



# CloudFormation Part 2



**Brock Tubre**  
TECHNICAL INSTRUCTOR

# Template Policies

```
"MyWaitCondition" : {
  "Type" : "AWS::CloudFormation::WaitCondition",
  "CreationPolicy" : {
    "ResourceSignal" : {
      "Timeout" : "PT30M",
      "Count" : "10"
    }
  }
}
```

## CreationPolicy

Prevents resources from being creation until dependent subject have also been created.

```
"Resources" : {
  "LnRDSMySQLDataBase" : {
    "Type" : "AWS::RDS::DBInstance",
    "DeletionPolicy" : "Snapshot"
  }
}
```

## DeletionPolicy

Determines what to retain when the CloudFormation stack is terminated.

```
"UpdatePolicy" : {
  "AutoScalingReplacingUpdate" : {
    "WillReplace" : true
  }
}
```

## UpdatePolicy and UpdateReplacePolicy

Determines what to do with existing resources if a template is updated and is set to overwrite the resources.

# DependsOn Attribute

## DependsOn

Allows us to control the order resources are deployed. You can specify that the creation of a specific resource follows another. When you add a DependsOn attribute to a resource, that resource is created only after the creation of the resource specified in the DependsOn attribute.

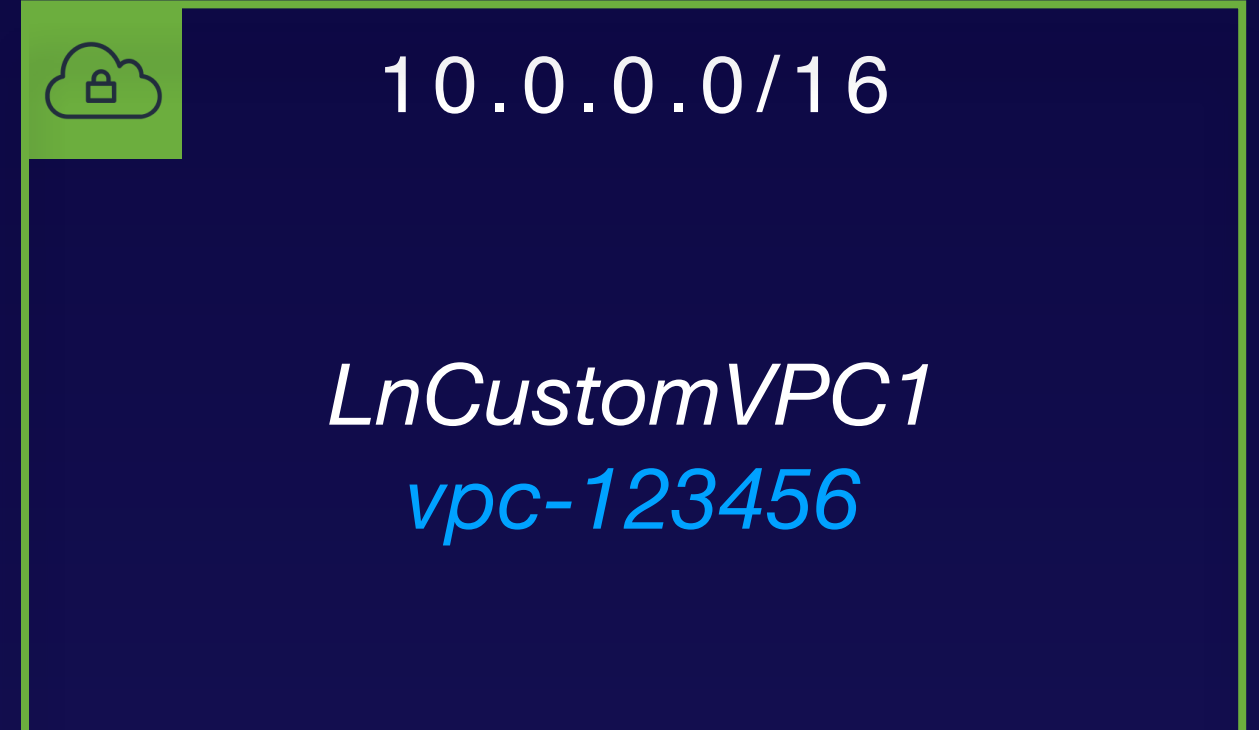


CloudFormation performs all resource deployments in parallel.

```
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId:
        Fn::FindInMap:
          - RegionMap
          - Ref: AWS::Region
          - AMI
      DependsOn: MyMySQLDB
  MyMySQLDB:
    Type: AWS::RDS::DBInstance
    Properties:
      AllocatedStorage: '5'
      DBInstanceClass: db.m1.small
      Engine: MySQL
      EngineVersion: '5.5'
      MasterUsername: MyName
      MasterUserPassword: MyPassword
```

