

# The COUNT Meta-Argument

# General Approach

- Let's assume, you need to create two Public IPs. One of the common approach is to define two separate resource blocks for `azurerm_public_ip`.

```
resource "azurerm_resource_group" "example" {  
  name      = "devoprg"  
  location  = "eastus"  
}  
  
resource "azurerm_public_ip" "first_ip" {  
  name                  = "first_ip"  
  resource_group_name   = azurerm_resource_group.example.name  
  location              = azurerm_resource_group.example.location  
  allocation_method     = "Static"  
}  
  
resource "azurerm_public_ip" "second_ip" {  
  name                  = "second_ip"  
  resource_group_name   = azurerm_resource_group.example.name  
  location              = azurerm_resource_group.example.location  
  allocation_method     = "Static"  
}
```

# Overview of Count Parameter

- With count parameter, we can simply specify the count value and the resource can be scaled accordingly.

```
resource "azurerm_resource_group" "example" {  
  name      = "devoprg"  
  location = "eastus"  
}  
  
resource "azurerm_public_ip" "example" {  
  name                        = "myip"  
  resource_group_name        = azurerm_resource_group.example.name  
  location                   = azurerm_resource_group.example.location  
  allocation_method          = "Static"  
  count = 2  
}
```

# Challenges

- With the below code, terraform will create Public Ips, but the problem here is, it will give the same name to all IPs then error will come.

```
resource "azurerm_resource_group" "example" {  
  name      = "devoprg"  
  location = "eastus"  
}  
  
resource "azurerm_public_ip" "example" {  
  name                        = "myip"  
  resource_group_name        = azurerm_resource_group.example.name  
  location                   = azurerm_resource_group.example.location  
  allocation_method          = "Static"  
  count = 2  
}
```

# Count Index

```
resource "azurerm_resource_group" "example" {  
  name      = "devoprg"  
  location = "eastus"  
}  
  
resource "azurerm_public_ip" "example" {  
  name                = "myip${count.index}"  
  resource_group_name = azurerm_resource_group.example.name  
  location            = azurerm_resource_group.example.location  
  allocation_method   = "Static"  
  count = 2  
}
```

# Challenges with Default Count Index

Having a ipname like mypip0, mypip1 might not always be suitable.

Better names like dev-pip, stage-pip, prod-pip is better. count.index can help in such scenario as well.

```
variable "pip_names" {  
  type = list  
  default = ["dev-pip", "stage-pip", "prod-pip"]  
}
```

# Solution

```
resource "azurerm_resource_group" "example" {
  name      = "devoprg"
  location  = "eastus"
}

resource "azurerm_public_ip" "example" {
  name                        = var.pip_names[count.index]
  resource_group_name        = azurerm_resource_group.example.name
  location                   = azurerm_resource_group.example.location
  allocation_method          = "Static"
  count = 3
}

variable "pip_names" {
  type = list
  default = ["dev_pip", "qa_pip", "prod_pip"]
}
```

# Join us in our Adventure



<https://www.linkedin.com/in/akash-kumar-480b3858/>



[https://www.instagram.com/akash\\_sinha08/](https://www.instagram.com/akash_sinha08/)