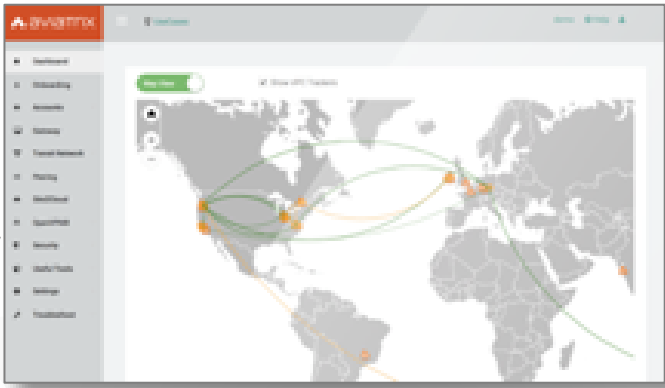


# Aviatrix Cloud Network Platform Software

Not a SaaS or  
Managed  
Service.  
It's Yours.

 **Terraform**  
Single Multi-Cloud Provider



Automation and  
Operational  
Control

**Aviatrix  
CoPilot**

**1** **Aviatrix  
Controller  
Software**

**2** **Aviatrix Gateways  
Software**

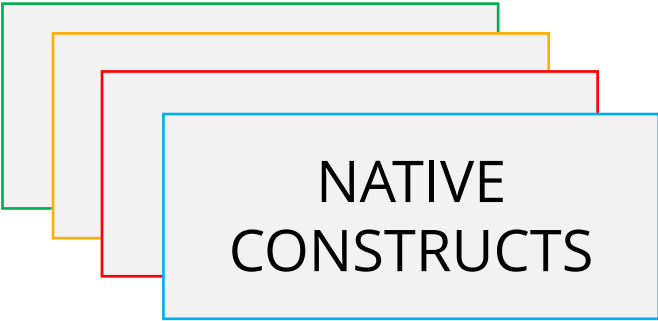
**Aviatrix Transit**

Advanced  
Networking  
and Security

Cloud Networking Abstraction

API

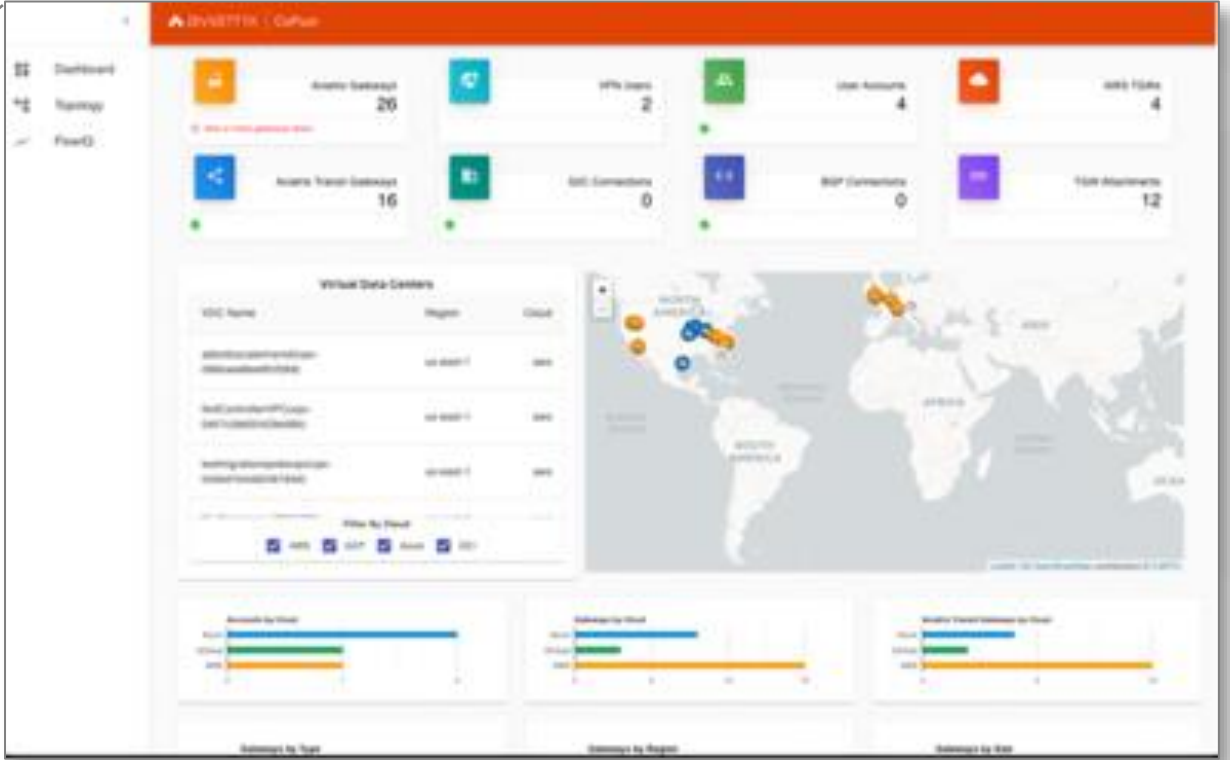
API

  
NATIVE  
CONSTRUCTS

Basic Cloud  
Network & Security

**FORTINET**  
 **Check Point**  
SOFTWARE TECHNOLOGIES LTD.  
 **paloalto**  
NETWORKS

**Service Insertion  
and Chaining**




**Multi-Cloud  
Operational  
Visibility**



**FlowIQ  
Multi-Cloud  
Traffic Flow  
Analysis**



**Multi-Cloud  
Dynamic  
Topology  
Mapping**



# AWS Immersion Day LAB 2

BUILD AND TROUBLESHOOT YOUR CLOUD NETWORK BACKBONE

**Brad Hedlund**  
Principal Solutions Architect,  
Aviatrix Systems

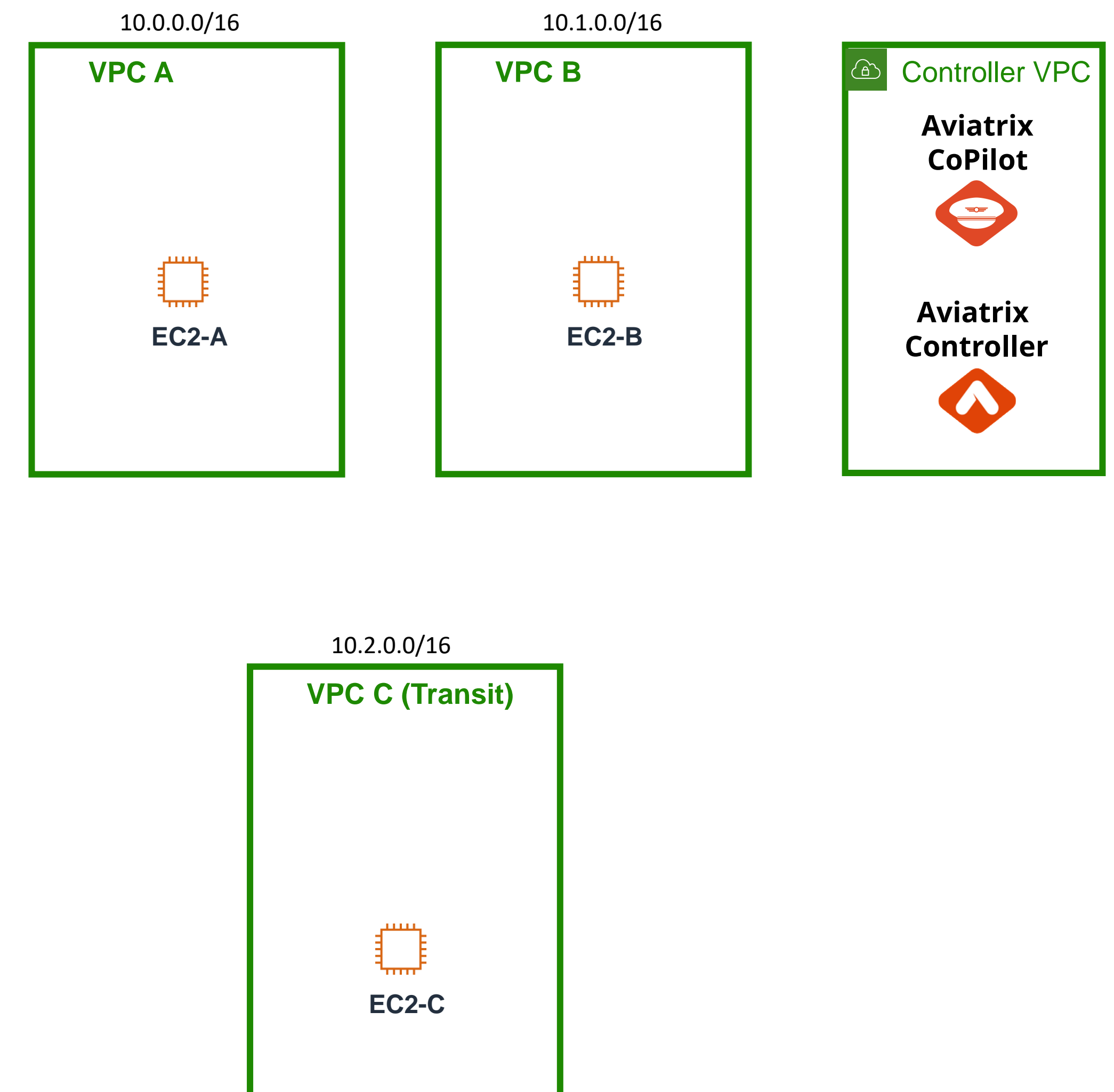
## Lab 1 Recap

In Lab 1 you created (3) AWS VPCs in the us-east-1 region, each with an EC2 instance.

There is a 4<sup>th</sup> VPC called the “Controller VPC” that contains your Aviatrix Controller and Aviatrix CoPilot as EC2 instances

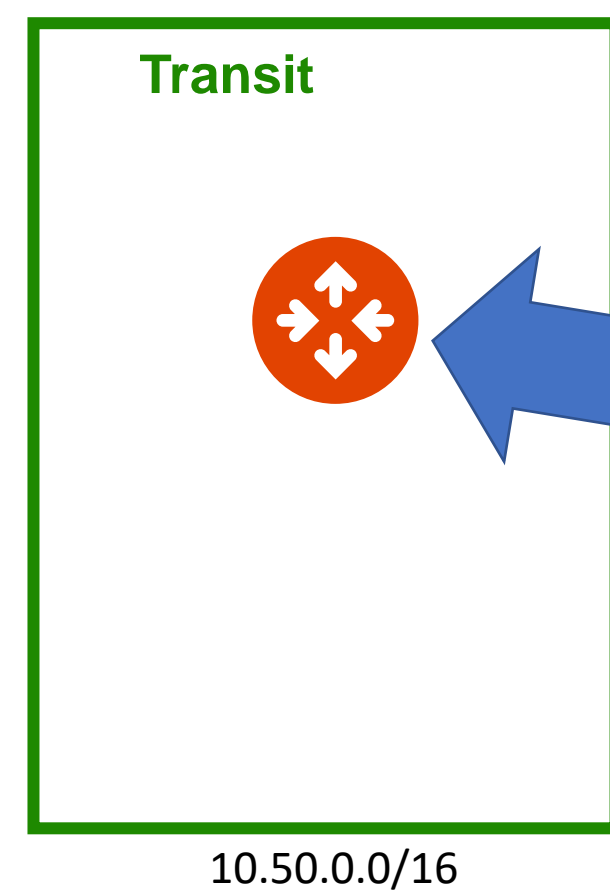
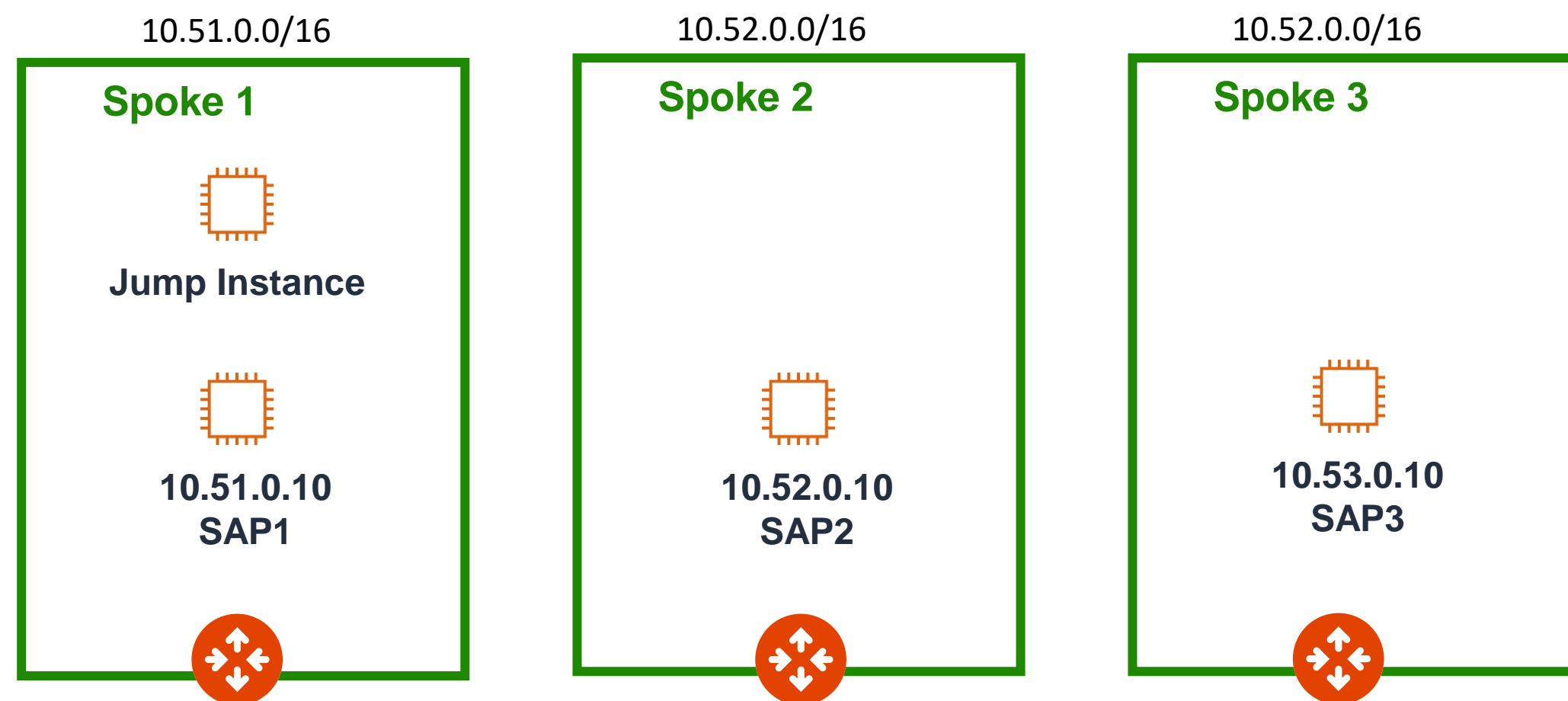
Next, you will use Aviatrix CoPilot to build a multi-region cloud network backbone...

