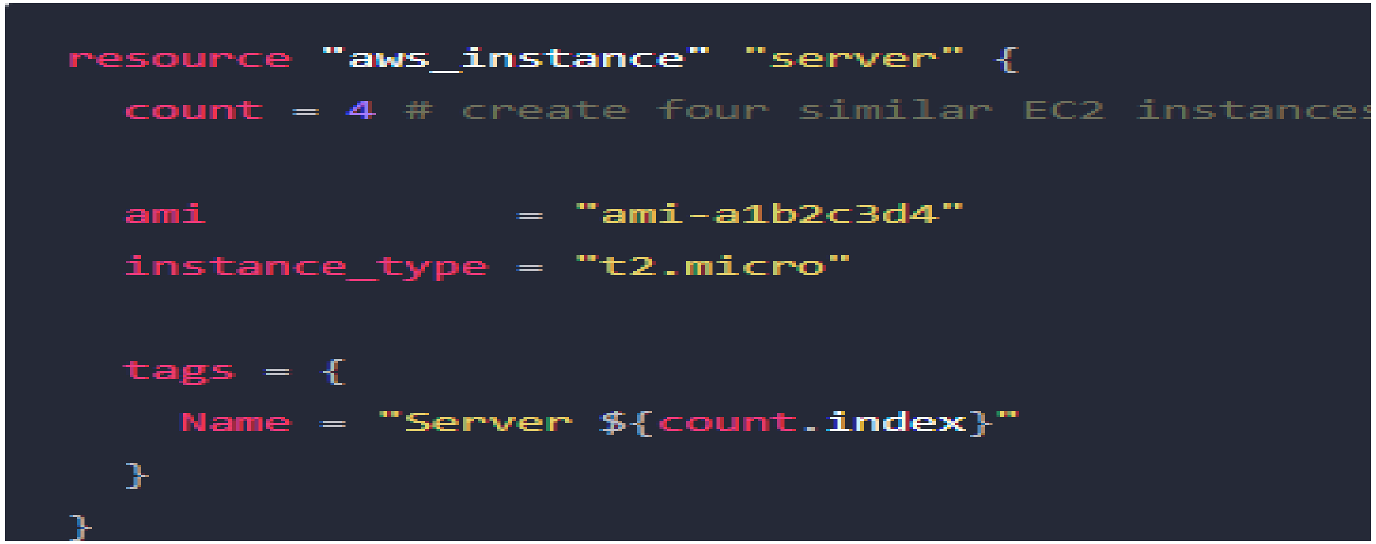
The count Meta-Argument

OW

Basic Syntax

- count is a meta-argument defined by the Terraform language. It can be used with modules and with every resource type.
- The count meta-argument accepts a whole number, and creates that many instances of the resource or module.



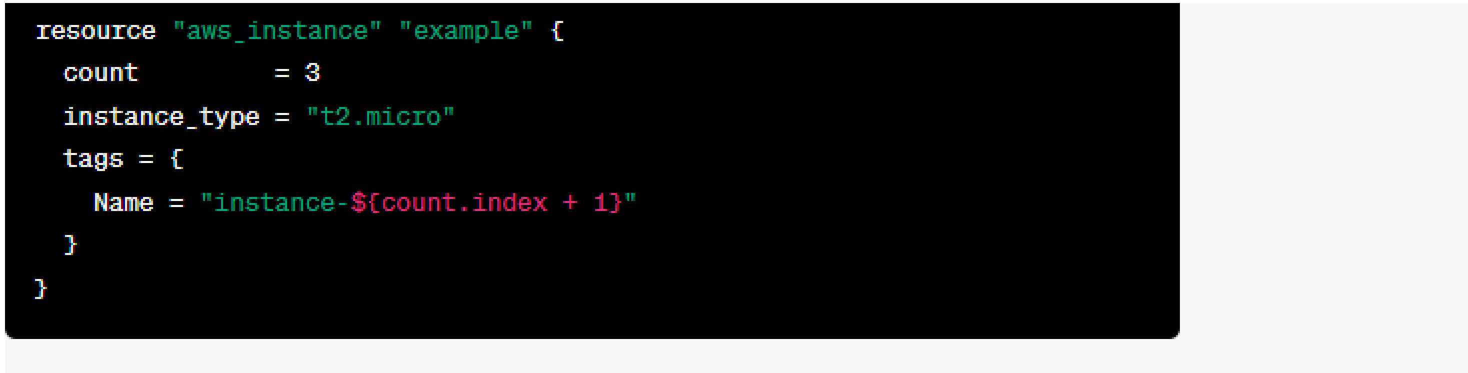
```
resource "aws_instance" "server" {
  count = 4 # create four similar EC2 instances

  ami           = "ami-a1b2c3d4"
  instance_type = "t2.micro"

  tags = {
    Name = "Server ${count.index}"
  }
}
```