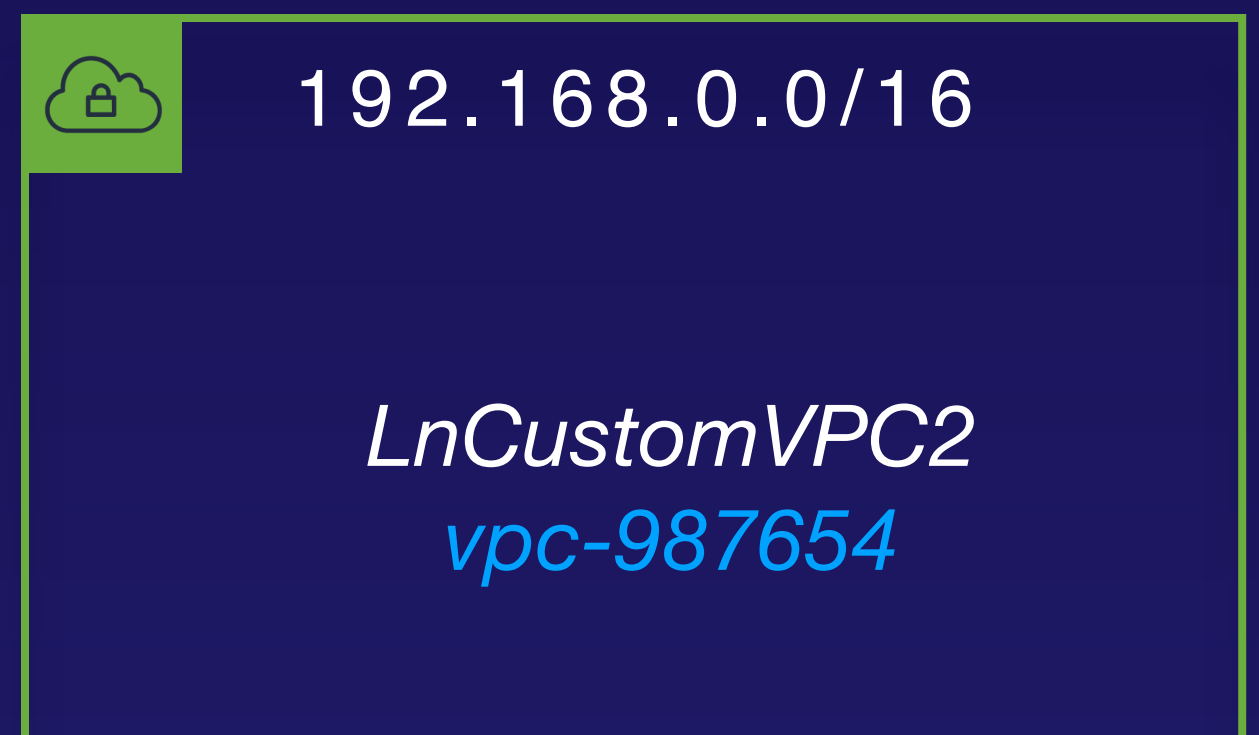
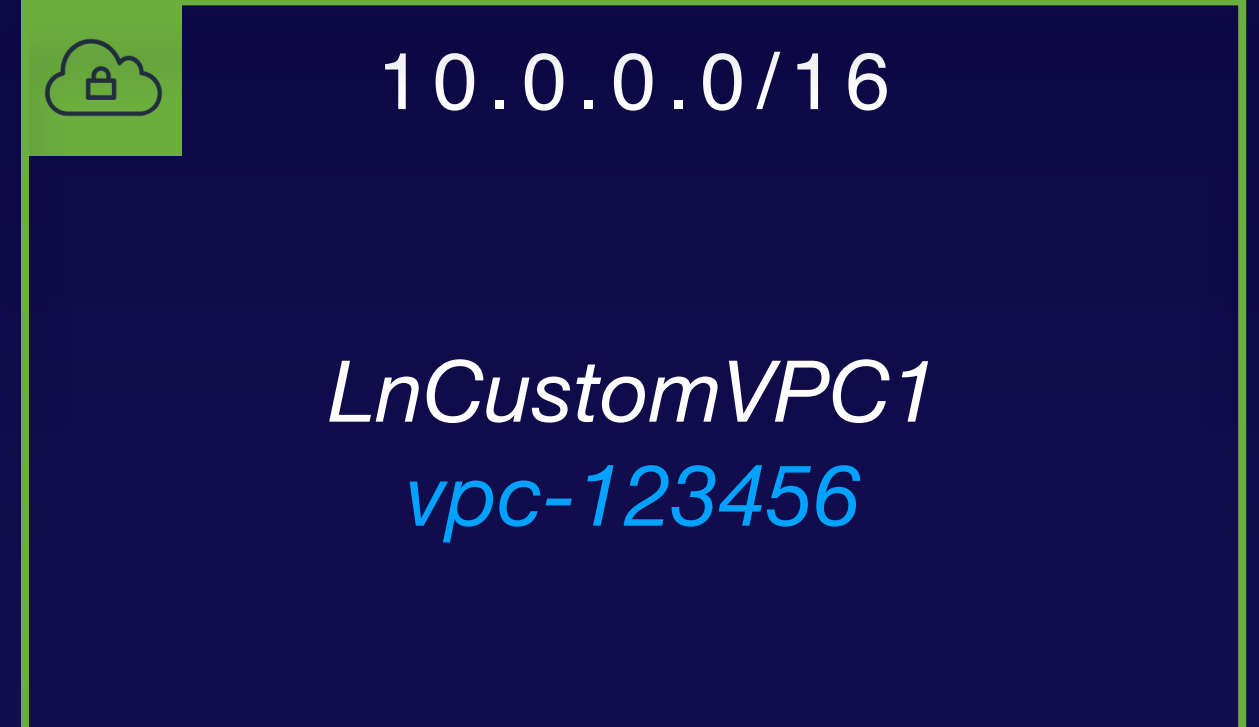
# VPC Peering In CloudFormation - Same Account

```
Resources:  
  LnCustomVPC1:  
    Type: AWS::EC2::VPC  
    Properties:  
      CidrBlock: '10.0.0.0/16'
```



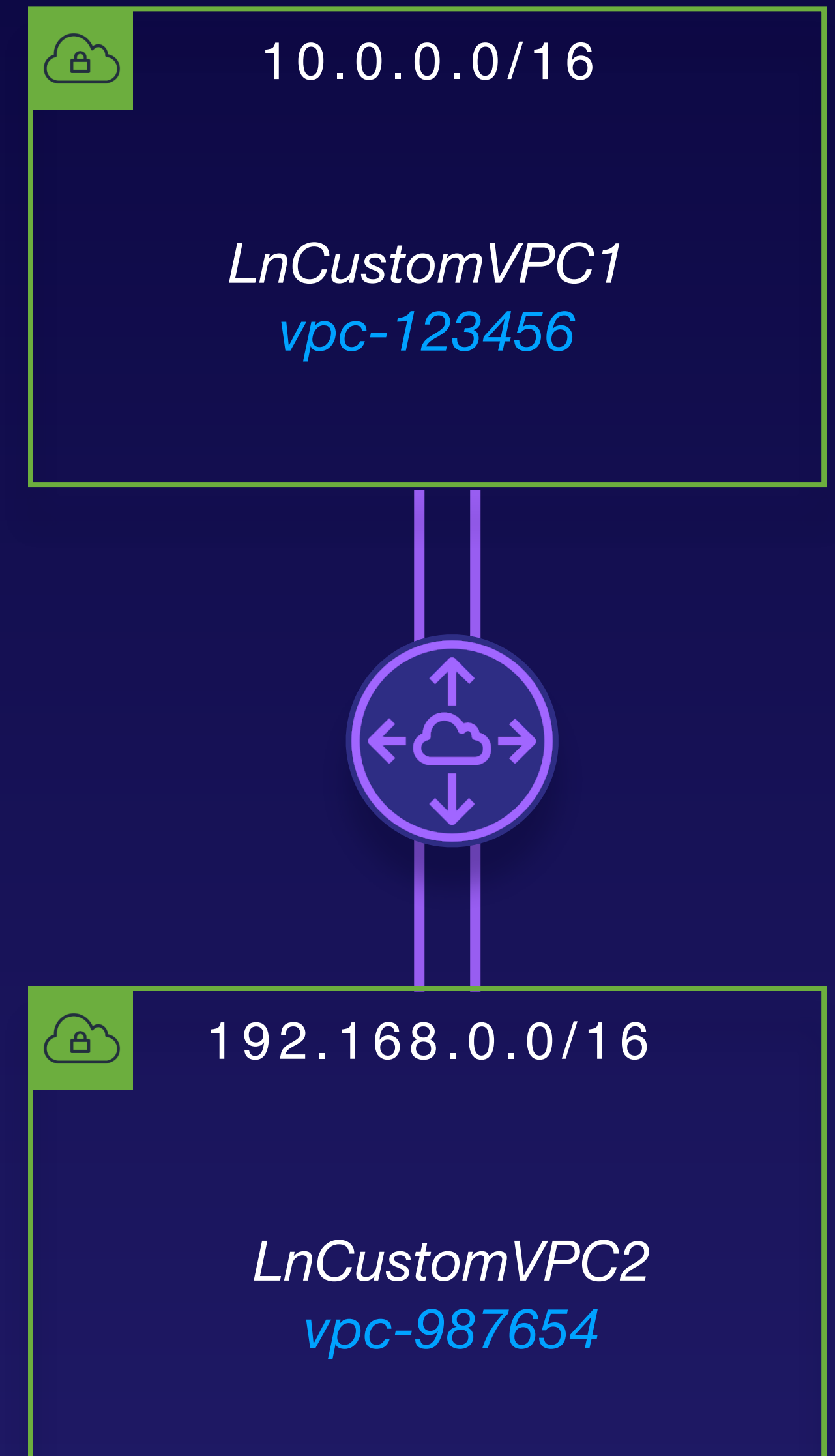
# VPC Peering In CloudFormation - Same Account

```
Resources:
  LnCustomVPC1:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
  LnCustomVPC2:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '192.168.0.0/16'
```