### AWS us-east-1



## Lab 2 Intro

### AWS us-west-2

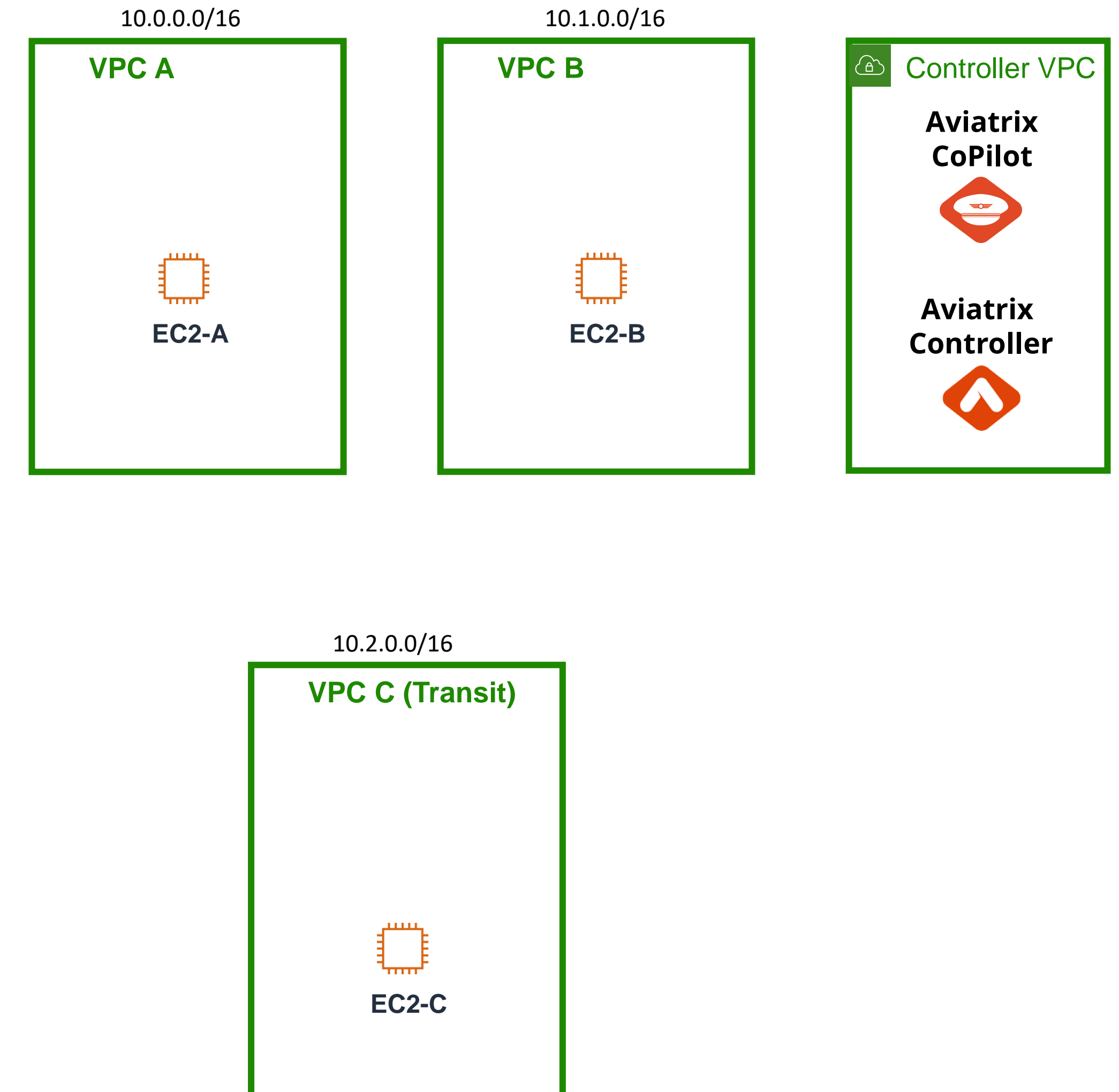


← Your AWS lab account also has (4) VPCs in the us-west-2 region, (3) Spoke VPCs and (1) Transit VPC.

← The VPCs in us-west-2 already have Aviaatrix Gateways deployed (but not connected to anything).

**In Lab 2 you will deploy Aviaatrix gateways in us-east-1 and connect the two AWS regions together.**

### AWS us-east-1

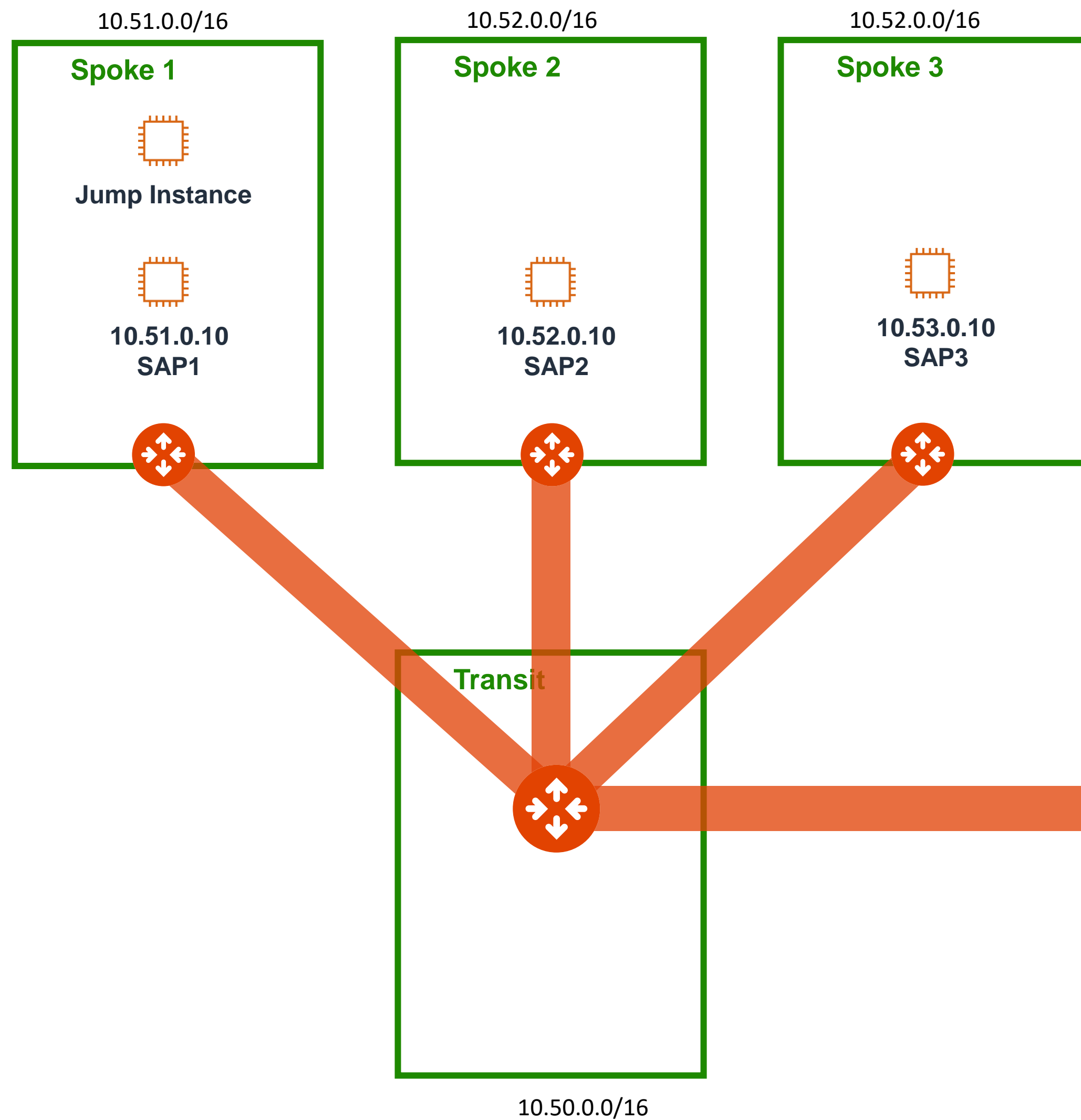




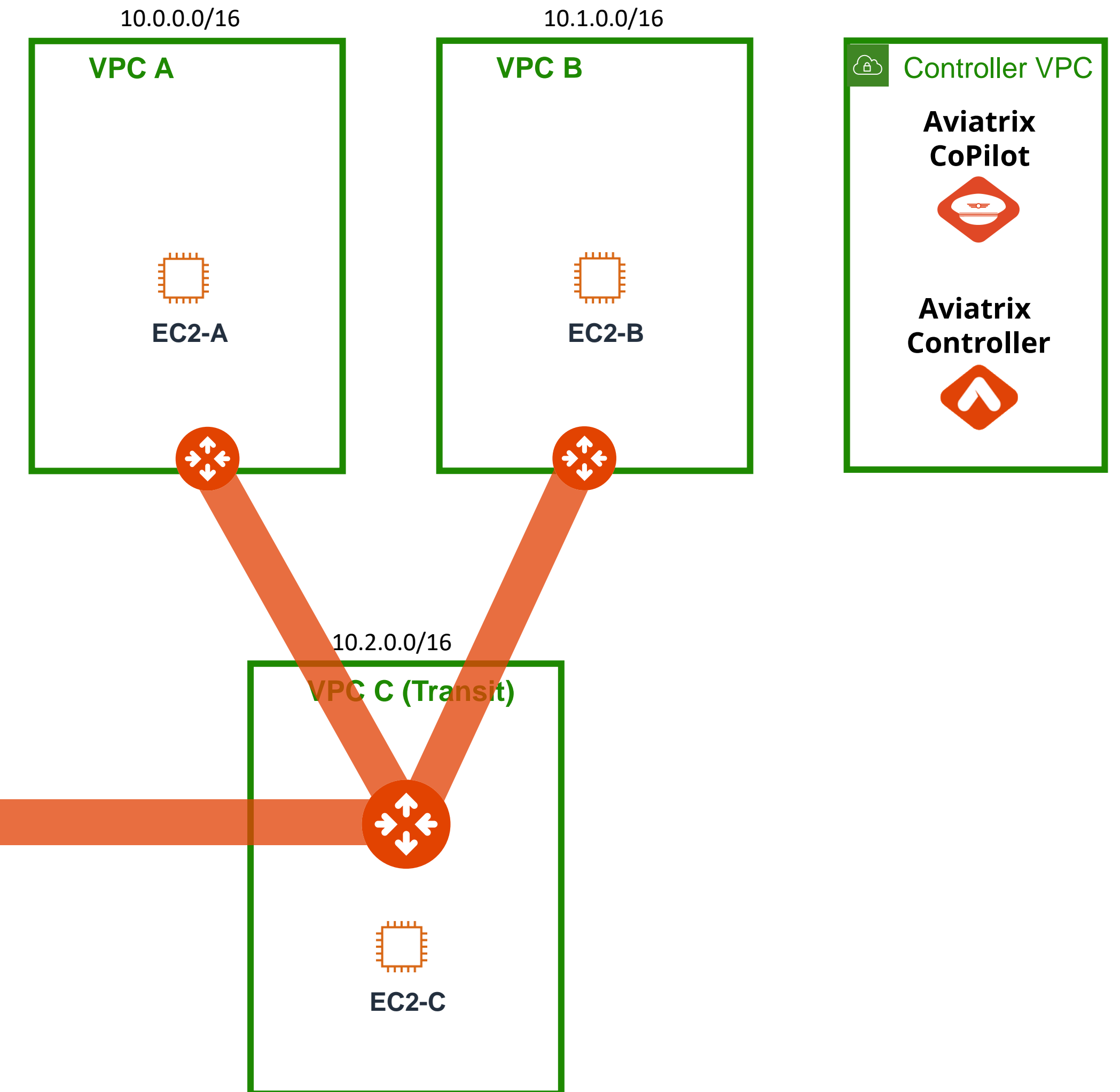
# Lab 2: Cloud Backbone

When you are done with Lab 2, your cloud network will look like this

## AWS us-west-2



## AWS us-east-1



Let's get started!

