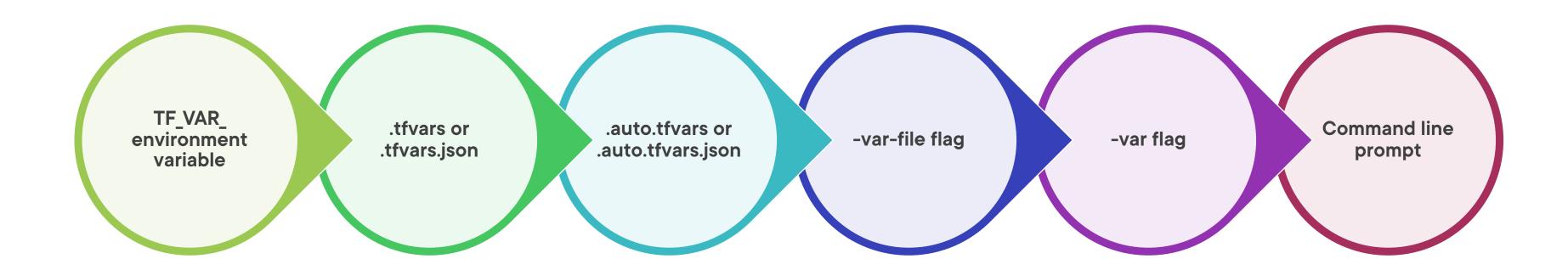
terraform.tfvars terraform.tfvars.json .auto.tfvars

.auto.tfvars.json

Environment variable TF\_VAR\_



## Evaluation Precedence



Last evaluated wins



## 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