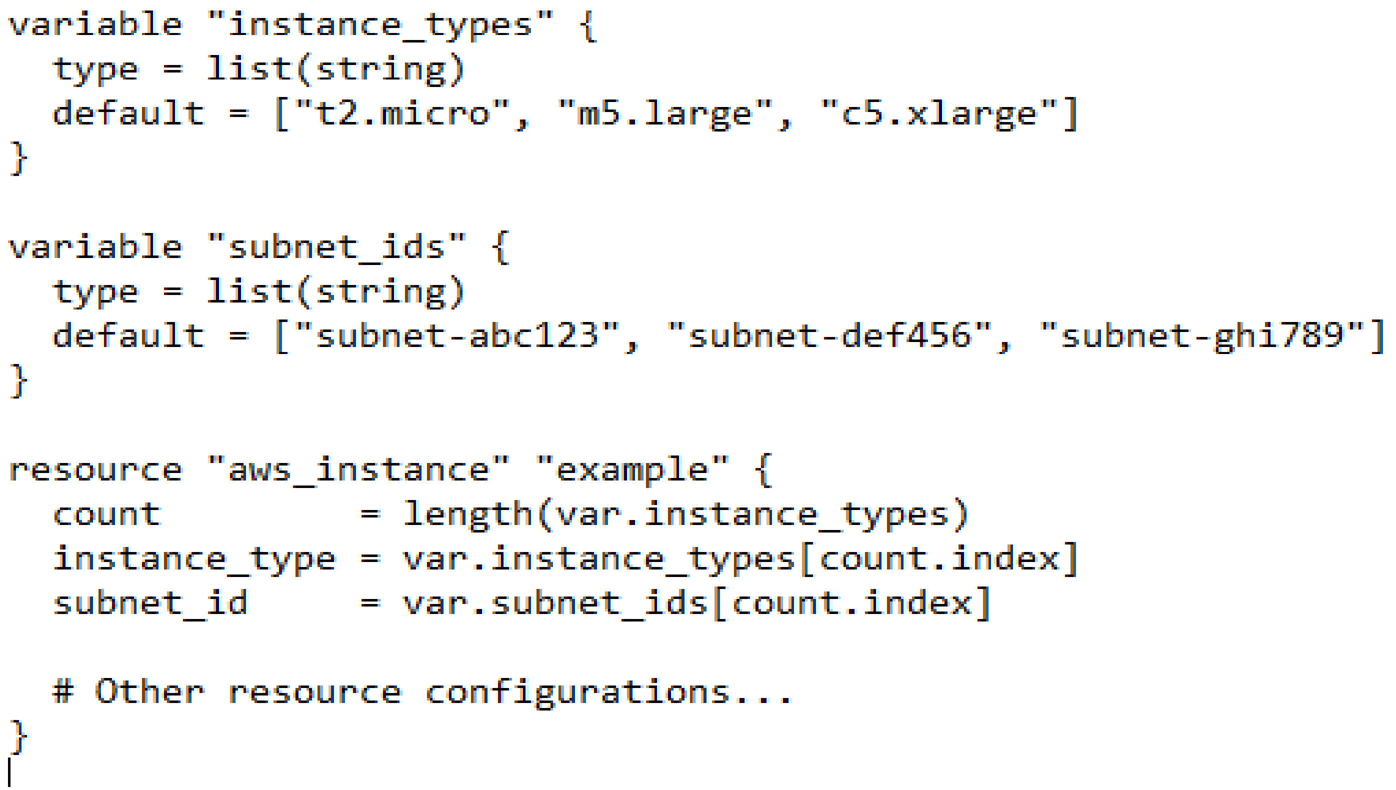
Count.index

- In blocks where count is set, an additional count object is available in expressions, so you can modify the configuration of each instance. This object has one attribute:
  - count.index — The distinct index number (starting with 0) corresponding to this instance.
  - count.index expression represents the current index of the resource being created.



```
resource "aws_instance" "example" {
  count         = 3
  instance_type = "t2.micro"
  tags = {
    Name = "instance-${count.index + 1}"
  }
}
```

..



```
variable "instance_types" {
  type = list(string)
  default = ["t2.micro", "m5.large", "c5.xlarge"]
}

variable "subnet_ids" {
  type = list(string)
  default = ["subnet-abc123", "subnet-def456", "subnet-ghi789"]
}

resource "aws_instance" "example" {
  count         = length(var.instance_types)
  instance_type = var.instance_types[count.index]
  subnet_id     = var.subnet_ids[count.index]

  # Other resource configurations...
}
```