## Lab 2: Step 2.0

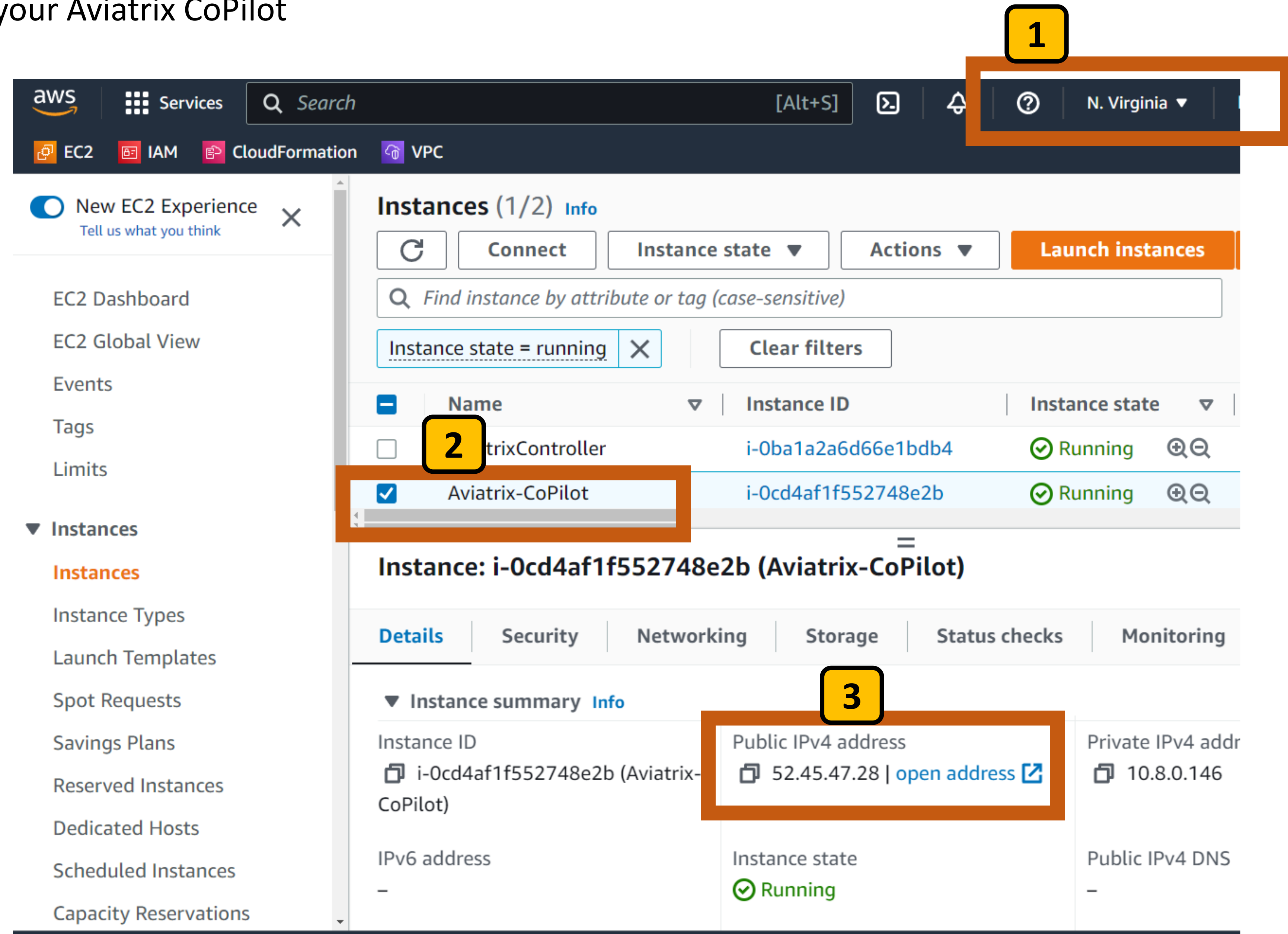
Log in to your Aviatrix CoPilot

Go to the us-east-1 **N. Virginia** region in your AWS Console **1**

Select the Aviatrix-Copilot EC2 instance **2**

On the Details tab, find the Public IP address and click on “open address” **3**

This will open a browser tab to log on to the Aviatrix Copilot UI



The screenshot shows the AWS Management Console interface. At the top right, the region is set to 'N. Virginia' (labeled with a yellow box and '1'). The left sidebar shows the 'Instances' section expanded. In the main content area, the 'Instances (1/2)' list shows two instances: 'AviatrixController' and 'Aviatrix-Copilot'. The 'Aviatrix-Copilot' instance is selected (labeled with a yellow box and '2'). Below the list, the 'Details' tab for instance 'i-0cd4af1f552748e2b (Aviatrix-Copilot)' is active. In the 'Instance summary' section, the 'Public IPv4 address' is '52.45.47.28', and the 'open address' link is highlighted (labeled with a yellow box and '3').

Name	Instance ID	Instance state
AviatrixController	i-0ba1a2a6d66e1bdb4	Running
Aviatrix-Copilot	i-0cd4af1f552748e2b	Running

Instance: i-0cd4af1f552748e2b (Aviatrix-Copilot)		
Details	Security	Networking
<b>Instance summary</b>		
Instance ID i-0cd4af1f552748e2b (Aviatrix-Copilot)	Public IPv4 address 52.45.47.28   <a href="#">open address</a>	Private IPv4 address 10.8.0.146
IPv6 address -	Instance state Running	Public IPv4 DNS -

## Lab 2: Step 2.1

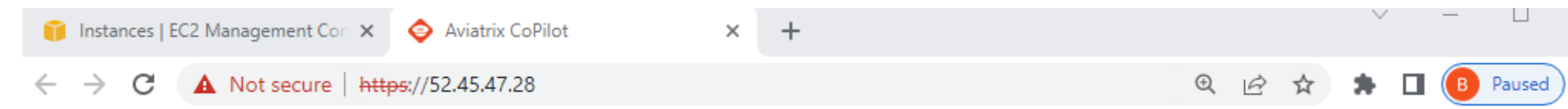
Log in to your Aviatix CoPilot

Acknowledge the Certificate warning by clicking Advanced, then Proceed (Chrome browser) **1**

You should see the Aviatix CoPilot logon page **2**

Username = lab\_student

Password = ImmersionDay123# **3**



Username

lab\_student **3**

Password

ImmersionDay123#

Log In

☐ Remember Me [Forgot Password](#)



Your connection is not private

Attackers might be trying to steal your information from 34.239.54.147 (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID

To get Chrome's highest level of security, [turn on enhanced protection](#)

Advanced

Back to safety

**1**

This server could not prove that it is 34.239.54.147; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an expired/invalid certificate. Proceeding to this page will expose you to a variety of security risks. Do not proceed with accepting your connection.

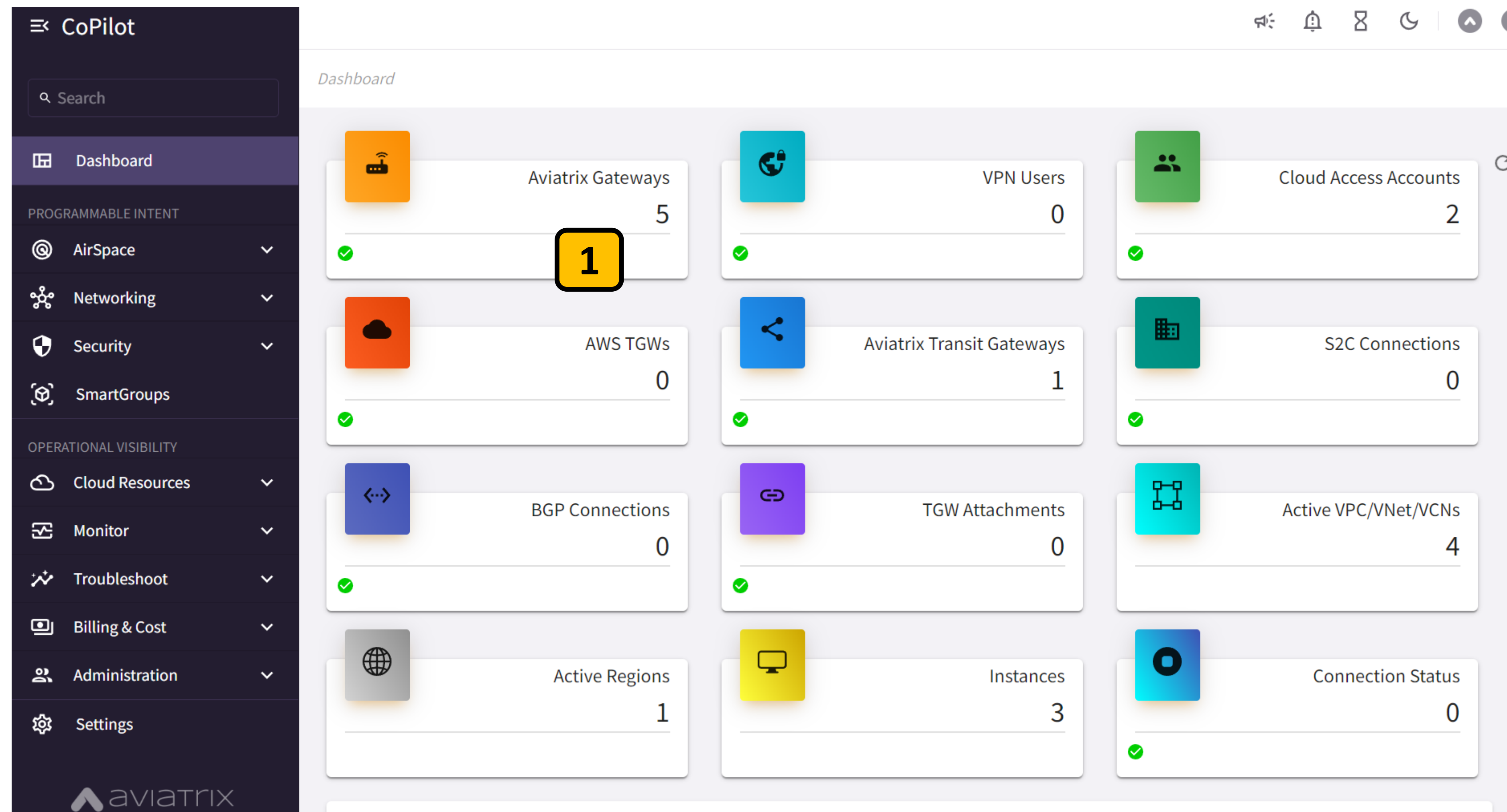
Proceed to 34.239.54.147 (unsafe)



## Lab 2: Step 2.2

Log in to your Aviatrix CoPilot

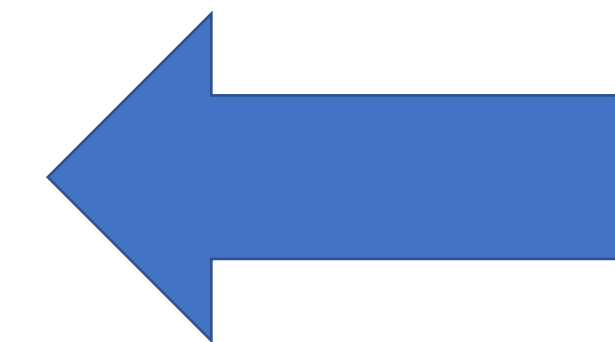
You should now see your Aviatrix CoPilot UI  
Take a minute to checkout the main Dashboard page and  
all the info it provides.



The screenshot shows the Aviatrix CoPilot Dashboard. On the left is a dark sidebar with the 'CoPilot' header, a search bar, and a 'Dashboard' button. Below this are sections for 'PROGRAMMABLE INTENT' (AirSpace, Networking, Security, SmartGroups) and 'OPERATIONAL VISIBILITY' (Cloud Resources, Monitor, Troubleshoot, Billing & Cost, Administration, Settings). The main dashboard area displays a grid of 12 widgets, each with a colored icon, a title, a value, and a green checkmark. A yellow box with the number '1' highlights the 'Aviatrix Gateways' widget. A blue arrow points from the text 'Click on any widget to get more info 1' towards the dashboard.

Widget Title	Value
Aviatrix Gateways	5
VPN Users	0
Cloud Access Accounts	2
AWS TGWs	0
Aviatrix Transit Gateways	1
S2C Connections	0
BGP Connections	0
TGW Attachments	0
Active VPC/VNet/VCNs	4
Active Regions	1
Instances	3
Connection Status	0

Click on any widget to  
get more info **1**





## Lab 2: Cloud Backbone: Step 2.3

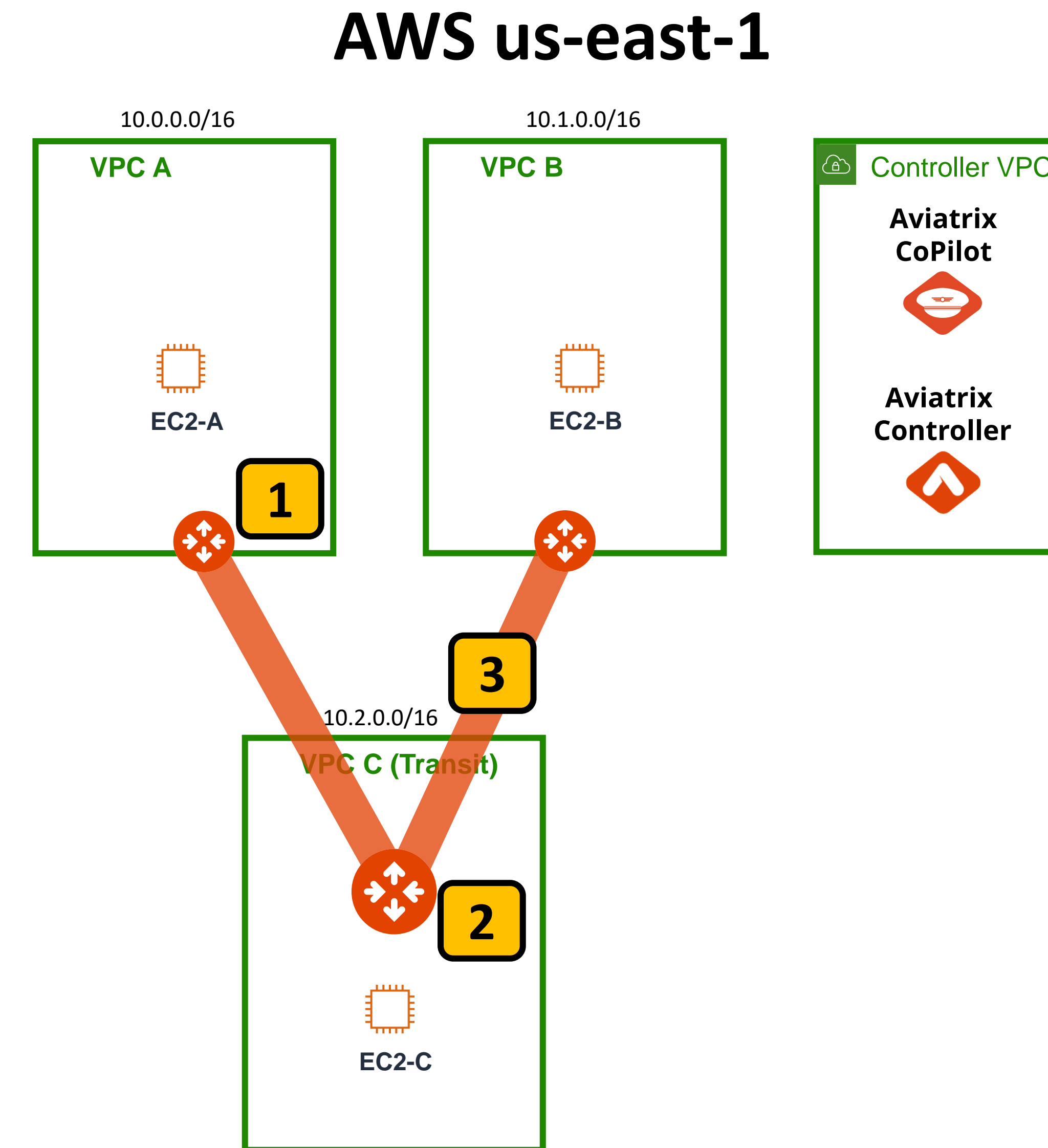
Network your VPCs in us-east-1 with Aviaatrix Spoke and Transit Gateways

You will start by creating a hub and spoke architecture in your first region, us-east-1.

