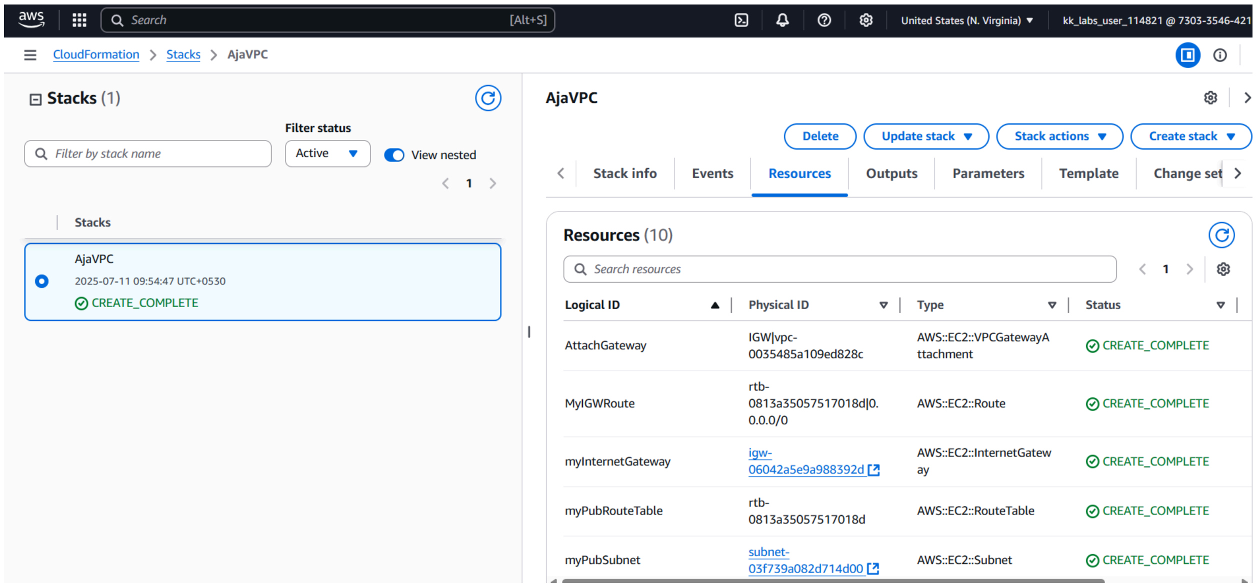
🚀 Successfully Deployed Multi-VPC AWS Infrastructure with CloudFormation & VPC Peering! 🔐🌐

I recently completed an **end-to-end AWS VPC and EC2 infrastructure setup** for two isolated environments—**Aja** and **IBM**—with secure inter-VPC communication using **VPC Peering**. This project automated most of the provisioning using **CloudFormation templates**, enabling scalable and repeatable infrastructure.

🔧 Key Highlights:

✅ Created two VPCs (Aja & IBM)