# VPC Peering In CloudFormation - Same Account

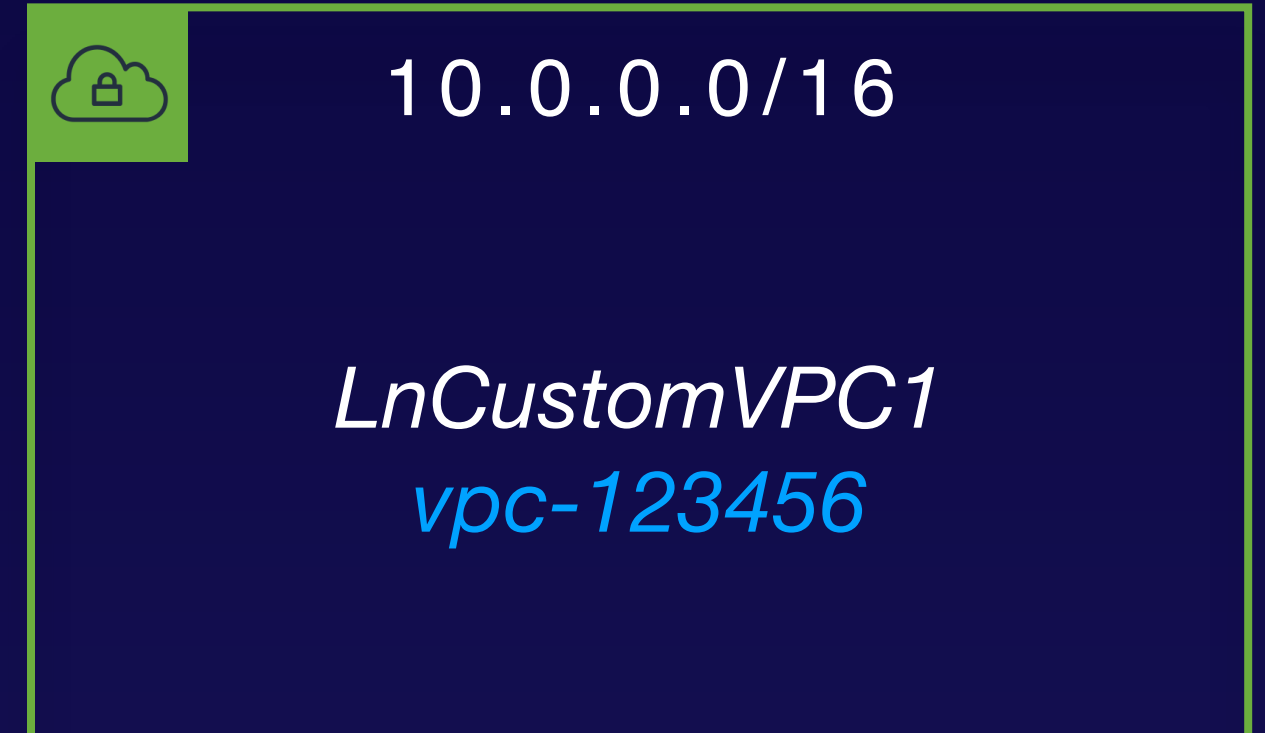
```
Resources:
  LnCustomVPC1:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
  LnCustomVPC2:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '192.168.0.0/16'
  LnPeeringConnection:
    Type: AWS::EC2::VPCPeeringConnection
    Properties:
      VpcId: !Ref LnCustomVPC1 ← vpc-123456
      PeerVpcId: !Ref LnCustomVPC2 ← vpc-987654
```



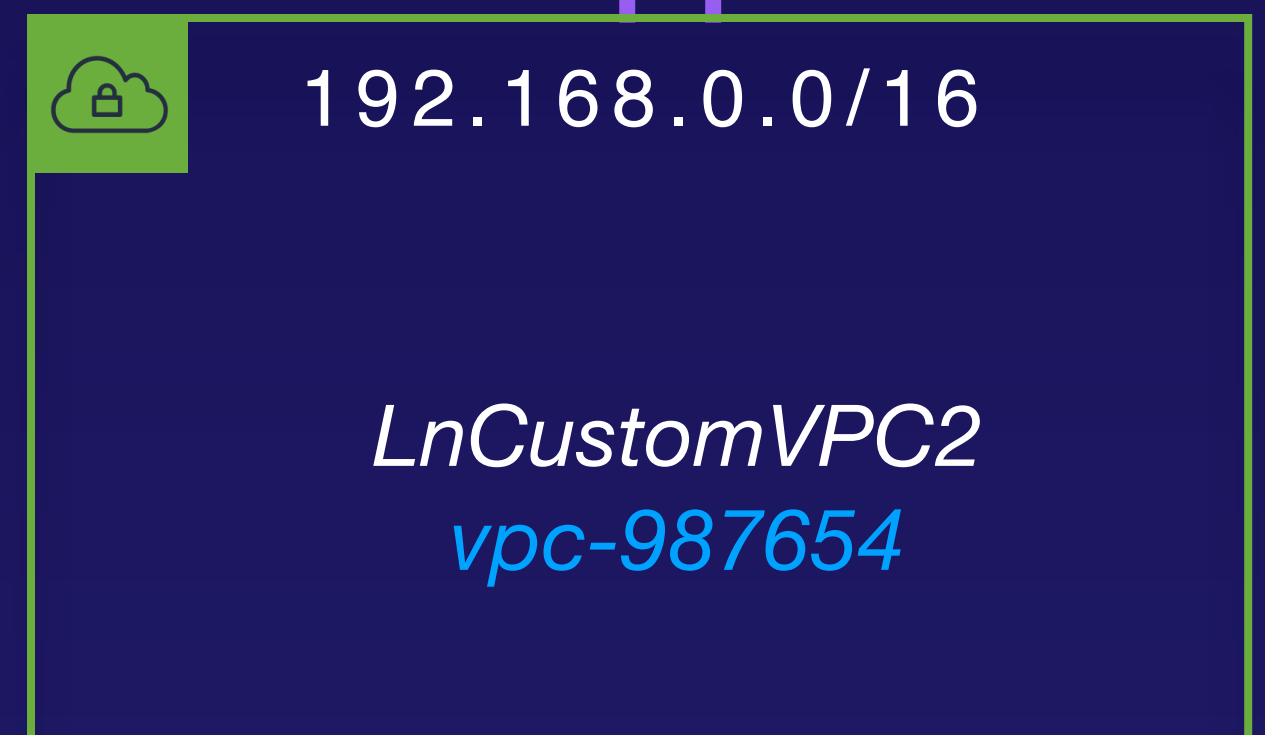
# VPC Peering In CloudFormation - Same Account

```
Resources:
  LnCustomVPC1:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
  LnCustomVPC2:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '192.168.0.0/16'
  LnPeeringConnection:
    Type: AWS::EC2::VPCPeeringConnection
    Properties:
      VpcId: !Ref LnCustomVPC1
      PeerVpcId: !Ref LnCustomVPC2
```