VPCs A & B will be our Spoke VPCs, where we will deploy Aviaatrix Spoke Gateways. **1**

VPC C will be the Transit VPC where we will deploy Aviaatrix Transit Gateway (the “hub”). **2**

We will connect the Spokes to the Transit with secure IPsec connections. **3**



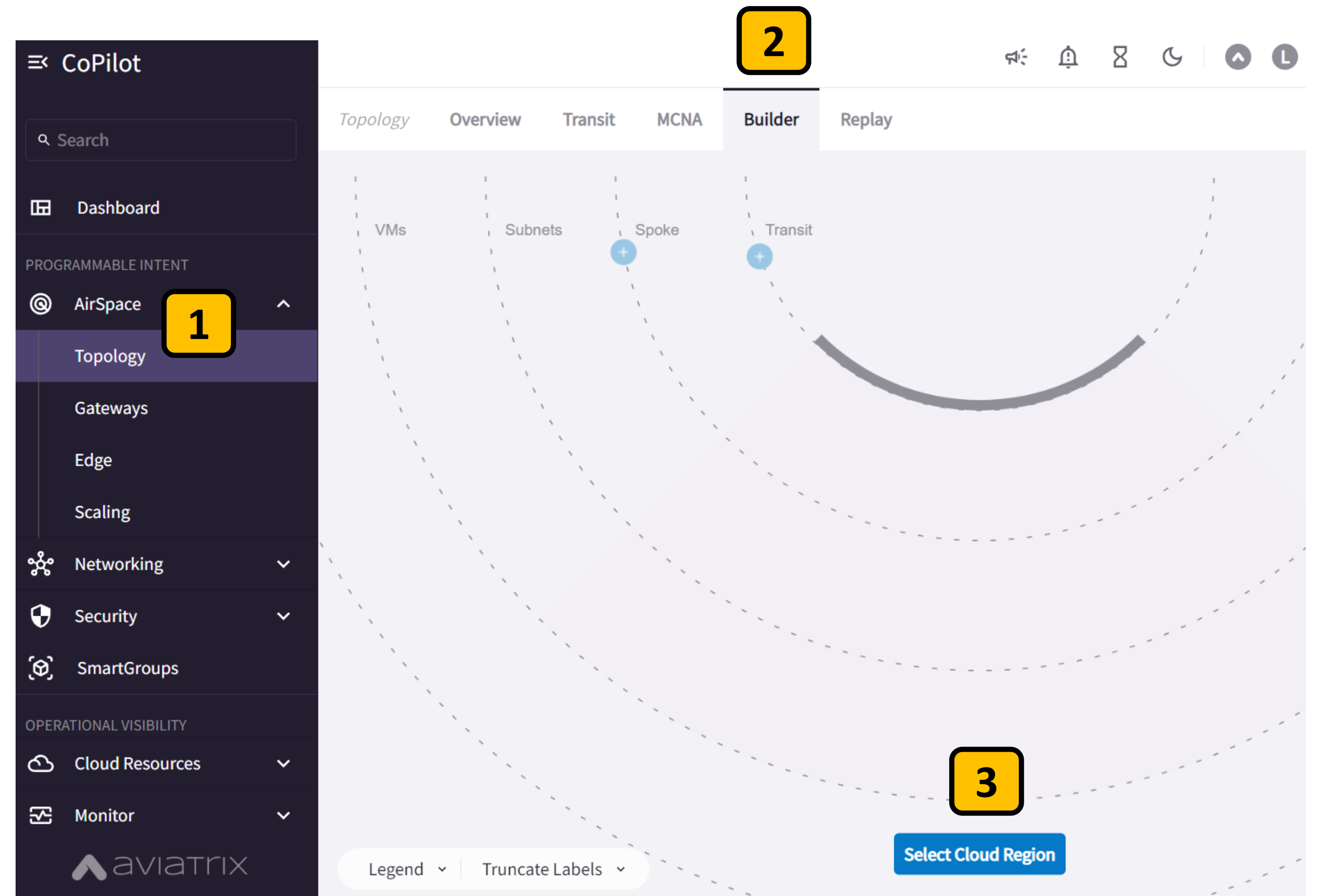
## Lab 2: Cloud Backbone: Step 2.4

Use the Topology Builder to launch an Aviaatrix Transit Gateway in a Transit VPC

From the CoPilot UI select AirSpace > Topology **1**

Select the Builder tab **2**

Click on Select Cloud Region **3**



## Lab 2: Cloud Backbone: Step 2.5

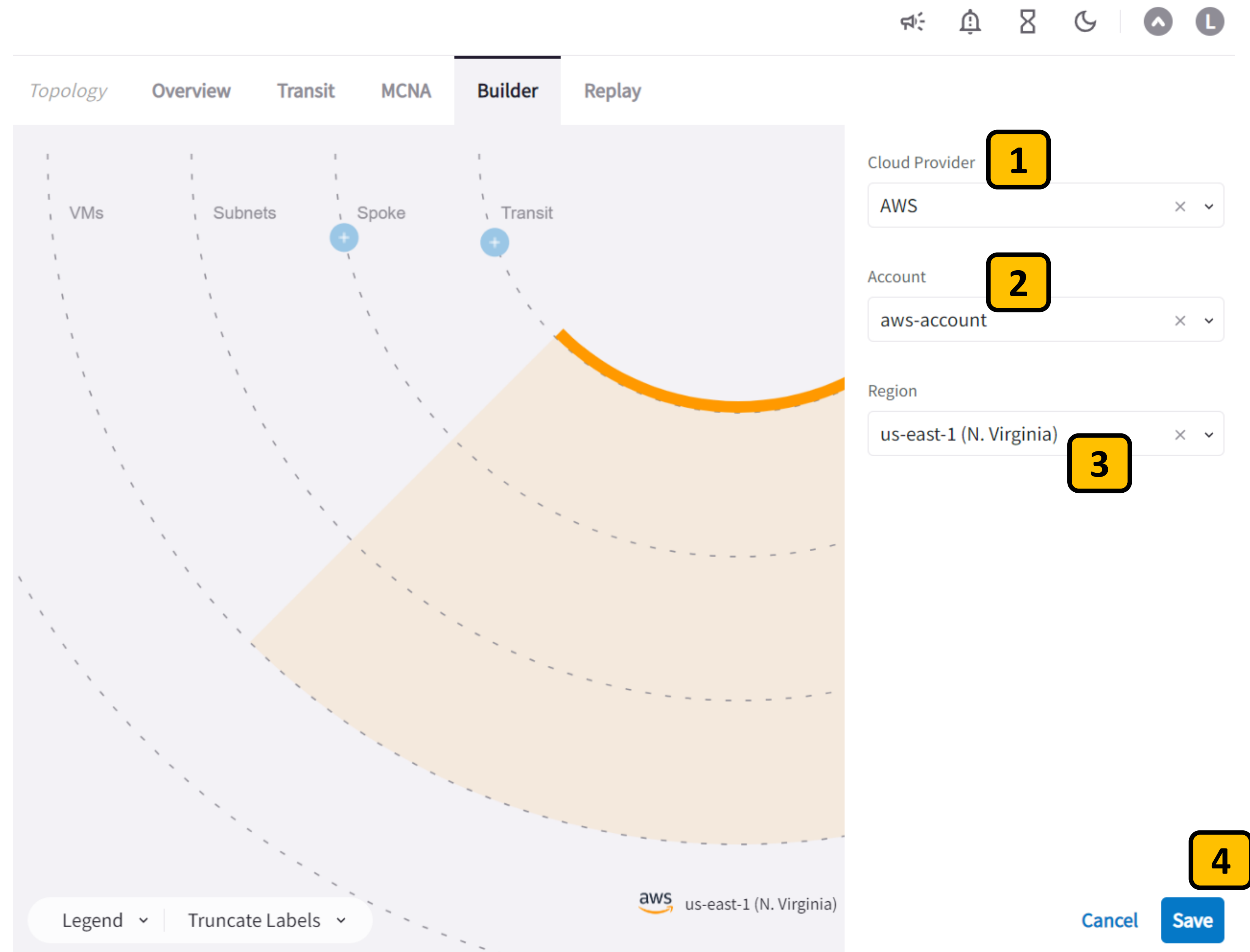
Use the Topology Builder to launch an Aviaatrix Transit Gateway in a Transit VPC

Select the Cloud Provider: **AWS** **1**

Select the AWS Account: **aws-account** **2**

Select the Region: **us-east-1** **3**

Click **Save** to start a new Build session in this cloud, account, and region. **4**



The screenshot shows the Aviaatrix Topology Builder interface. The 'Builder' tab is selected, and a large orange curved arrow indicates the flow from the configuration panel on the right to the main workspace. The configuration panel on the right has four numbered steps:

- 1** Cloud Provider: AWS
- 2** Account: aws-account
- 3** Region: us-east-1 (N. Virginia)
- 4** Save

The main workspace shows a diagram with labels for VMs, Subnets, Spoke, and Transit. The bottom right corner of the workspace displays the AWS logo and the region 'us-east-1 (N. Virginia)'. The bottom of the interface has a 'Legend' and 'Truncate Labels' dropdown menu.

## Lab 2: Cloud Backbone: Step 2.6

Use the Topology Builder to launch an Aviatrix Transit Gateway in a Transit VPC

Click the **blue +** under Transit **1**

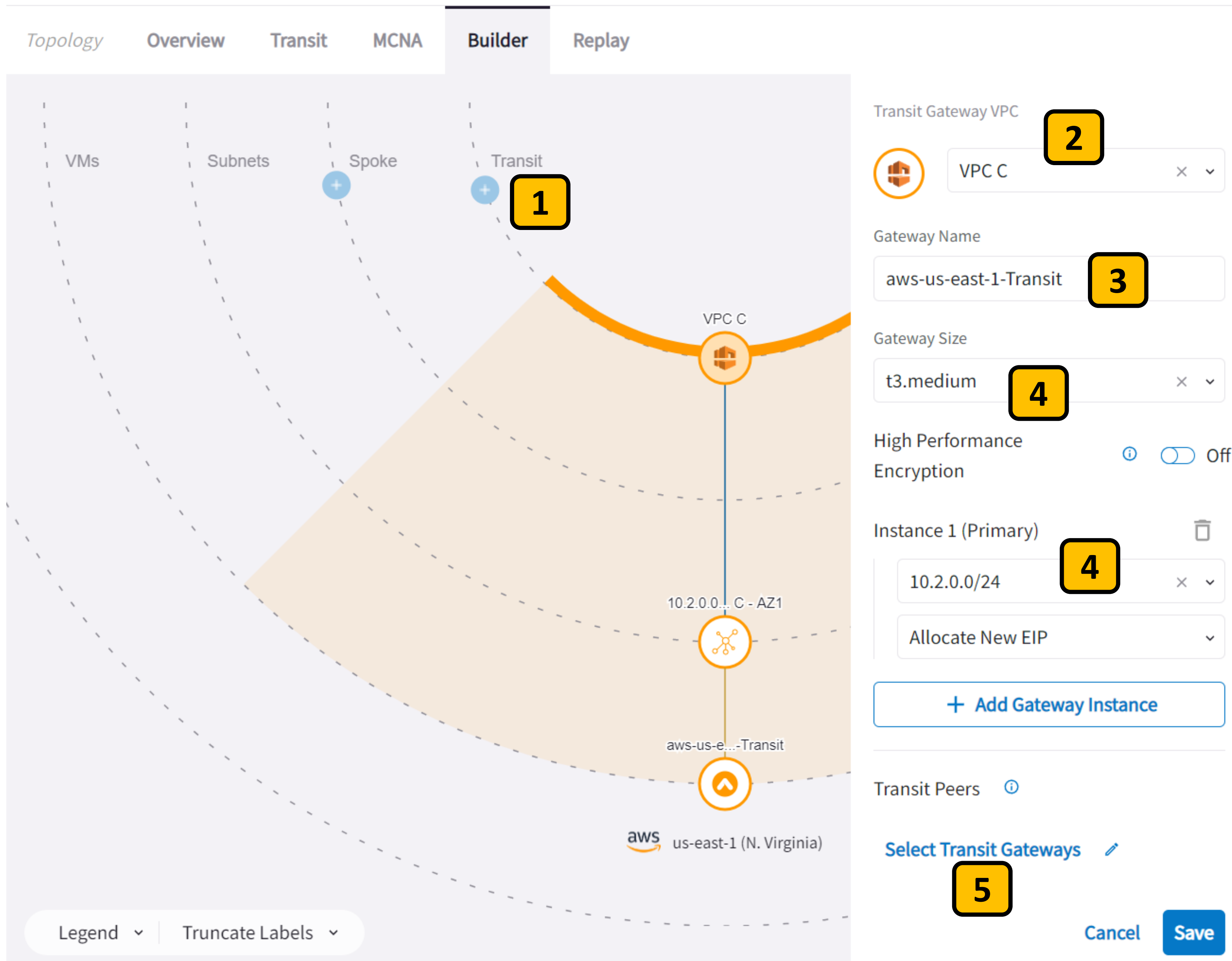
Select **VPC C** to be the Aviatrix Transit Gateway VPC **2**

Name the gateway: **aws-us-east-1-Transit** **3**

Select **t3.medium** for the gateway instance size **4**

Select the 10.2.0.0/24 as the public subnet to deploy the gateway in to **4**

Let's peer this Transit to us-west-2 by clicking **Select Transit Gateways** **5**



The screenshot shows the Aviatrix Topology Builder interface with the 'Builder' tab selected. The main canvas displays a network topology with a central 'Transit' node (blue circle with a plus sign) and a 'Spoke' node (blue circle with a plus sign). A yellow shaded area represents the 'Transit Gateway VPC' (VPC C) and its associated subnets. The gateway is named 'aws-us-east-1-Transit' and is configured with a 't3.medium' instance size. The public subnet '10.2.0.0/24' is selected for deployment. The gateway is connected to the 'aws-us-east-1 (N. Virginia)' region. The right sidebar shows the configuration details for the Transit Gateway VPC, including the VPC name, gateway name, size, and instance configuration. The bottom right corner has 'Cancel' and 'Save' buttons.