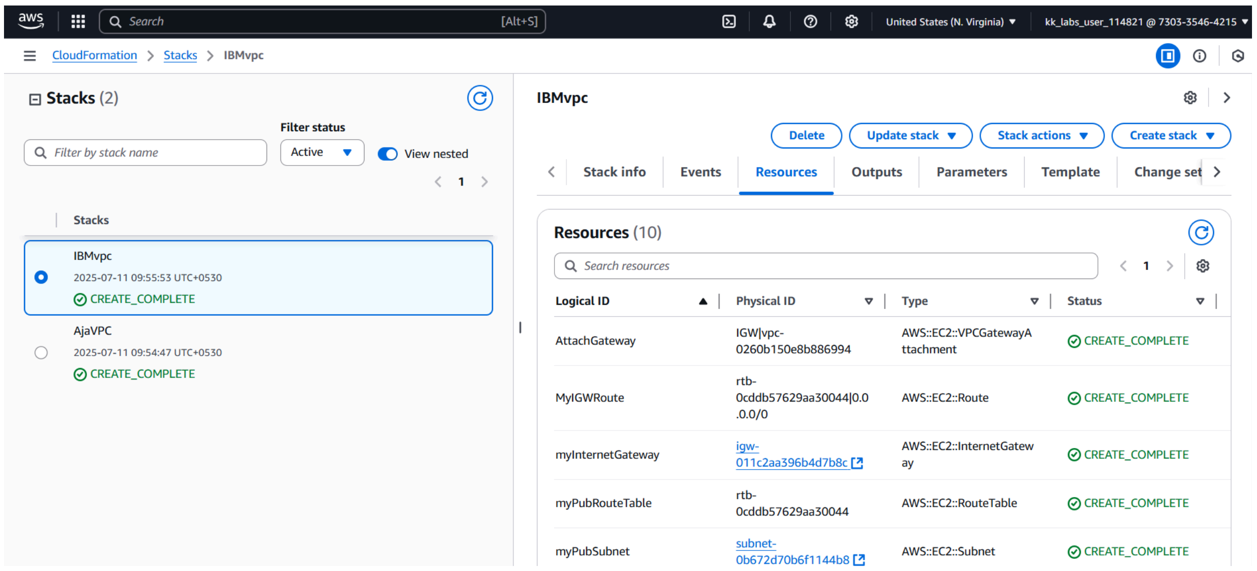
CloudFormation > Stacks > AjaVPC

Stacks (1)

Stacks
AjaVPC 2025-07-11 09:54:47 UTC+0530 CREATE_COMPLETE

AjaVPC — Resources (10)

Logical ID	Physical ID	Type	Status
AttachGateway	IGW|vpc-0035485a109ed828c	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE
MyIGWRoute	rtb-0813a35057517018d|0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE
myInternetGateway	igw-06042a5e9a988392d	AWS::EC2::InternetGateway	CREATE_COMPLETE
myPubRouteTable	rtb-0813a35057517018d	AWS::EC2::RouteTable	CREATE_COMPLETE
myPubSubnet	subnet-03f739a082d714d00	AWS::EC2::Subnet	CREATE_COMPLETE



CloudFormation > Stacks > IBMvpc

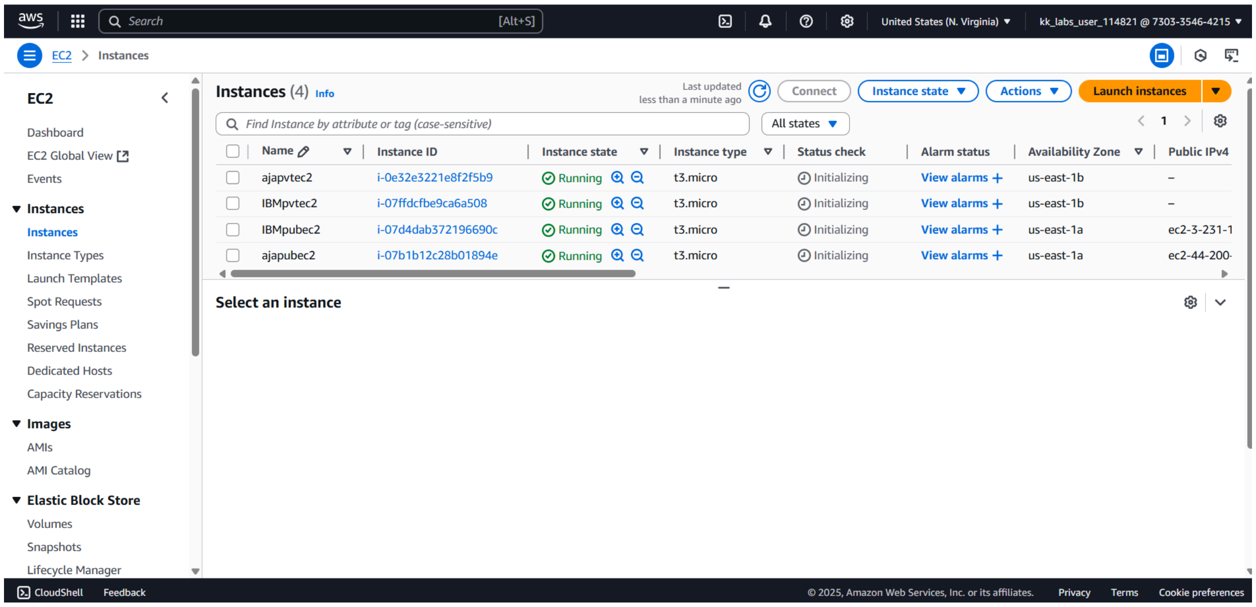
Stacks (2)