us-east-1



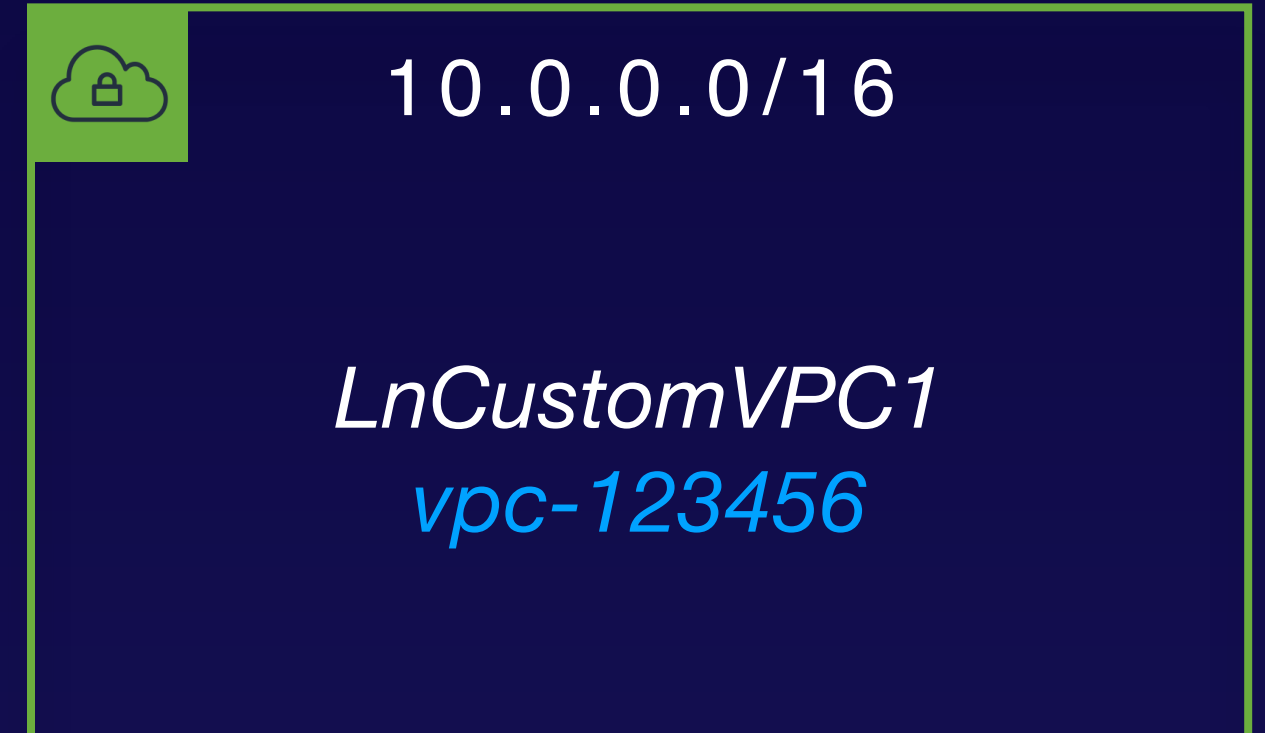
us-west-2



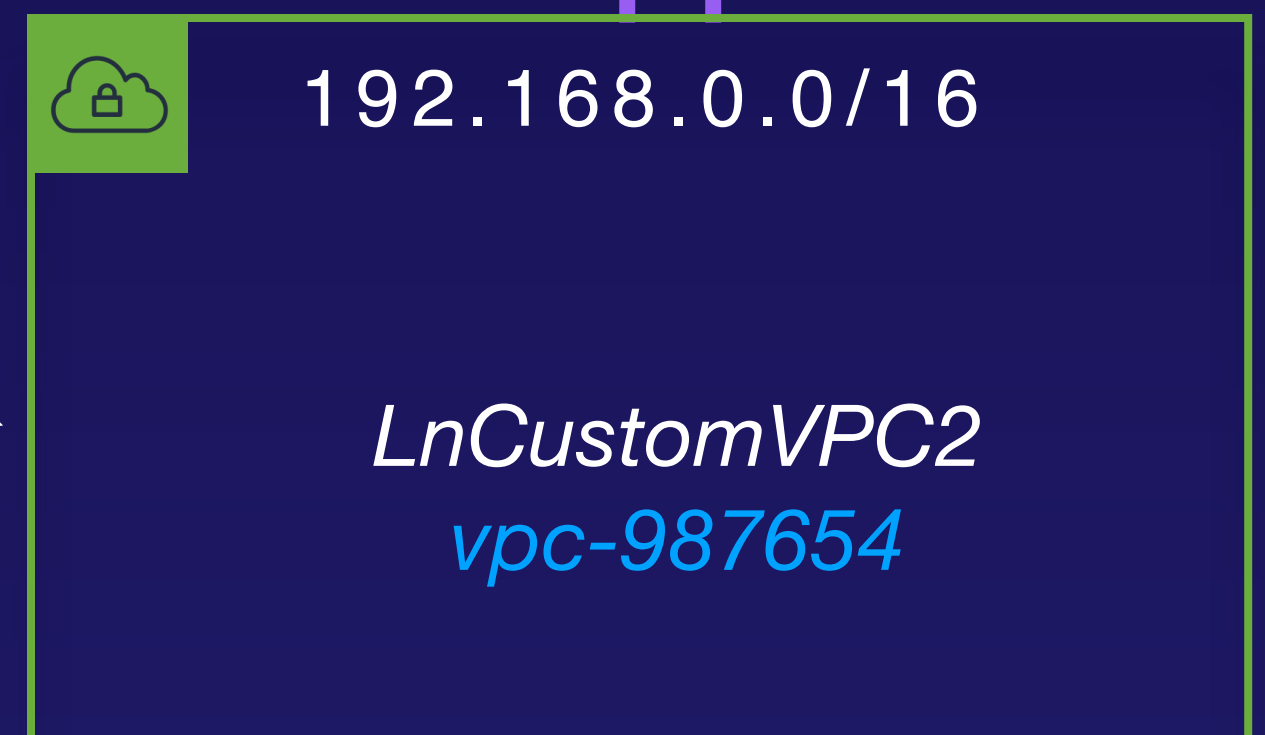
# VPC Peering In CloudFormation - Same Account

```
Resources:
  LnCustomVPC1:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
  LnCustomVPC2:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '192.168.0.0/16'
  LnPeeringConnection:
    Type: AWS::EC2::VPCPeeringConnection
    Properties:
      VpcId: !Ref LnCustomVPC1
      PeerVpcId: !Ref LnCustomVPC2
      PeerRegion: us-west-2
```