Topology Overview Transit MCNA Builder Replay

VMs Subnets Spoke Transit

VPC C

10.2.0.0/24 C - AZ1

aws-us-east-1-Transit

aws us-east-1 (N. Virginia)

Legend Truncate Labels

Transit Gateway VPC

VPC C

Gateway Name

aws-us-east-1-Transit

Gateway Size

t3.medium

High Performance Encryption

Off

Instance 1 (Primary)

10.2.0.0/24

Allocate New EIP

+ Add Gateway Instance

Transit Peers

Select Transit Gateways

Cancel Save



## Lab 2: Cloud Backbone: Step 2.7

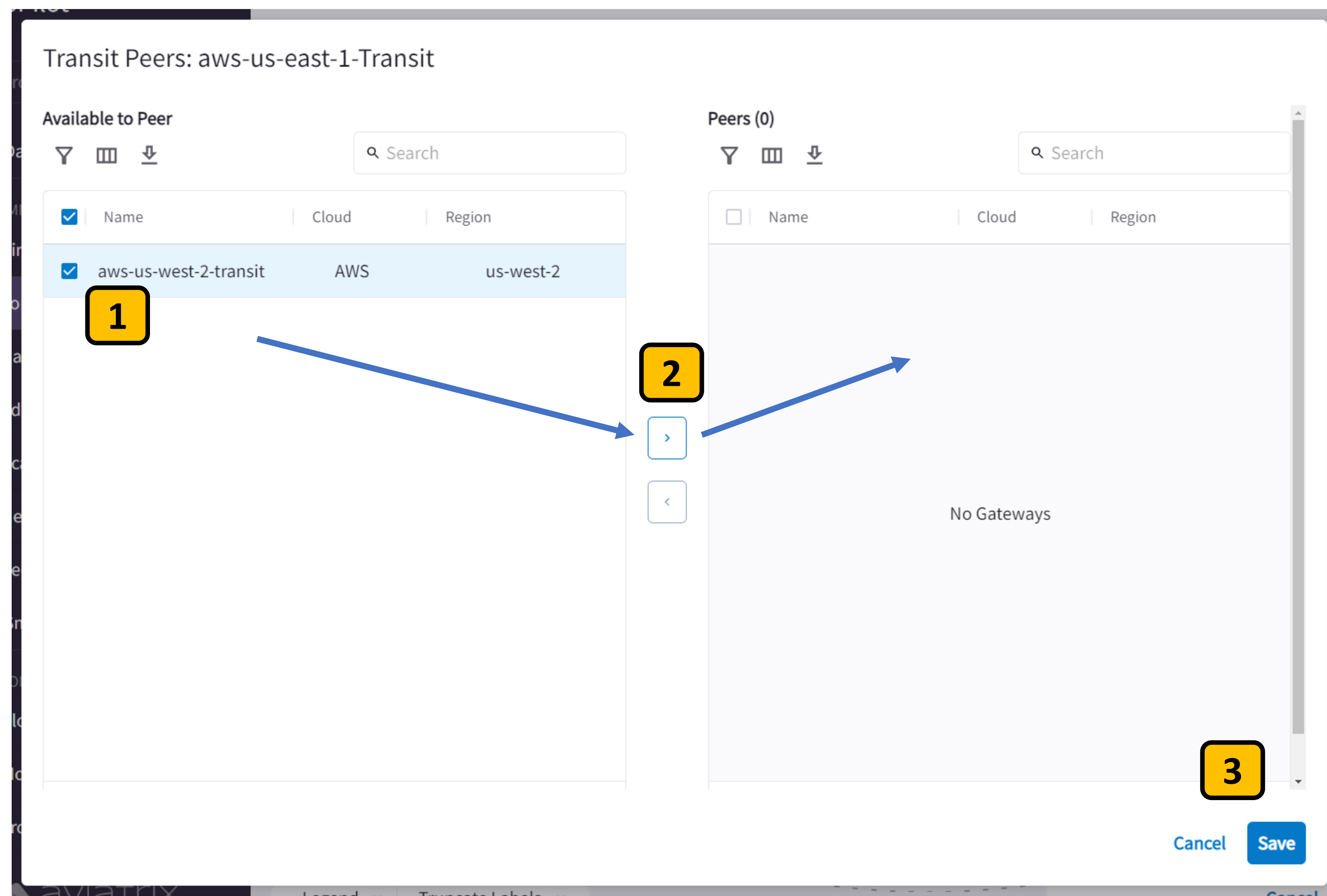
Use the Topology Builder to launch an Aviatrix Transit Gateway in a Transit VPC

You will be asked which Aviatrix Transit you want to peer to. Select the aws-us-west-2-Transit. **1**

Click the right arrow to move the aws-us-west-2-Transit to the peering column **2**

We have now instructed CoPilot to peer our new Aviatrix Transit in us-east-1 to the Aviatrix Transit in us-west-2.

Click Save **3**



## Lab 2: Cloud Backbone: Step 2.8

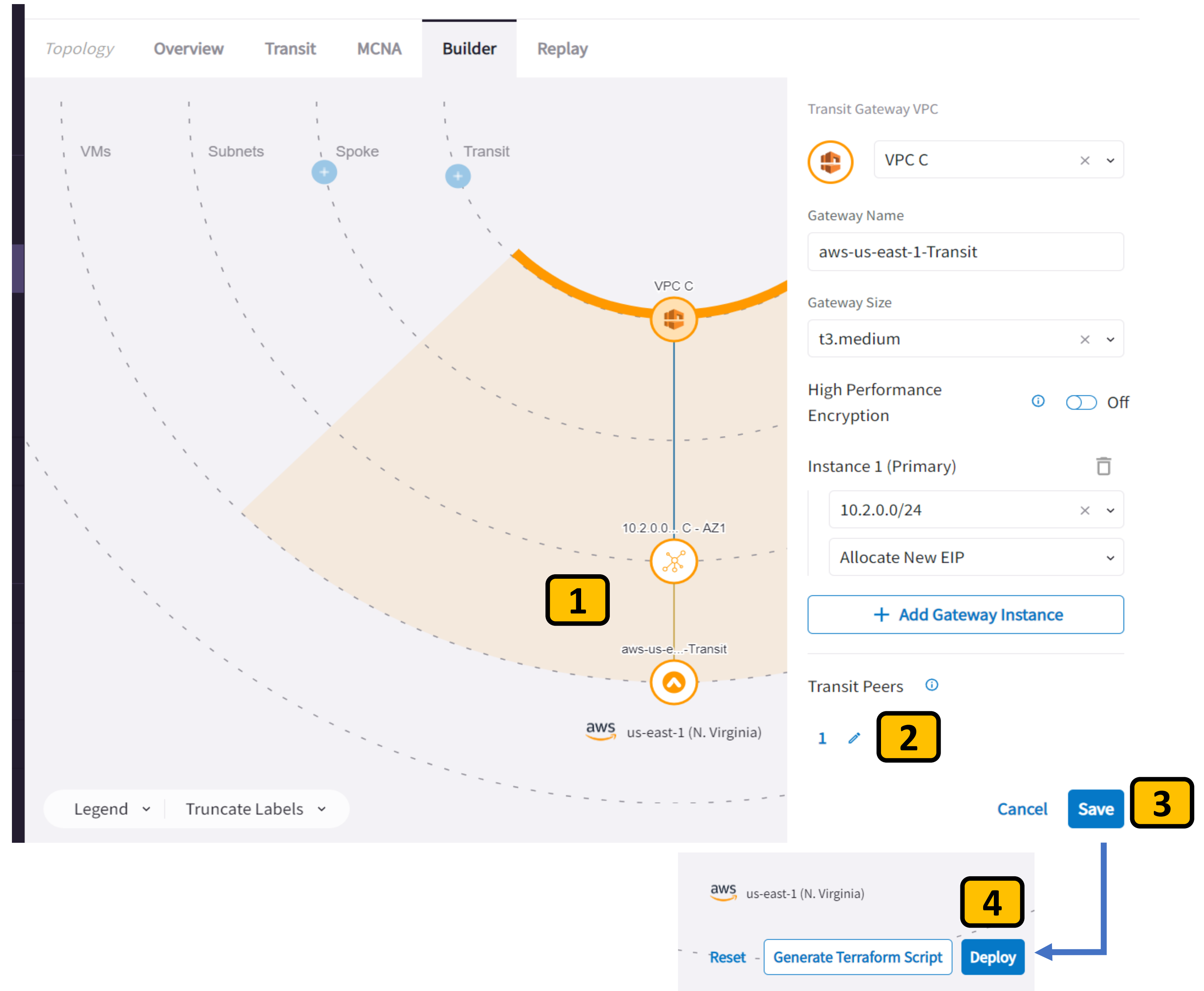
Use the Topology Builder to launch an Aviatrix Transit Gateway in a Transit VPC

You build session should now look like this. **1**

After deployment, CoPilot will peer this Aviatrix Transit Gateway to (1) other transit that you specified, aws-us-west-2-Transit **2**

We are ready to deploy, so.. Click Save. **3**

Then click Deploy **4**



The screenshot displays the Aviatrix Topology Builder interface. The 'Builder' tab is selected, showing a network diagram with a central VPC C and a transit gateway instance. The right sidebar contains configuration options for the Transit Gateway VPC, including Gateway Name (aws-us-east-1-Transit), Gateway Size (t3.medium), High Performance Encryption (Off), and Instance 1 (Primary) configuration. The bottom right shows the Transit Peers section with a list of peers and a 'Save' button. The bottom left shows the 'Deploy' button.

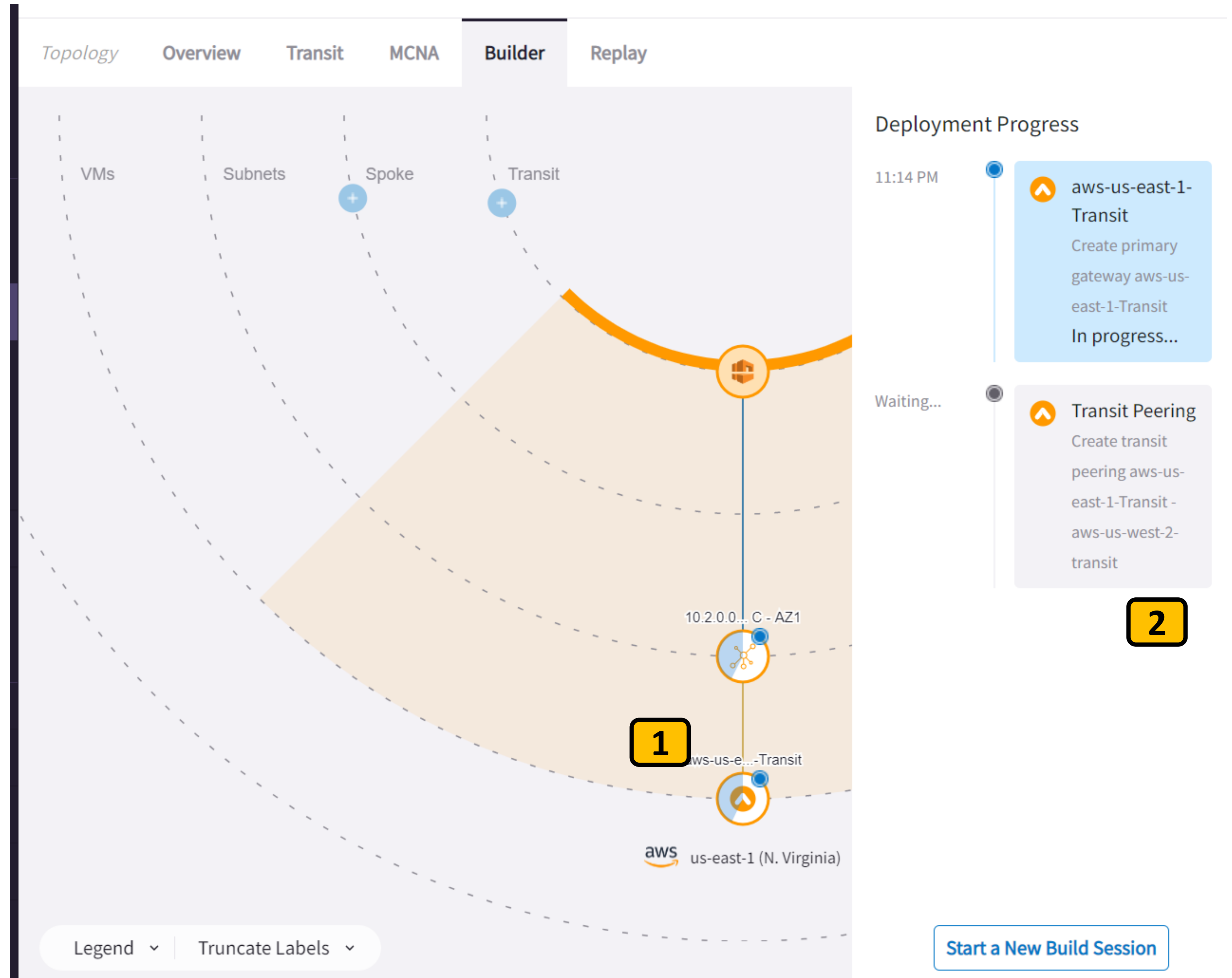
## Lab 2: Cloud Backbone: Step 2.9

Use the Topology Builder to launch an Aviatrix Transit Gateway in a Transit VPC

Observe the Deployment Progress of your Aviatrix Transit Gateway in us-east-1. **1**

After the gateway is deployed, CoPilot will peer this transit to us-west-2, just like you asked it to. **2**

The total deployment time should take about 5 minutes.