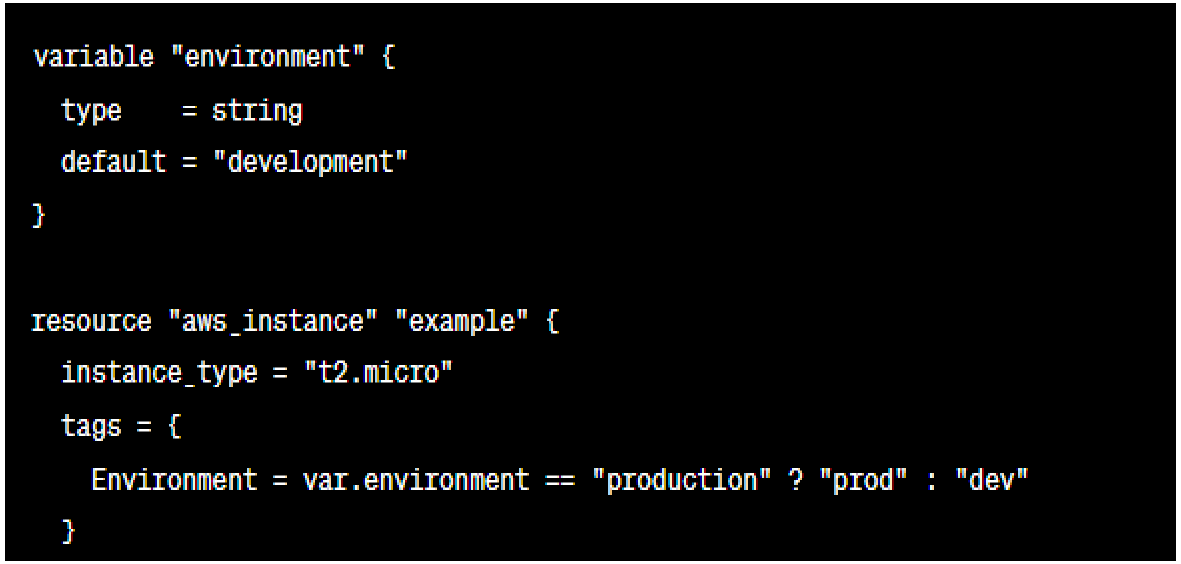
Conditional expression

- In Terraform, conditional expressions allow you to create conditional logic within your configurations. You can use conditional expressions to make decisions, set values, or control the flow of your Terraform code based on certain conditions. Here's an overview of conditional expressions in Terraform:

  Conditional Operator (condition ? true_val : false_val):

- The conditional operator ? : allows you to evaluate a condition and return different values based on whether the condition is true or false.

..



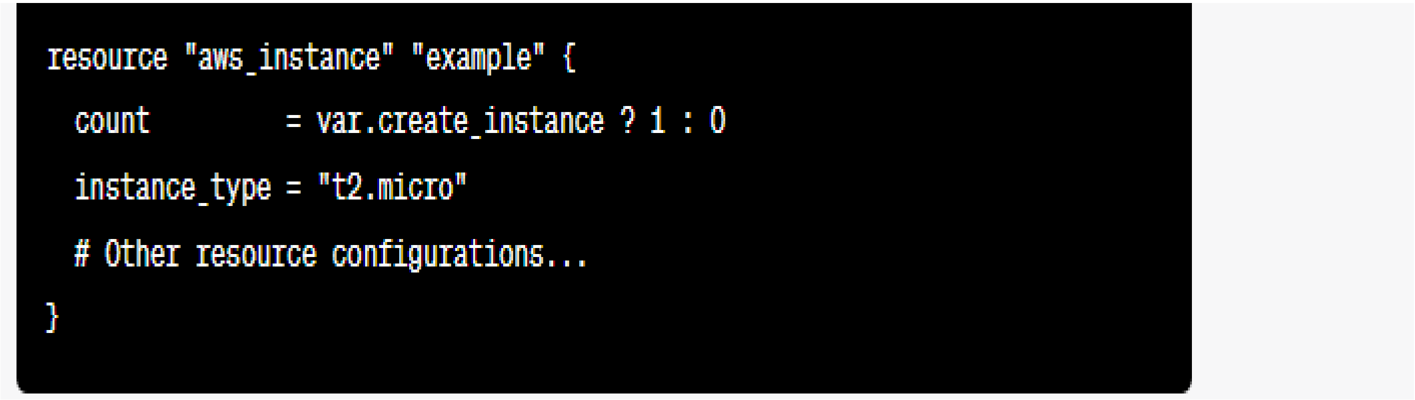
```
variable "environment" {
  type    = string
  default = "development"
}

resource "aws_instance" "example" {
  instance_type = "t2.micro"
  tags = {
    Environment = var.environment == "production" ? "prod" : "dev"
  }
```

In this example, the Environment tag value will be set to "prod" if the environment variable is equal to "production", and "dev" otherwise.

count with Conditional Expression:

- In this example, the aws_instance resource will be created only if the create_instance variable is set to true. If the variable is false, no instances will be created



```
resource "aws_instance" "example" {
  count         = var.create_instance ? 1 : 0
  instance_type = "t2.micro"
  # Other resource configurations...
}
```

- In this example, the aws_instance resource will be created only if the create_instance variable is set to true. If the variable is false, no instances will be created.