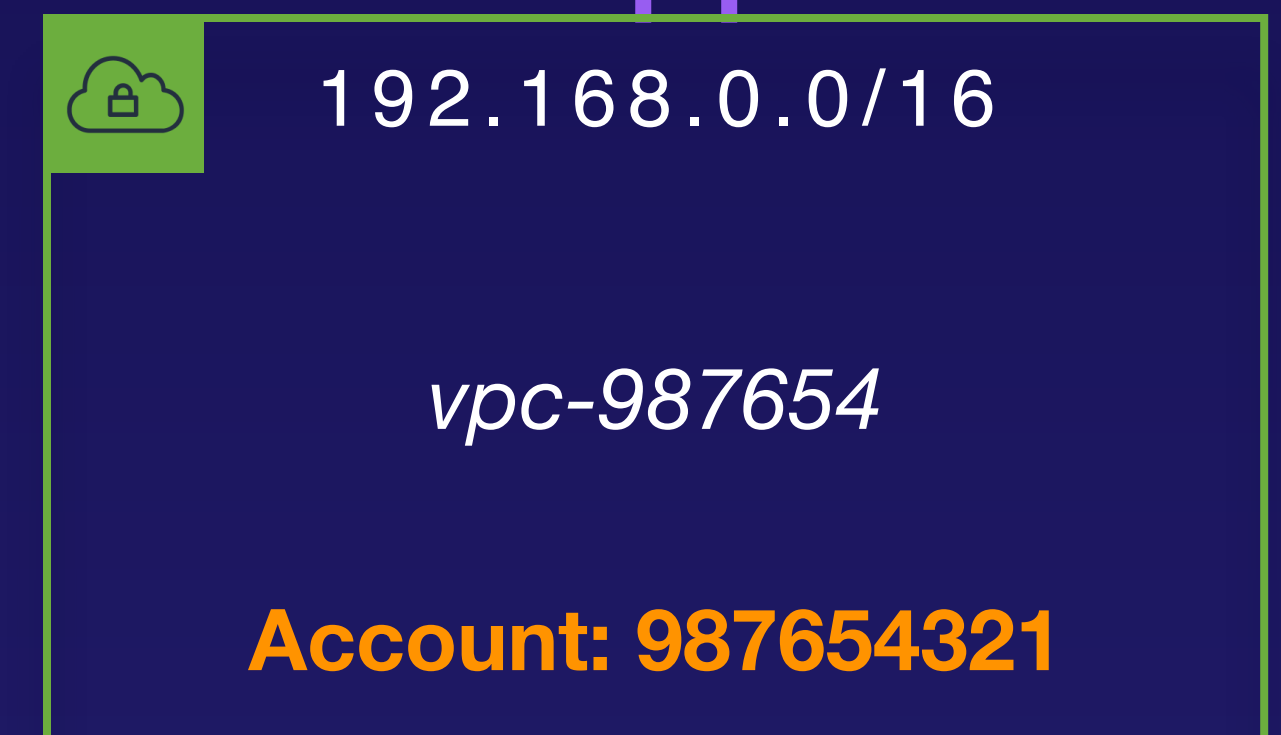
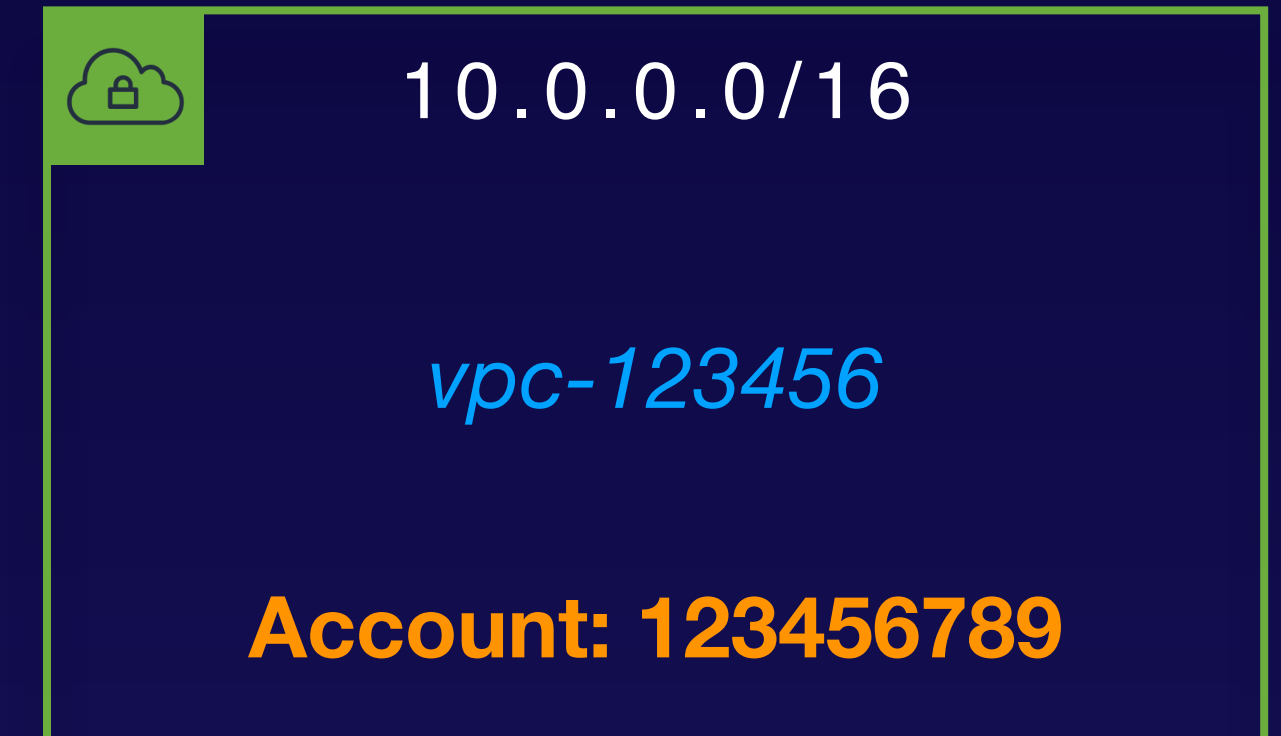
us-east-1



us-west-2



# VPC Peering In CloudFormation - Different Accounts



Create a role allowing account 123456789 cross account VPC peering.