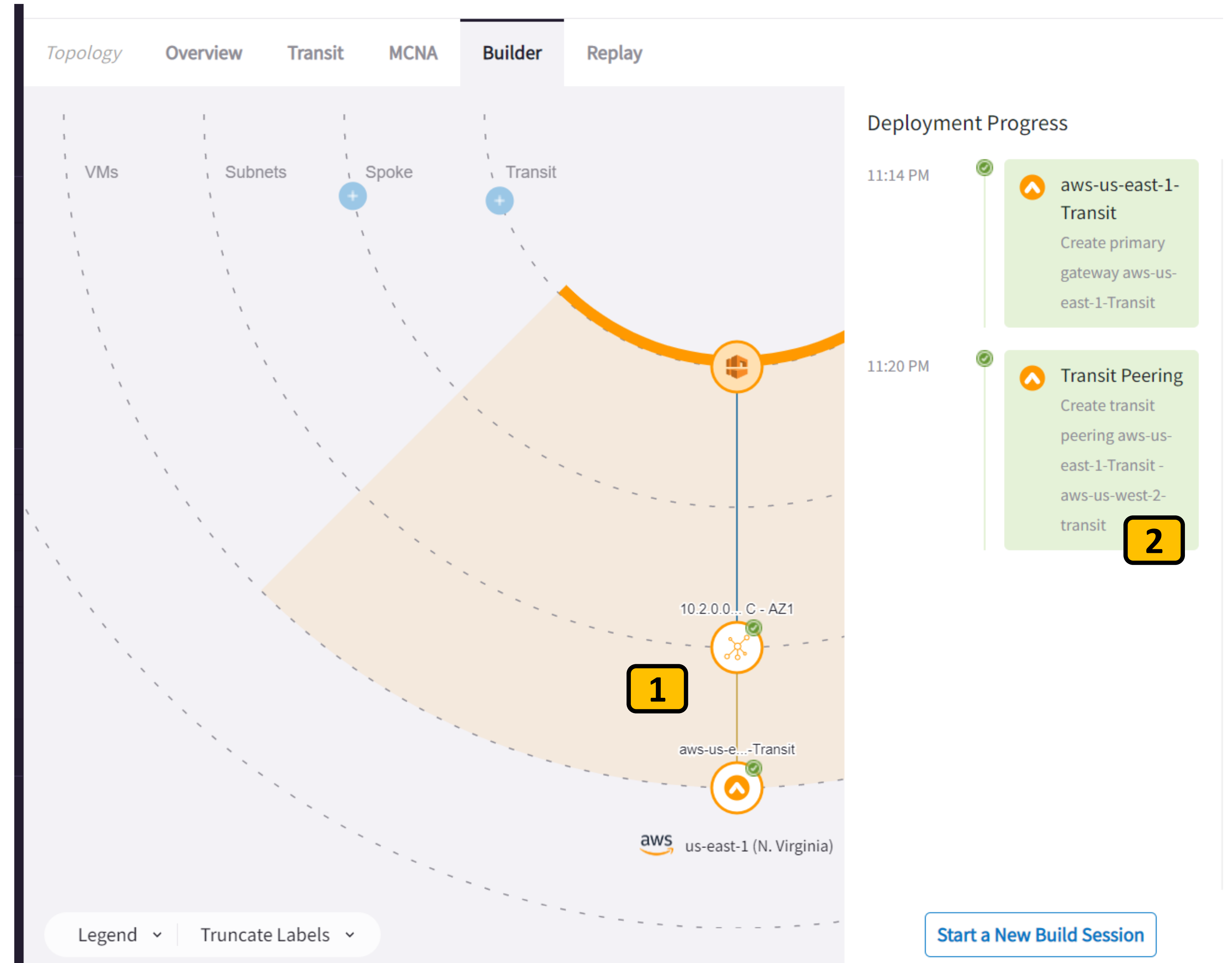


## Lab 2: Cloud Backbone: Step 2.10

Use the Topology Builder to launch an Aviaatrix Transit Gateway in a Transit VPC

Deploy Success will look like this. **1**

You have now successfully created a Transit VPC and Gateway in us-east-1 and peered it to the Transit VPC and Gateway in us-west-2 **2**

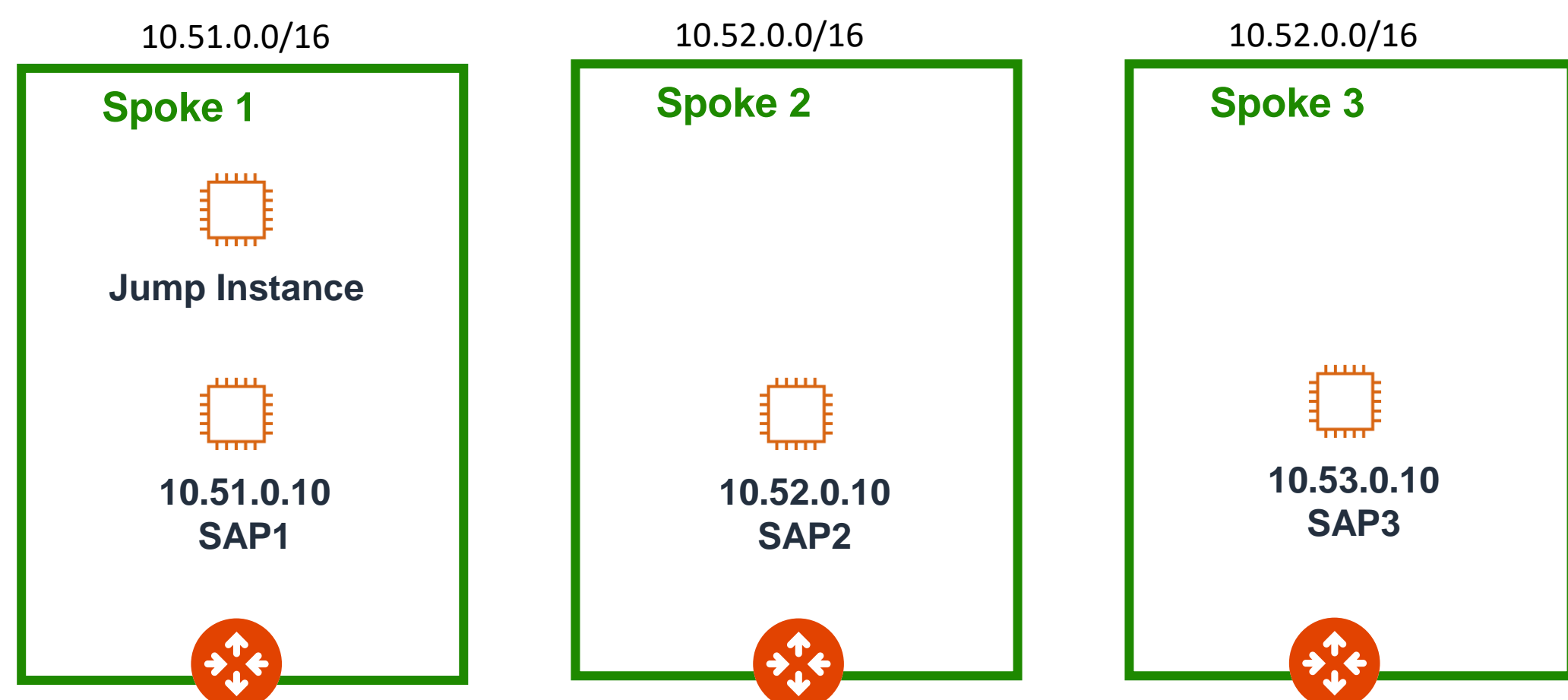




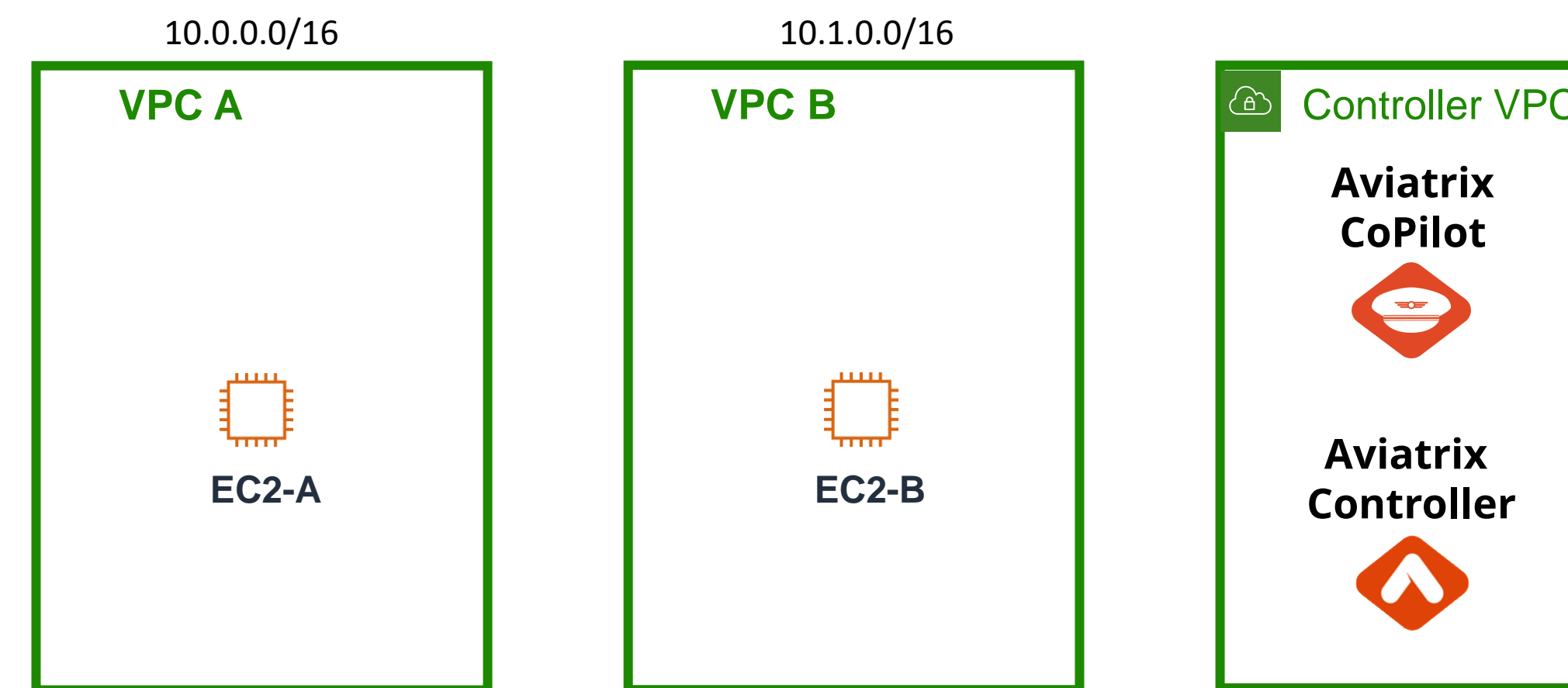
# Lab 2: Cloud Backbone: Progress Check

Your backbone architecture now looks like *this*

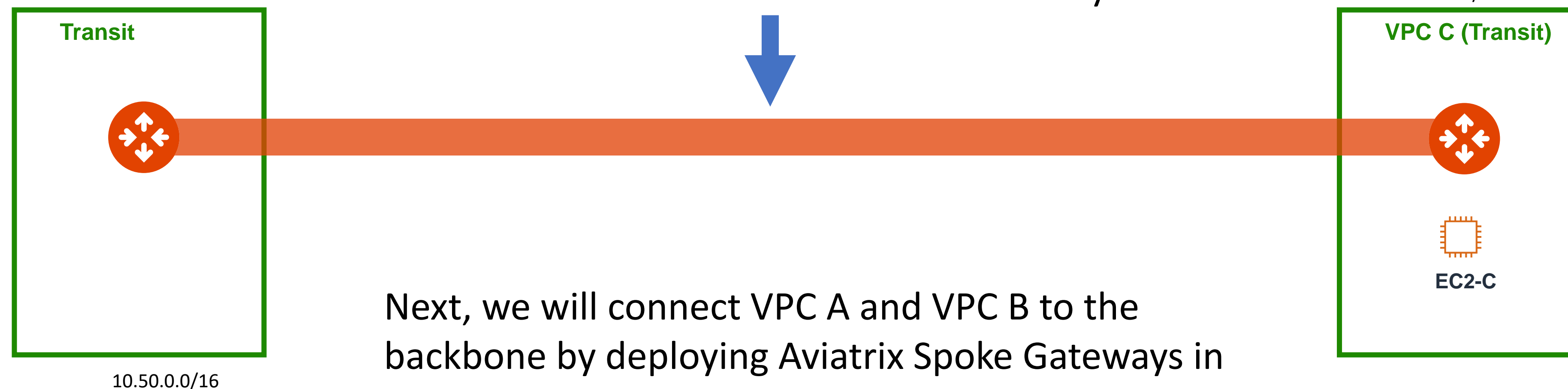
## AWS us-west-2



## AWS us-east-1



You have peered the two AWS regions using transit VPCs and Aviatrix Transit Gateways