Stacks
IBMvpc 2025-07-11 09:55:53 UTC+0530 CREATE_COMPLETE
AjaVPC 2025-07-11 09:54:47 UTC+0530 CREATE_COMPLETE

IBMvpc — Resources (10)

Logical ID	Physical ID	Type	Status
AttachGateway	IGW|vpc-0260b150e8b886994	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE
MyIGWRoute	rtb-0cddb57629aa30044|0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE
myInternetGateway	igw-011c2aa396b4d7b8c	AWS::EC2::InternetGateway	CREATE_COMPLETE
myPubRouteTable	rtb-0cddb57629aa30044	AWS::EC2::RouteTable	CREATE_COMPLETE
myPubSubnet	subnet-0b672d70b6f1144b8	AWS::EC2::Subnet	CREATE_COMPLETE

✅ Launched EC2 instances (public & private) in both VPCs



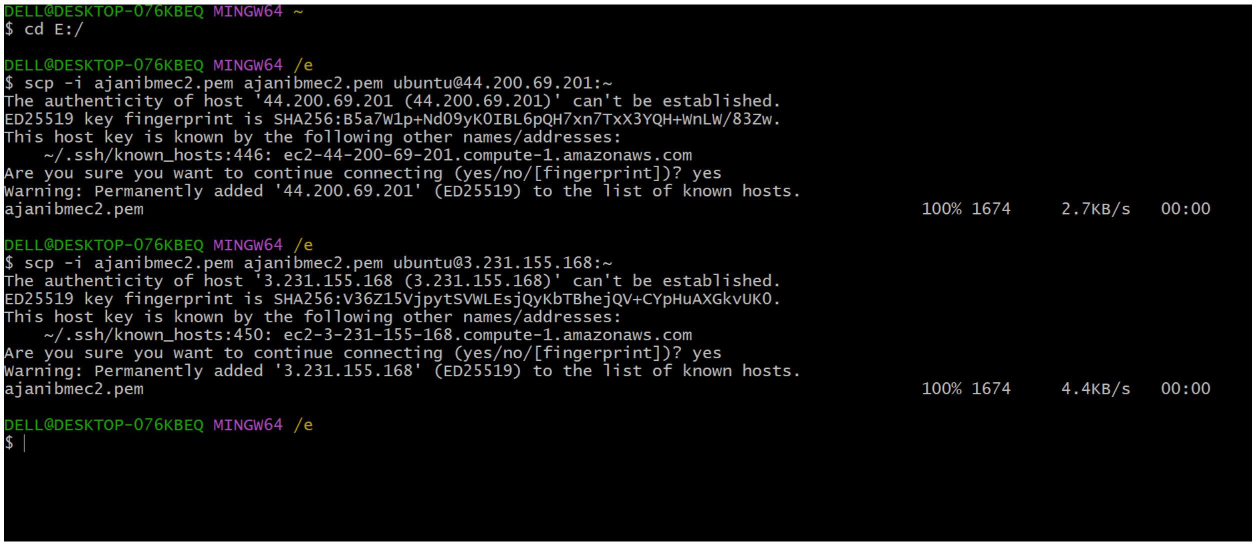
Instances (4) Info

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
ajapvtec2	i-0e32e3221e8f2f5b9	Running	t3.micro	Initializing	View alarms +	us-east-1b	–
IBMpvtec2	i-07ffdcfbe9ca6a508	Running	t3.micro	Initializing	View alarms +	us-east-1b	–
IBMpubec2	i-07d4dab372196690c	Running	t3.micro	Initializing	View alarms +	us-east-1a	ec2-3-231-1
ajapubec2	i-07b1b12c28b01894e	Running	t3.micro	Initializing	View alarms +	us-east-1a	ec2-44-200-

