Lab: Introduction to the Terraform Output Block

Terraform output values allow you to export structured data about your resources. You can use this data to configure other parts of your infrastructure with automation tools, or as a data source for another Terraform workspace. Outputs are also necessary to share data from a child module to your root module.

As with all other blocks in HashiCorp Configuration Language (HCL), the output block has a particular syntax that needs to be followed when creating output blocks. Each output name should be unique. The snytax looks like this:

Template

```
output “<NAME>” {
  # Block body
  value= <EXPRESSION> # Argument
}
```

Example

```
output "web_server_ip" {
  description = "Public IP Address of Web Server on EC2"
  value       = aws_instance.web_server.public_ip
  sensitive   = true
}
```

- Task 1: Add Output Block to Export Attributes of Resources
- Task 2: Output Meaningful Data using Static and Dynamic Values
- Task 3: Generate Machine-Readable Outputs in JSON

Task 1: Add Output Block to Export Attributes of Resources

In the same working directory that contains our `main.tf` and `variables.tf` files, create a new file called `outputs.tf`. This file is commonly used to store all output blocks within your working directory. In the new file, add the following code:

```
output "hello-world" {
  description = "Print a Hello World text output"
  value = "Hello World"
}

output "vpc_id" {
  description = "Output the ID for the primary VPC"
  value = aws_vpc.vpc.id
}
```

Task 1.1

After saving the new `outputs.tf`file, run a `terraform apply -auto-approve`to see our new outputs for our infrastructure.

```
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration
     and found no
differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

hello-world = "Hello World"
vpc_id = "vpc-058d23c9d5d2f70b5"
```

Task 2: Output Meaningful Data using Static and Dynamic Values

Resource attribute data can also be included within a string to include additional information or create a particular output, such as a website URL or database connection string. Modify the `outputs.tf` file to include the additional output blocks as shown below:

```
output "public_url" {
  description = "Public URL for our Web Server"
  value = "https://${aws_instance.web_server.private_ip}:8080/index.html"
}

output "vpc_information" {
  description = "VPC Information about Environment"
```

```
  value = "Your ${aws_vpc.vpc.tags.Environment} VPC has an ID of ${aws_vpc
      .vpc.id}"
}
```

Task 2.1

After saving the new `outputs.tf`file, run a `terraform apply -auto-approve`to see our new output. The output now provides a full URL that contains the IP address of our public web server. We also have an additional string that provides information about our VPC as well.

```
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration
     and found no differences, so no changes
are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

hello-world = "Hello World"
public_url = "https://10.0.101.10:8080/index.html"
vpc_id = "vpc-058d23c9d5d2f70b5"
vpc_information = "Your demo_environment VPC has an ID of vpc-058
    d23c9d5d2f70b5"
```

Task 2.2

If you ever need to read the values of your Terraform outputs but might be nervous about running a `terraform apply`, you can use the `terraform output`command to view them.

Run a `terraform output`and view the list of outputs from your configuration:

```
$ terraform output
hello-world = "Hello World"
public_url = "https://10.0.101.10:8080/index.html"
vpc_id = "vpc-058d23c9d5d2f70b5"
vpc_information = "Your demo_environment VPC has an ID of vpc-058
    d23c9d5d2f70b5"
```

Task 3: Generate Machine-Readable Outputs in JSON

When working with Terraform in your organization, it is common practice to use an automation tool to automate your terraform deployments. But don't fret! We can still use outputs and make them "machine-readable" so other automation tools can parse the information, if needed.

Run a `terraform output -json`and view the list of outputs in JSON format from your configuration:

```
terraform output -json
{
  "hello-world": {
    "sensitive": false,
    "type": "string",
    "value": "Hello World"
  },
  "public_url": {
    "sensitive": false,
    "type": "string",
    "value": "https://10.0.101.10:8080/index.html"
  },
  "vpc_id": {
    "sensitive": false,
    "type": "string",
    "value": "vpc-058d23c9d5d2f70b5"
  },
  "vpc_information": {
    "sensitive": false,
    "type": "string",
    "value": "Your demo_environment VPC has an ID of vpc-058d23c9d5d2f70b5
       "
  }
}
```