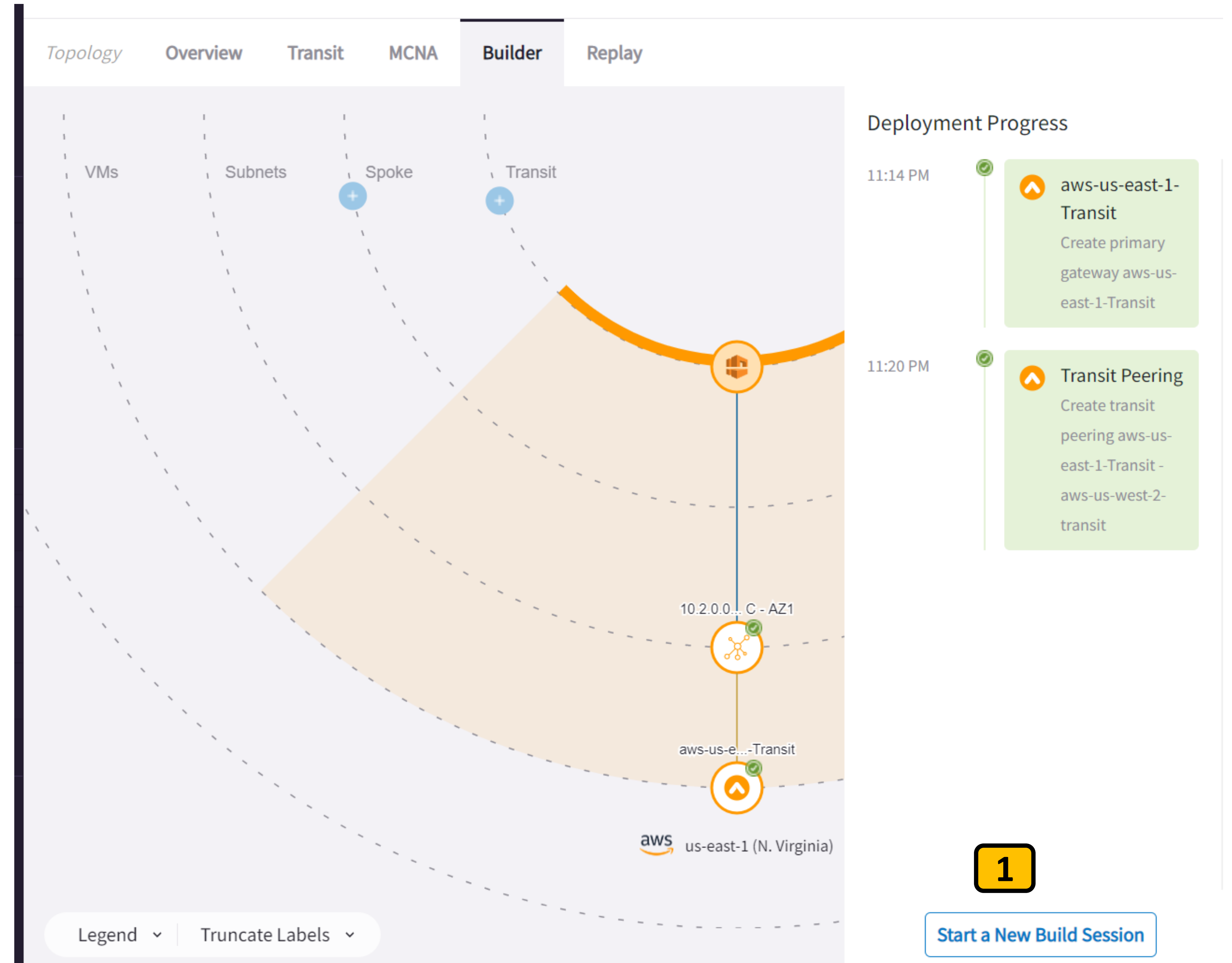
Next, we will connect VPC A and VPC B to the backbone by deploying Aviatrix Spoke Gateways in those VPCs and connecting them to the Aviatrix Transit Gateway in VPC C.

## Lab 2: Cloud Backbone: Step 2.11

Use the Topology Builder to launch Aviatrix Spoke Gateways and connect them to Transit

Now let's deploy Aviatrix Spoke Gateways in VPC A and VPC B using the same Topology Builder tool.

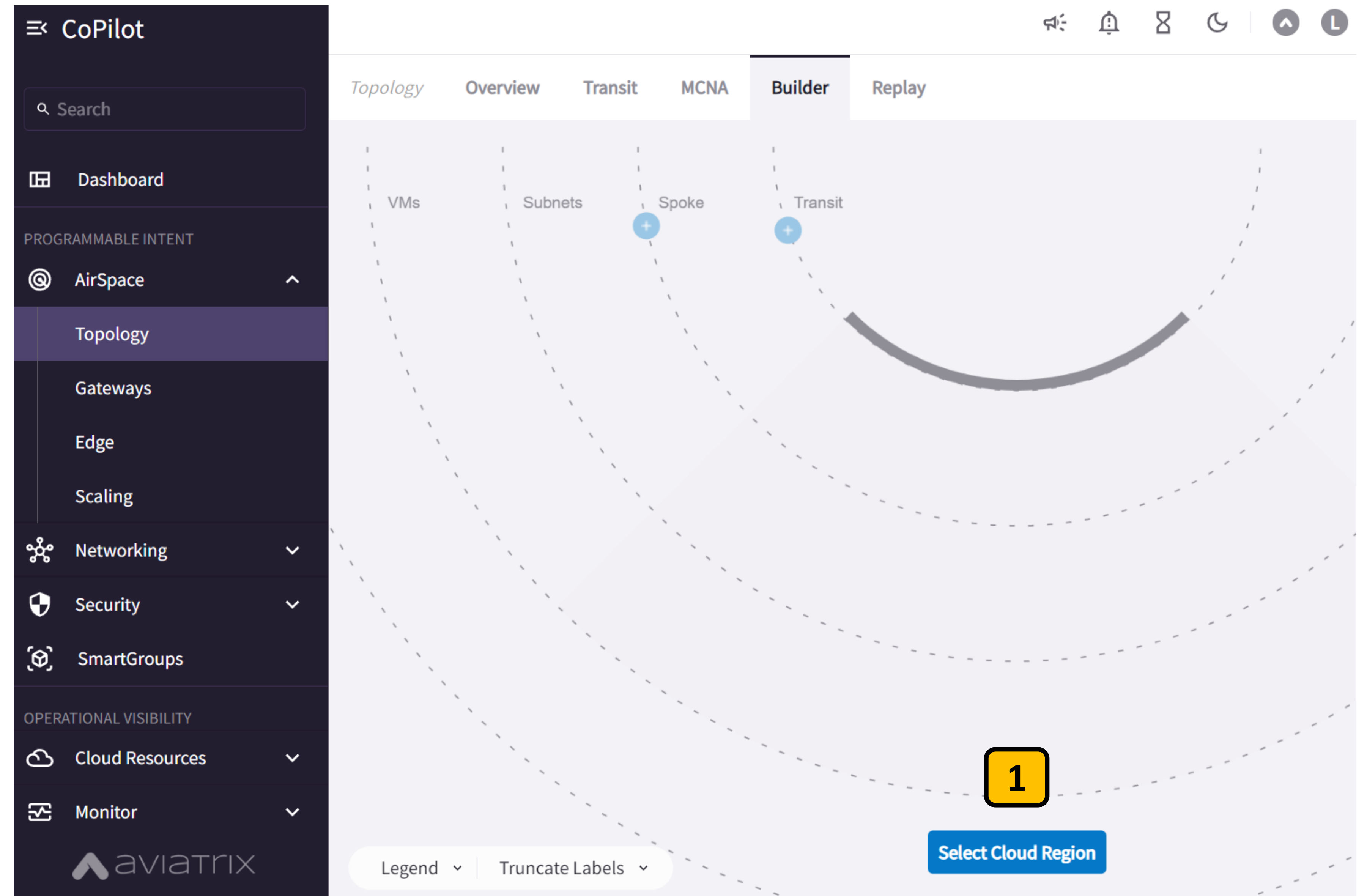
Click Start a New Build Session **1**



## Lab 2: Cloud Backbone: Step 2.12

Use the Topology Builder to launch Aviaatrix Spoke Gateways and connect them to Transit

In the Topology Builder,  
click **Select Cloud Region** 1



## Lab 2: Cloud Backbone: Step 2.13

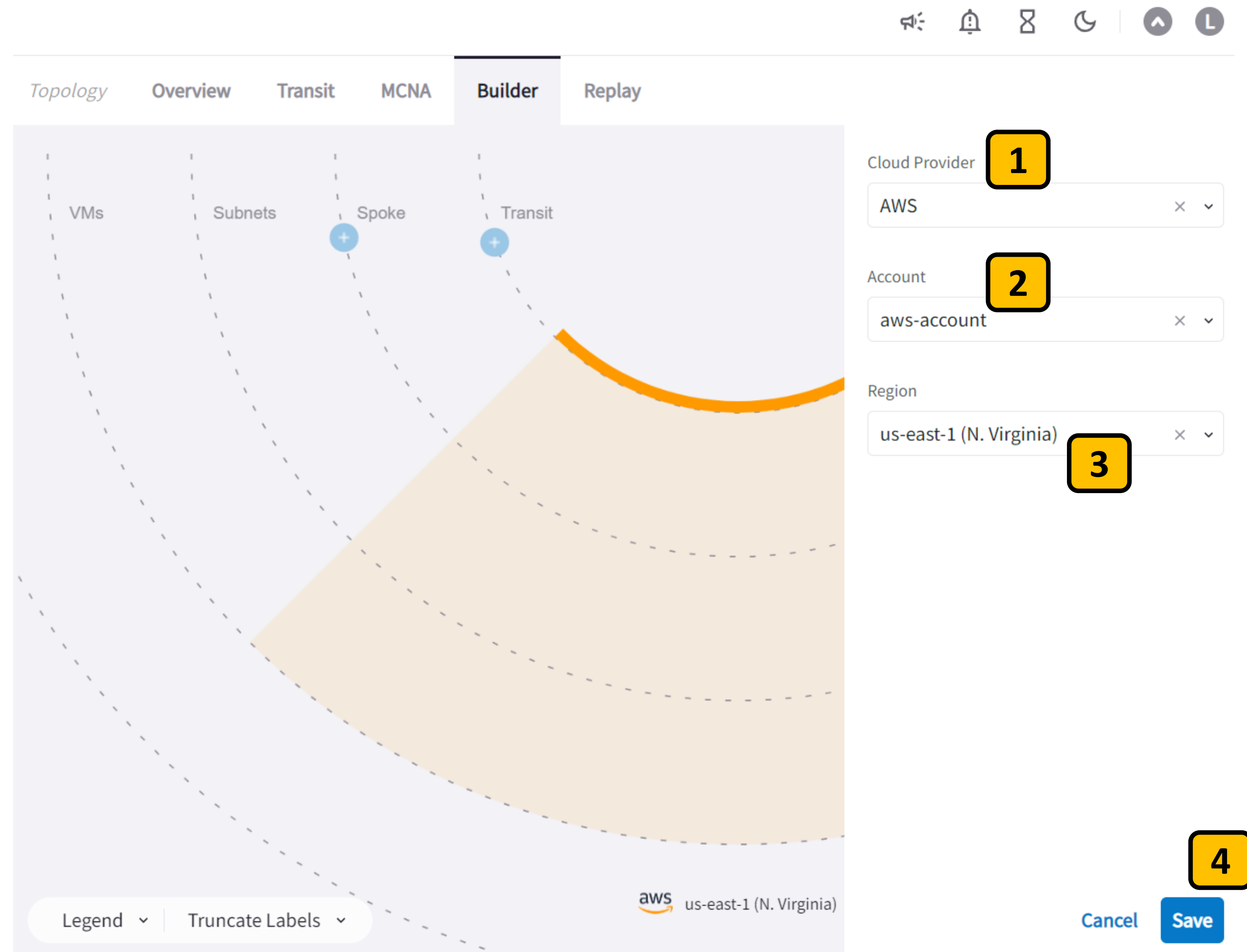
Use the Topology Builder to launch Aviaatrix Spoke Gateways and connect them to Transit

Select the Cloud Provider: **AWS** **1**

Select the AWS Account: **aws-account** **2**

Select the Region: **us-east-1** **3**

Click **Save** to start a new Build session in this cloud, account, and region. **4**



The screenshot shows the Aviaatrix Topology Builder interface. The 'Builder' tab is selected, and a diagram shows a 'Spoke' and a 'Transit' node connected by a dashed line. The right sidebar contains configuration options for the build session:

- Cloud Provider:** AWS (Step 1)
- Account:** aws-account (Step 2)
- Region:** us-east-1 (N. Virginia) (Step 3)

At the bottom right, there are 'Cancel' and 'Save' buttons (Step 4). The bottom of the interface shows the 'aws us-east-1 (N. Virginia)' logo and a 'Legend' dropdown.