Select an instance

✅ Configured **SSH bastion access** for private EC2s



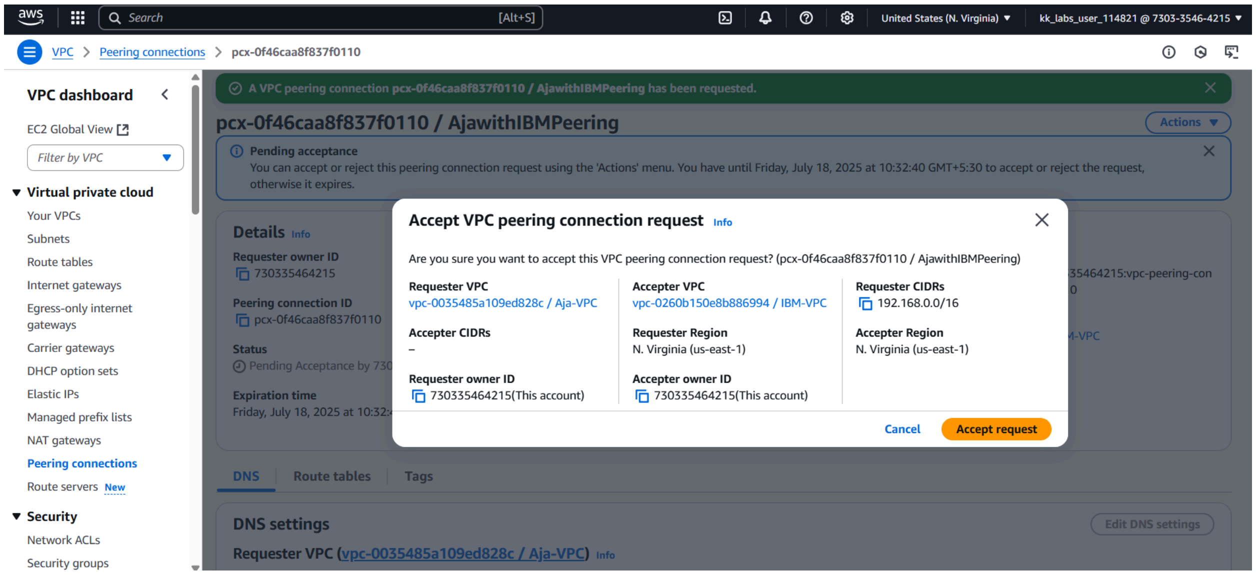
```
DELL@DESKTOP-076KBEQ MINGW64 ~
$ cd E:/

DELL@DESKTOP-076KBEQ MINGW64 /e
$ scp -i ajanibmec2.pem ajanibmec2.pem ubuntu@44.200.69.201:~
The authenticity of host '44.200.69.201 (44.200.69.201)' can't be established.
ED25519 key fingerprint is SHA256:B5a7W1p+Nd09yKOIBL6pQH7xn7TxX3YQH+WnLW/83Zw.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:446: ec2-44-200-69-201.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.200.69.201' (ED25519) to the list of known hosts.
ajanibmec2.pem                                                       100% 1674     2.7KB/s   00:00

DELL@DESKTOP-076KBEQ MINGW64 /e
$ scp -i ajanibmec2.pem ajanibmec2.pem ubuntu@3.231.155.168:~
The authenticity of host '3.231.155.168 (3.231.155.168)' can't be established.
ED25519 key fingerprint is SHA256:V36Z15VjpytSVWLEsjQyKbTBhejQV+CYpHuAXGkvUKO.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:450: ec2-3-231-155-168.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.231.155.168' (ED25519) to the list of known hosts.
ajanibmec2.pem                                                       100% 1674     4.4KB/s   00:00

DELL@DESKTOP-076KBEQ MINGW64 /e
$ |
```