`arn:aws:iam::987654321:role/allow-peer-requestor-role`

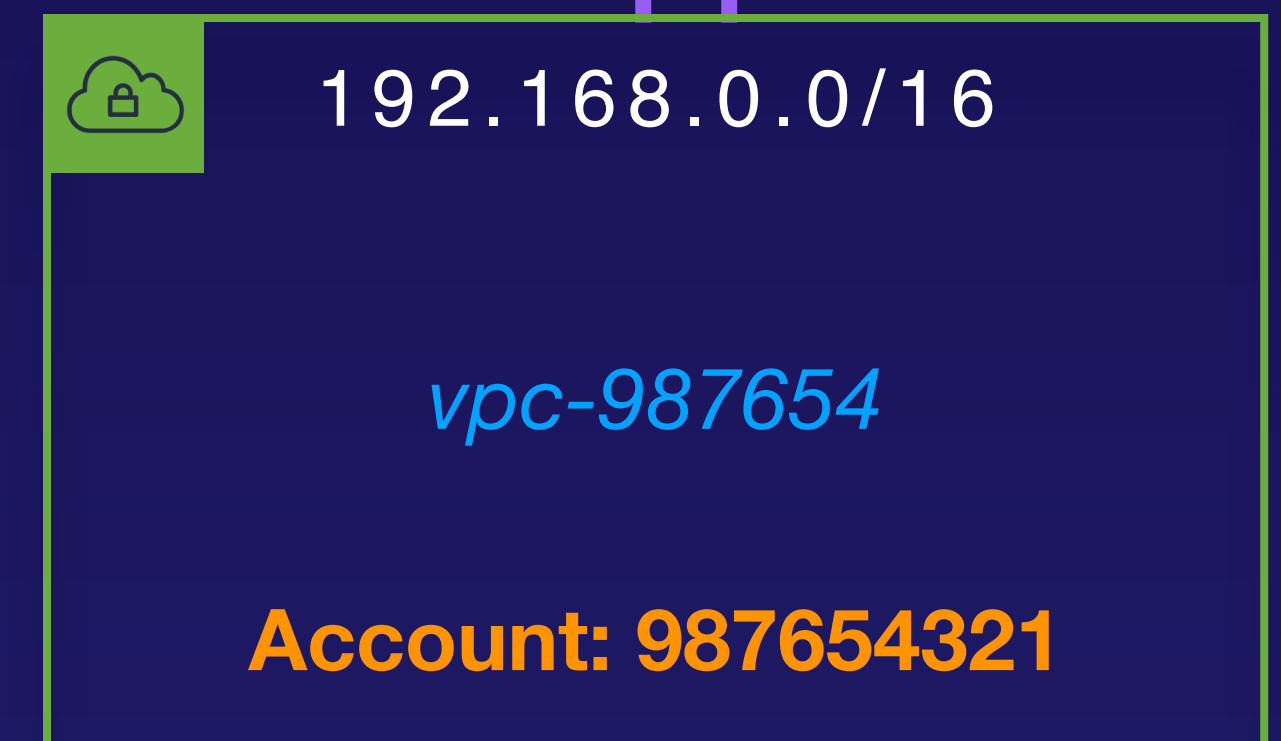
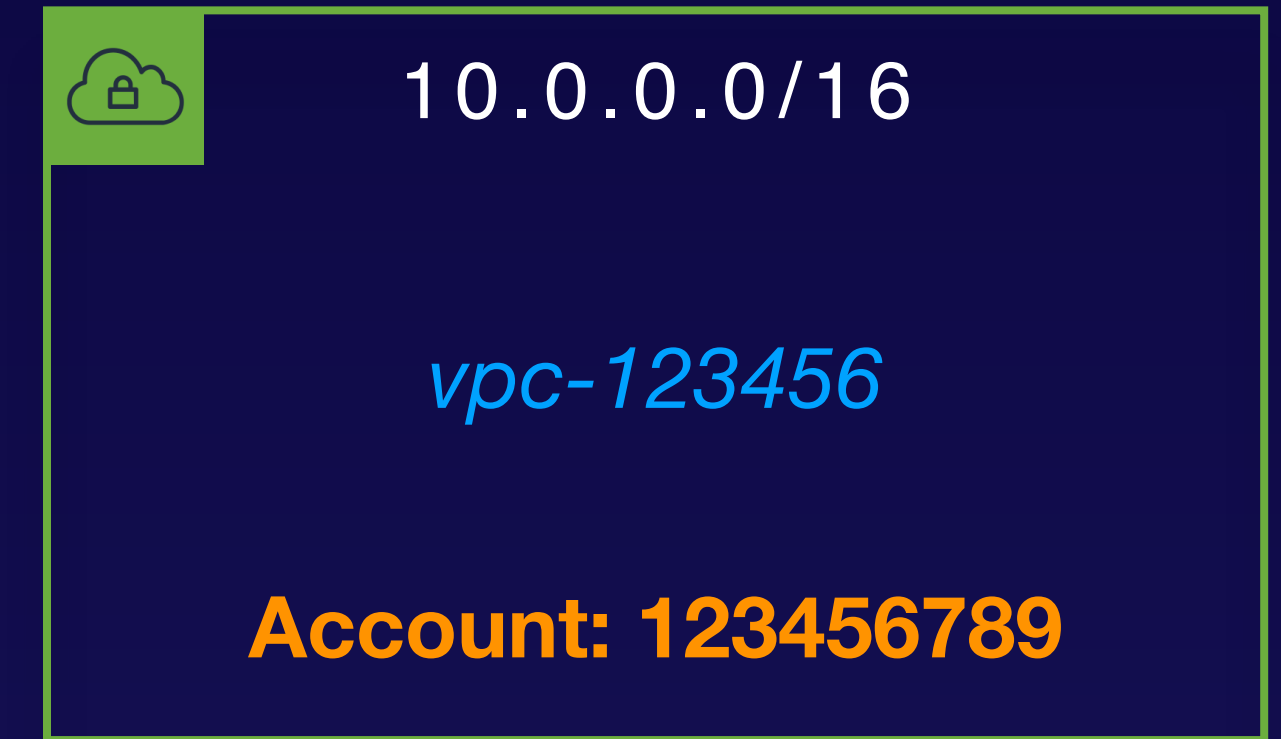


# VPC Peering In CloudFormation - Different Accounts

Resources:

```

...
LnPeeringConnection:
  Type: AWS::EC2::VPCPeeringConnection
    
```



