VPC Subnet Module Integration

I want o create vpc and private public subnet using modules. How can we use vpc_id from vpc module into subnet module

When creating a **VPC and subnets using Terraform modules**, you need to **output the VPC ID from the VPC module** and **use it as an input in the subnet module**. Here's how you can do it:

Step 1: Create a VPC Module (`modules/vpc/main.tf`)

This module will create the VPC and output the `vpc_id`.

```
hcl

# modules/vpc/main.tf
resource "aws_vpc" "main" {
  cidr_block = var.cidr_block
  enable_dns_support = true
  enable_dns_hostnames = true

  tags = {
    Name = var.vpc_name
  }
}

output "vpc_id" {
  value = aws_vpc.main.id
}
```

Variables for VPC module (`modules/vpc/variables.tf`):

```
hcl

variable "cidr_block" {
  description = "CIDR block for the VPC"
  type        = string
}

variable "vpc_name" {
  description = "Name of the VPC"
  type        = string
}
```

Step 2: Create a Subnet Module (`modules/subnet/main.tf`)

This module will create public and private subnets using the `vpc_id` from the VPC module.

```
h

# modules/subnet/main.tf
resource "aws_subnet" "public" {
  vpc_id                  = var.vpc_id
  cidr_block              = var.public_subnet_cidr
```

```
  map_public_ip_on_launch = true
  availability_zone       = var.availability_zone

  tags = {
    Name = "Public Subnet"
  }
}

resource "aws_subnet" "private" {
  vpc_id            = var.vpc_id
  cidr_block        = var.private_subnet_cidr
  availability_zone = var.availability_zone

  tags = {
    Name = "Private Subnet"
  }
}

output "public_subnet_id" {
  value = aws_subnet.public.id
}

output "private_subnet_id" {
  value = aws_subnet.private.id
}
```

Variables for Subnet module (`modules/subnet/variables.tf`):

```
hcl


variable "vpc_id" {
  description = "VPC ID where the subnets will be created"
  type        = string
}

variable "public_subnet_cidr" {
  description = "CIDR block for the public subnet"
  type        = string
}

variable "private_subnet_cidr" {
  description = "CIDR block for the private subnet"
  type        = string
}

variable "availability_zone" {
  description = "Availability Zone for the subnets"
  type        = string
}
```

Step 3: Call the Modules from `main.tf`

Now, in your **root Terraform file (`main.tf`)**, you need to reference these modules.

```
hcl


# main.tf
module "vpc" {
  source     = "./modules/vpc"
  cidr_block = "10.0.0.0/16"
  vpc_name   = "MyVPC"
}

module "subnet" {
  source              = "./modules/subnet"
  vpc_id              = module.vpc.vpc_id  # Using VPC ID from VPC module
  public_subnet_cidr  = "10.0.1.0/24"
```

```
  private_subnet_cidr = "10.0.2.0/24"
  availability_zone   = "us-east-1a"
}
```

Step 4: Run Terraform

1. Initialize Terraform:

```
sh

terraform init
```

2. Plan the deployment:

```
sh

terraform plan
```

3. Apply the changes:

```
sh

terraform apply -auto-approve
```

Summary

- The **VPC module** creates the VPC and **outputs the `vpc_id`**.
- The **Subnet module** takes the `vpc_id` as an input to create public and private subnets.
- The **root `main.tf` file** calls both modules and passes the **VPC ID** dynamically.

This modular approach ensures **reusability and maintainability**. 🚀 Let me know if you need further clarifications!