✅ Established **VPC Peering** for seamless cross-VPC private communication



A VPC peering connection pcx-0f46caa8f837f0110 / AjawithIBMPeering has been requested.

pcx-0f46caa8f837f0110 / AjawithIBMPeering

Pending acceptance
You can accept or reject this peering connection request using the 'Actions' menu. You have until Friday, July 18, 2025 at 10:32:40 GMT+5:30 to accept or reject the request, otherwise it expires.

Details Info
Requester owner ID
730335464215
Peering connection ID
pcx-0f46caa8f837f0110
Status
Pending Acceptance by 730
Expiration time
Friday, July 18, 2025 at 10:32:

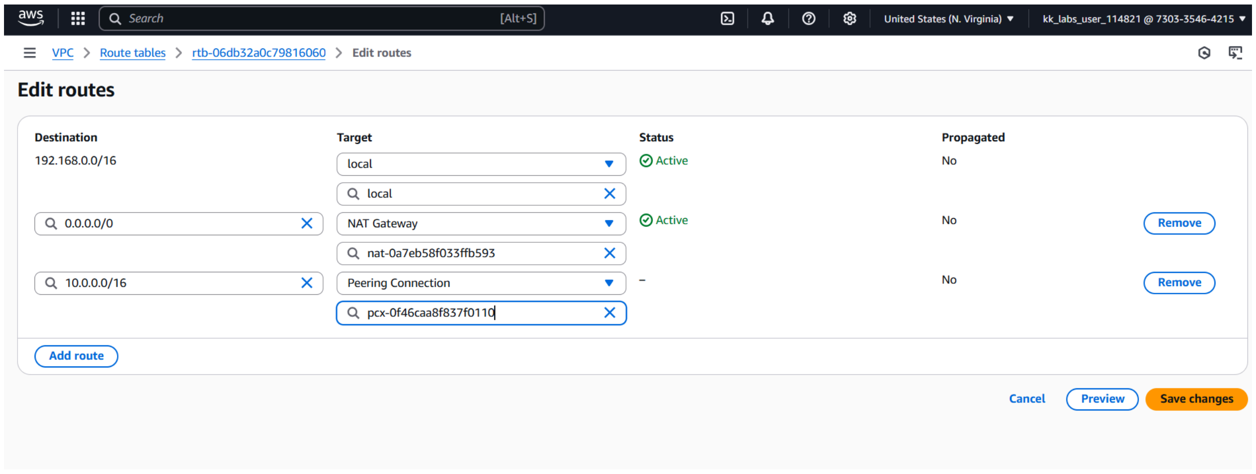
Accept VPC peering connection request Info

Are you sure you want to accept this VPC peering connection request? (pcx-0f46caa8f837f0110 / AjawithIBMPeering)

Requester VPC	Accepter VPC	Requester CIDRs
vpc-0035485a109ed828c / Aja-VPC	vpc-0260b150e8b886994 / IBM-VPC	192.168.0.0/16
Accepter CIDRs	**Requester Region**	**Accepter Region**
–	N. Virginia (us-east-1)	N. Virginia (us-east-1)
Requester owner ID	**Accepter owner ID**	
730335464215(This account)	730335464215(This account)	