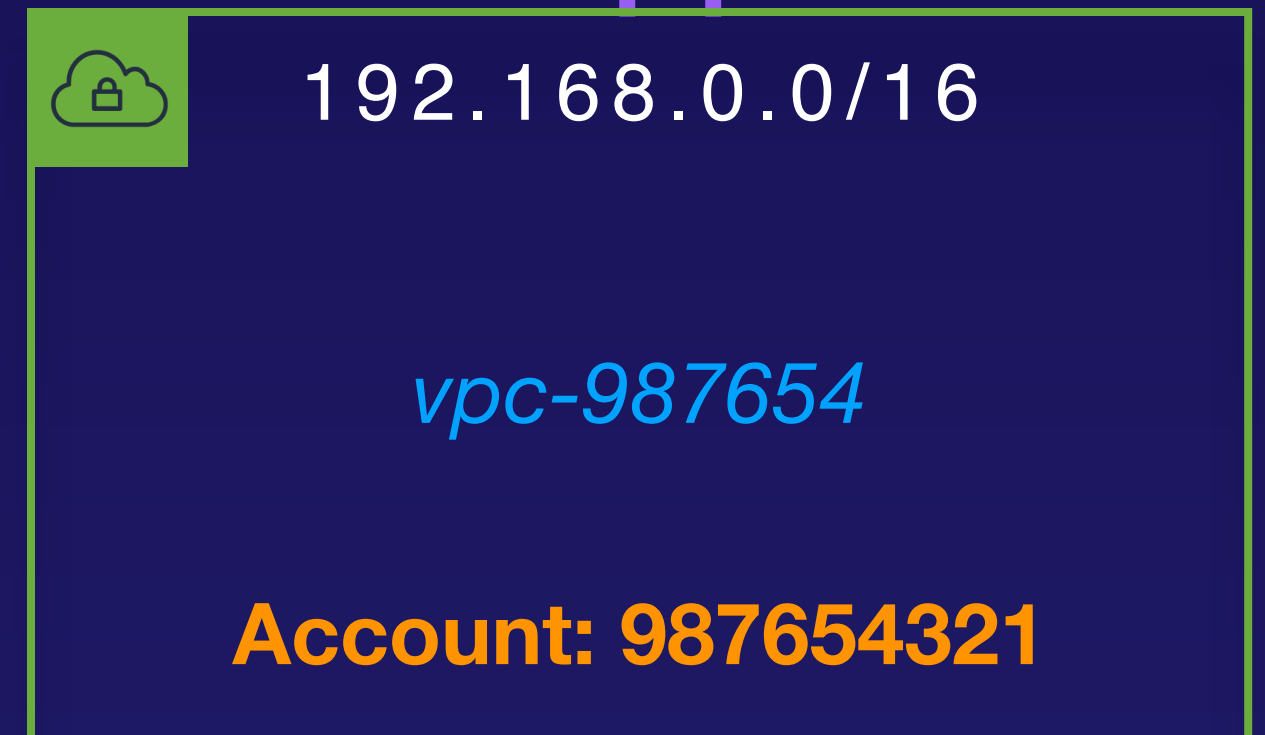
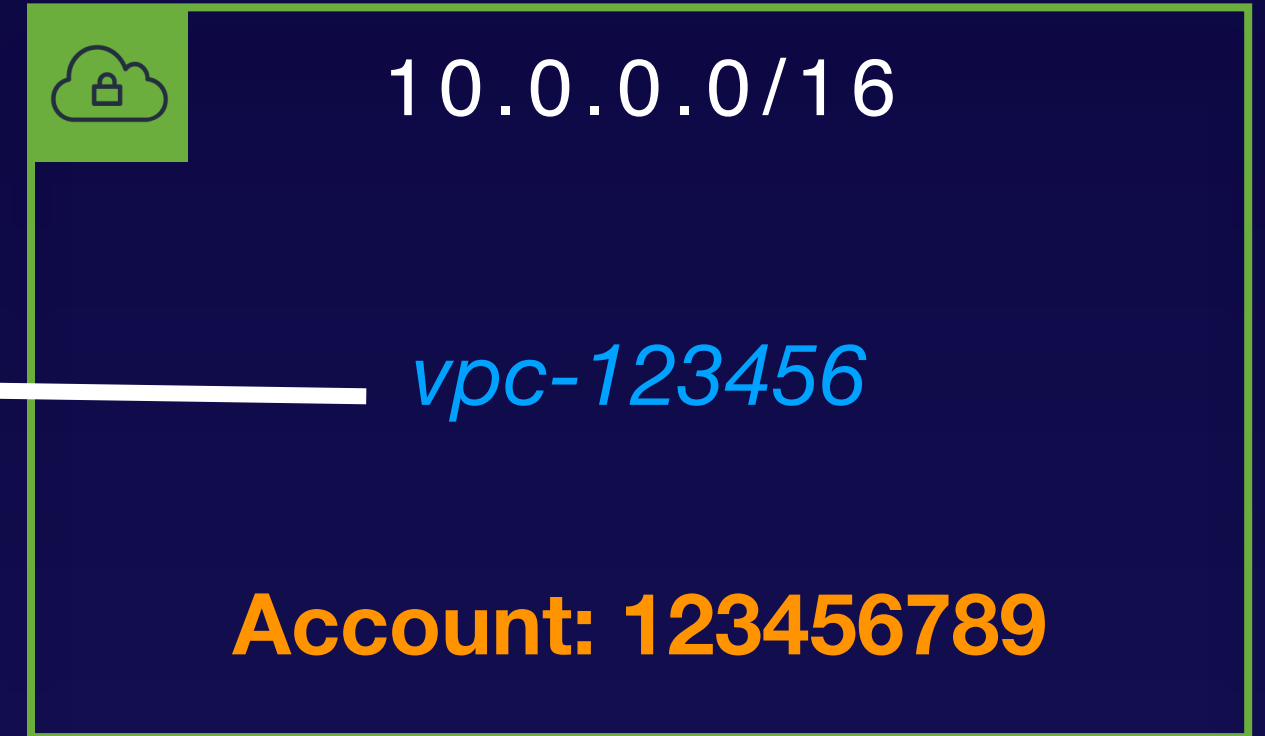
arn:aws:iam::987654321:role/allow-peer-requestor-role ←

# VPC Peering In CloudFormation - Different Accounts

Resources:

```

...
LnPeeringConnection:
  Type: AWS::EC2::VPCPeeringConnection
  Properties:
    VpcId: vpc-123456
    
```



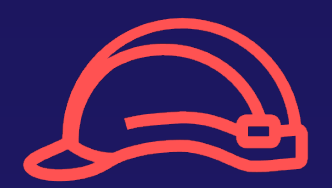
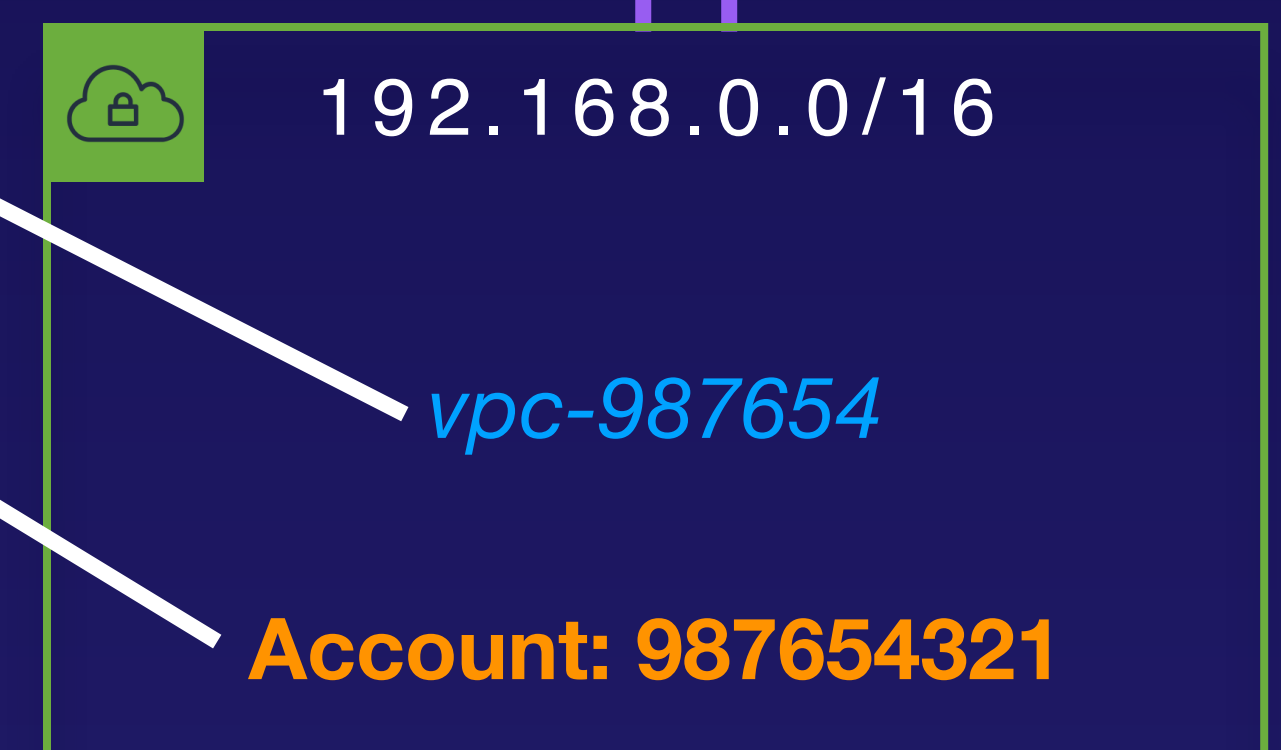
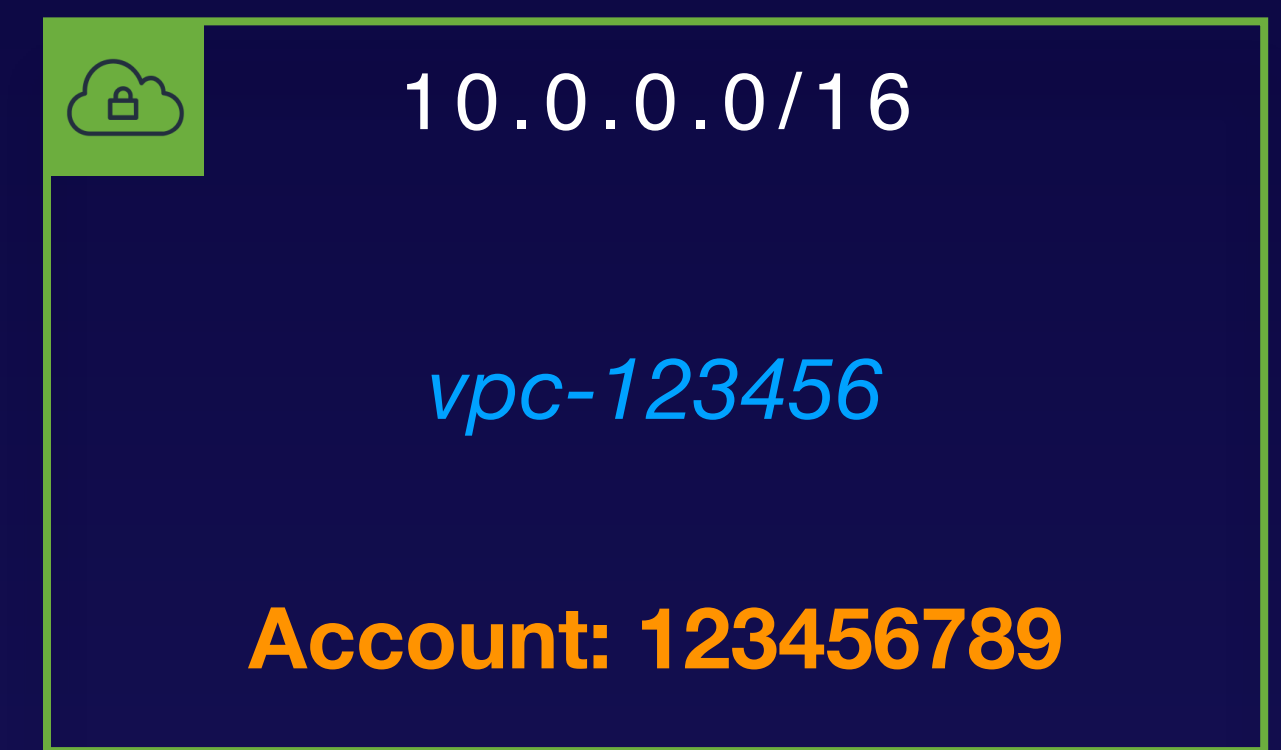
arn:aws:iam::987654321:role/allow-peer-requestor-role

