

Terraform Key Concepts



Explain below concepts in terraform with examples -> Count and index, For_each, length, dynamic blocks, map, list, set.

Terraform Concepts with Examples

1. Count and Index

- The `count` meta-argument is used to create multiple instances of a resource.
- The `count.index` is used to reference the current iteration.

Example:

```
hcl

resource "aws_instance" "example" {
  count      = 3
  ami       = "ami-123456"
  instance_type = "t2.micro"

  tags = {
    Name = "Instance-${count.index}"
  }
}
```

👉 This creates **3 instances** with names `Instance-0`, `Instance-1`, and `Instance-2`.

2. For_each

- `for_each` is used for iterating over **maps** or **sets** (not lists).
- The current element is accessed using `each.key` and `each.value`.

Example:

```
hcl

resource "aws_instance" "example" {
  for_each = toset(["dev", "staging", "prod"])
  ami     = "ami-123456"
  instance_type = "t2.micro"

  tags = {
    Name = "Instance-${each.key}"
  }
}
```

👉 Creates **three instances** named `Instance-dev`, `Instance-staging`, and `Instance-prod`.

3. Length

- The `length` function returns the number of elements in a **list, map, or set**.

Example:

```
hcl

variable "names" {
  default = ["Alice", "Bob", "Charlie"]
}

output "total_names" {
  value = length(var.names) # Output: 3
}
```

4. Dynamic Blocks

- Used when defining **nested blocks dynamically** inside a resource.

Example:

```
hcl

resource "aws_security_group" "example" {
  name = "example_sg"

  dynamic "ingress" {
    for_each = [22, 80, 443] # Creating rules for these ports

    content {
      from_port = ingress.value
      to_port   = ingress.value
      protocol  = "tcp"
      cidr_blocks = ["0.0.0.0/0"]
    }
  }
}
```

👉 Creates **ingress rules** for ports 22, 80, and 443 dynamically.

5. Map

- A **map** is a key-value data structure.

Example:

```
hcl

variable "instance_types" {
  default = {
    dev    = "t2.micro"
    staging = "t2.small"
    prod   = "t2.large"
  }
}

output "prod_instance_type" {
  value = var.instance_types["prod"] # Output: t2.large
}
```

6. List

- A **list** is an ordered collection of values.

Example:

```
h

variable "names" {
  default = ["Alice", "Bob", "Charlie"]
}

output "second_name" {
  value = var.names[1] # Output: Bob
}
```

7. Set

- A **set** is an unordered collection of unique values.

Example:

```
hcl

variable "unique_names" {
  default = toset(["Alice", "Bob", "Charlie"])
}

output "name_exists" {
  value = contains(var.unique_names, "Alice") # Output: true
}
```

👉 Unlike a list, a set **does not allow duplicate values**.

Summary

Concept	Description
<code>`count`</code> & <code>`index`</code>	Used for creating multiple instances with an index.
<code>`for_each`</code>	Iterates over maps or sets.
<code>`length`</code>	Returns the number of elements in a list, map, or set.
<code>`dynamic`</code>	Creates nested blocks dynamically.
<code>`map`</code>	A key-value collection.
<code>`list`</code>	An ordered collection of values.
<code>`set`</code>	An unordered, unique collection of values.

Let me know if you need further explanation! 🚀