DNS settings
Requester VPC (vpc-0035485a109ed828c / Aja-VPC) Info

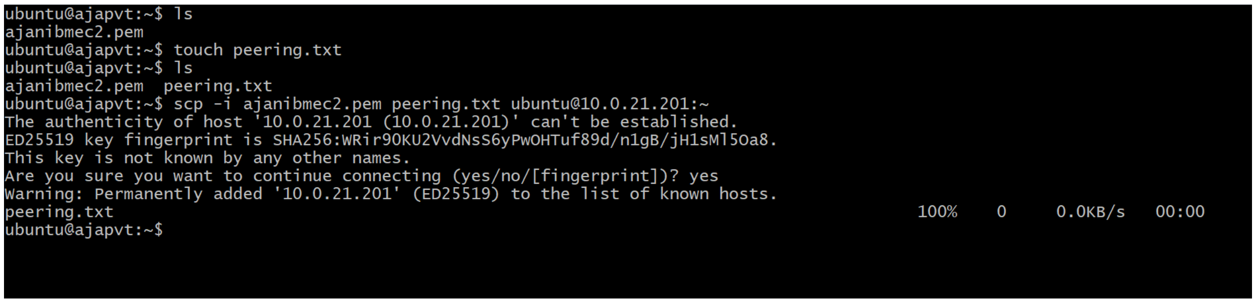
✅ Route tables updated to allow internal traffic across peered VPCs


aws
Search [Alt+S]
United States (N. Virginia) ▾
kk_labs_user_114821 @ 7303-3546-4215 ▾
VPC > Route tables > rtb-06db32a0c79816060 > Edit routes
Remove
Remove
Add route
Cancel
Preview
Save changes

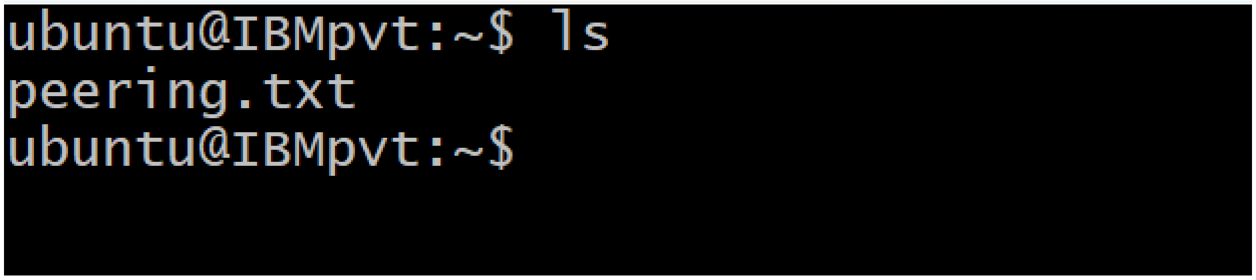
Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local / local	Active	No
0.0.0.0/0	NAT Gateway / nat-0a7eb58f033ffb593	Active	No
10.0.0.0/16	Peering Connection / pcx-0f46caa8f837f0110	-	No

✅ Demonstrated **file transfer** between private EC2 instances in different VPCs—completely Private!



```
ubuntu@ajapvt:~$ ls
ajanibmec2.pem
ubuntu@ajapvt:~$ touch peering.txt
ubuntu@ajapvt:~$ ls
ajanibmec2.pem  peering.txt
ubuntu@ajapvt:~$ scp -i ajanibmec2.pem peering.txt ubuntu@10.0.21.201:~
The authenticity of host '10.0.21.201 (10.0.21.201)' can't be established.
ED25519 key fingerprint is SHA256:WRir9OKU2VvdNsS6yPwOHTuf89d/n1gB/jH1sMl5Oa8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.21.201' (ED25519) to the list of known hosts.
peering.txt                                                                  100%    0     0.0KB/s   00:00
ubuntu@ajapvt:~$
```



```
ubuntu@IBMpvt:~$ ls
peering.txt
ubuntu@IBMpvt:~$
```

📄 All steps were meticulously scripted using **YAML-based CloudFormation**—ensuring automation, security, and scalability.

💡 This setup is ideal for **multi-environment architectures** (e.g., Dev/Prod or Partner Integrations) needing isolated control and secure connectivity.