# VPC Peering In CloudFormation - Different Accounts

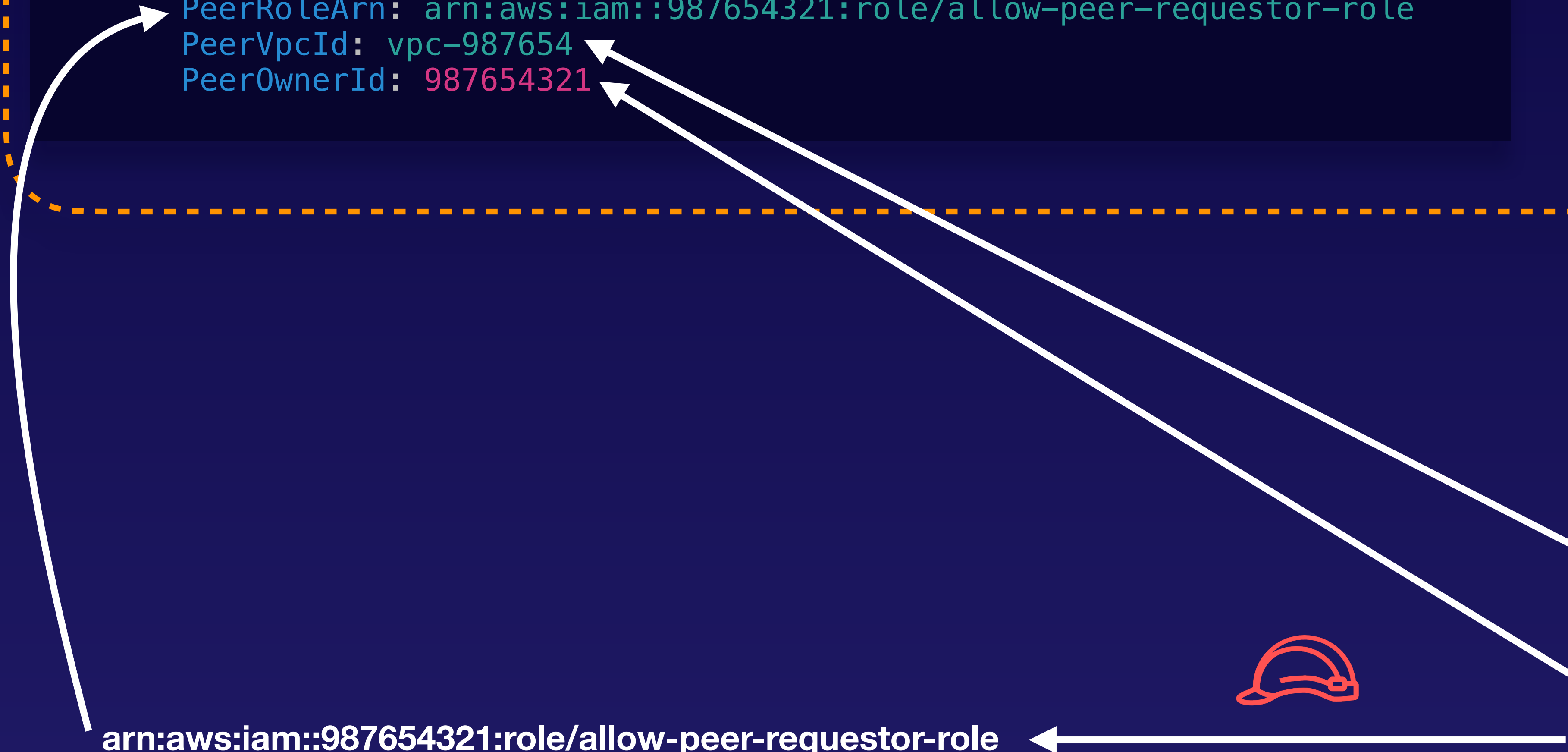
Resources:

```

...
LnPeeringConnection:
  Type: AWS::EC2::VPCPeeringConnection
  Properties:
    VpcId: vpc-123456
    PeerRoleArn: arn:aws:iam::987654321:role/allow-peer-requestor-role
    PeerVpcId: vpc-987654
    PeerOwnerId: 987654321
    
```



arn:aws:iam::987654321:role/allow-peer-requestor-role



Make sure you understand how the DependsOn attribute works and when to use it.

---

Understand how to peer VPCs with CloudFormation.