## Lab 2: Cloud Backbone: Step 2.14

Use the Topology Builder to launch Aviatrix Spoke Gateways and connect them to Transit

Click the **blue +** under Spoke **1**

Select **VPC A** to be the Spoke VPC **2**

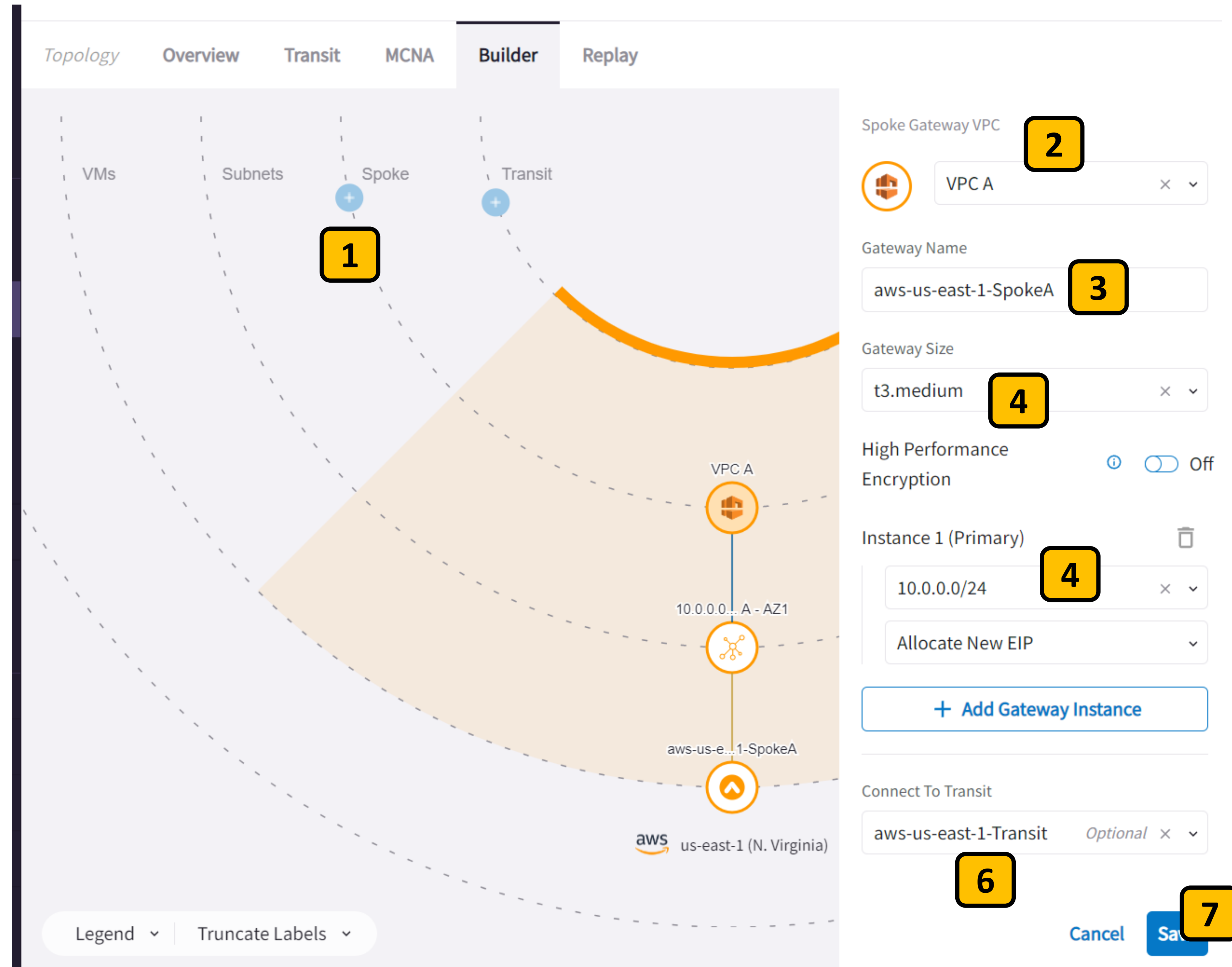
Name the Spoke gateway:  
**aws-us-east-1-SpokeA** **3**

Select **t3.medium** for the gateway  
instance size **4**

Select the 10.0.0.0/24 as the public  
subnet to deploy the gateway in to **5**

Click Connect to Transit and select the  
aws-us-east-1-Transit. **6**

Click Save **7**



The screenshot shows the Aviatrix Topology Builder interface with the 'Builder' tab selected. The main canvas displays a network topology with a 'Spoke' and a 'Transit' component. A yellow callout box highlights the 'Spoke' component, and a blue '+' icon is visible next to it, indicating where to click to add a new spoke gateway.

On the right side, the configuration panel for the 'Spoke Gateway VPC' is shown. The configuration includes:

- Spoke Gateway VPC:** VPC A (selected)
- Gateway Name:** aws-us-east-1-SpokeA
- Gateway Size:** t3.medium
- High Performance Encryption:** Off
- Instance 1 (Primary):** 10.0.0.0/24 (selected), Allocate New EIP
- Connect To Transit:** aws-us-east-1-Transit (Optional)

At the bottom of the configuration panel, there is a '+ Add Gateway Instance' button. The bottom of the interface shows a 'Legend' and 'Truncate Labels' dropdown menu. The bottom right corner has 'Cancel' and 'Save' buttons.

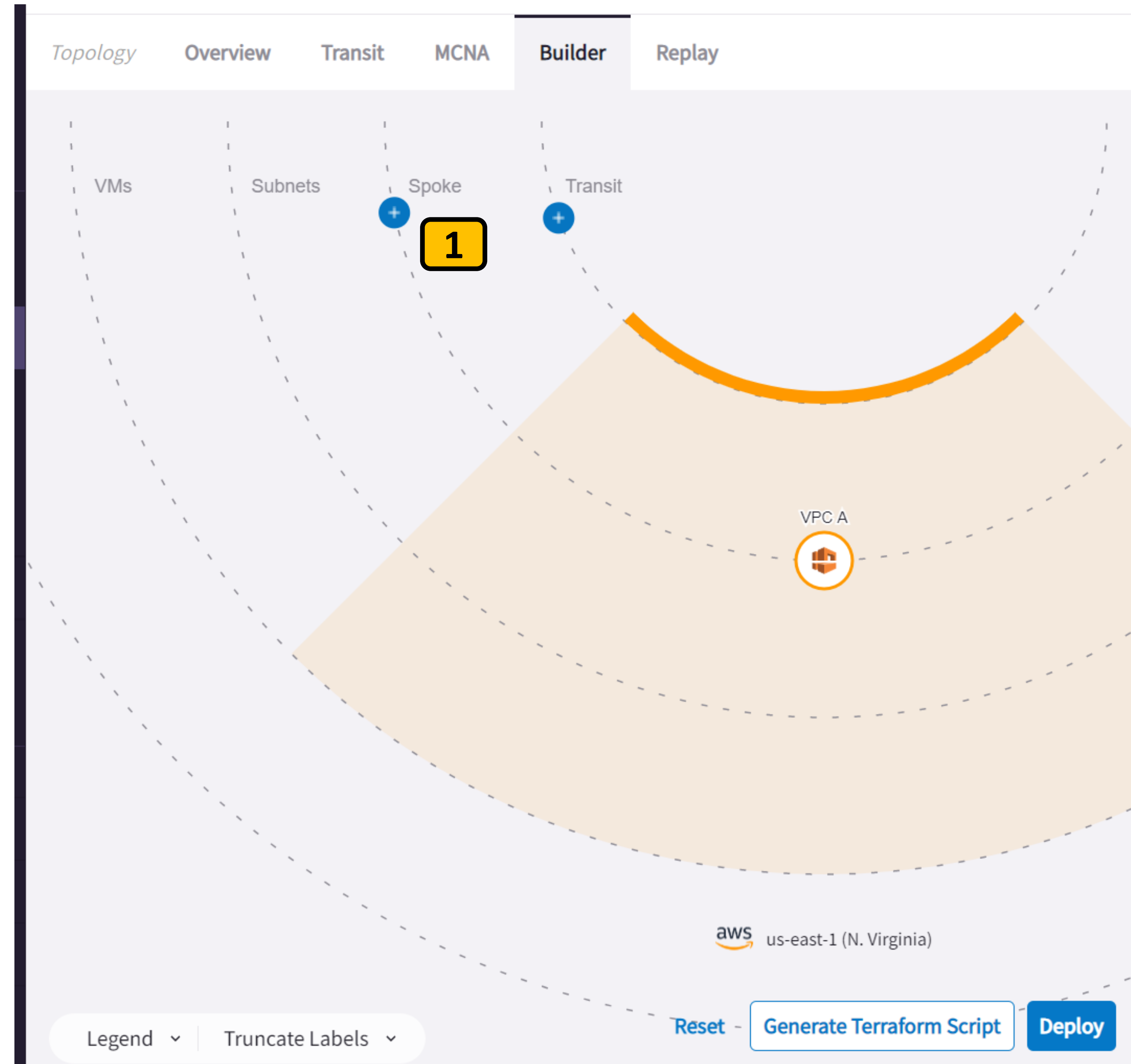
## Lab 2: Cloud Backbone: Step 2.15

Use the Topology Builder to launch Aviaatrix Spoke Gateways and connect them to Transit

**Don't click Deploy yet!**

Let's add VPC B to this build session to get another Spoke Gateway deployed all in the same build session.

Click the blue+ under Spoke



## Lab 2: Cloud Backbone: Step 2.16

Use the Topology Builder to launch Aviatrix Spoke Gateways and connect them to Transit

Select **VPC B** to be a Spoke VPC **1**

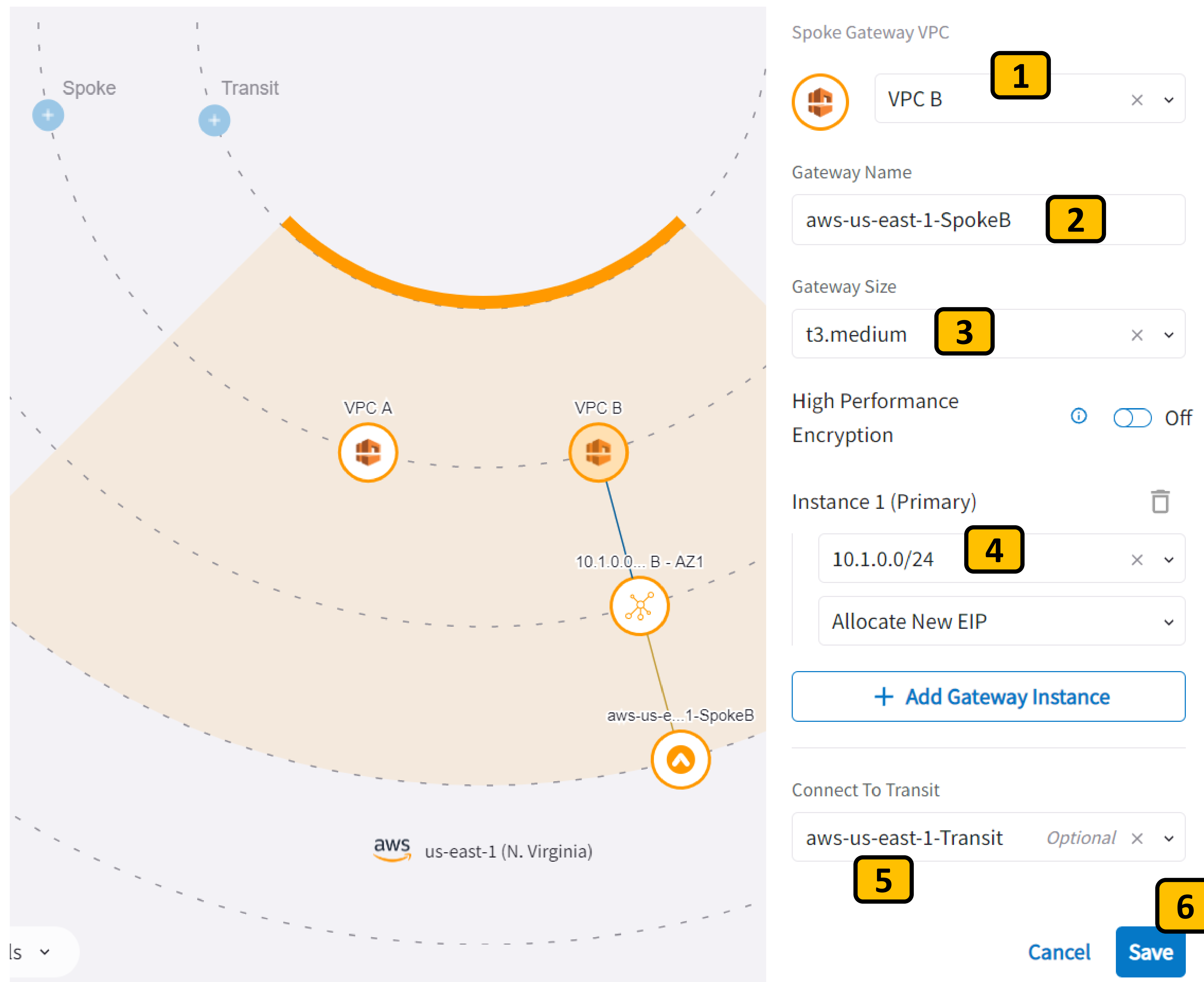
Name the Spoke gateway:  
**aws-us-east-1-SpokeB** **2**

Select **t3.medium** for the gateway  
instance size **3**

Select the 10.1.0.0/24 as the public  
subnet to deploy the gateway in to **4**

Click Connect to Transit and select the  
aws-us-east-1-Transit. **5**

Click Save **6**



The screenshot shows the Aviatrix Topology Builder interface. On the left, a diagram illustrates the network topology with VPC A and VPC B connected to a Transit gateway. VPC B is highlighted as a Spoke VPC. On the right, the configuration panel for the Spoke Gateway VPC is shown, with numbered steps 1 through 6 indicating the configuration process.


**Spoke Gateway VPC**

- 1** VPC B
- 2** Gateway Name: aws-us-east-1-SpokeB
- 3** Gateway Size: t3.medium
- 4** Instance 1 (Primary): 10.1.0.0/24
- 5** Connect To Transit: aws-us-east-1-Transit
- 6** Save

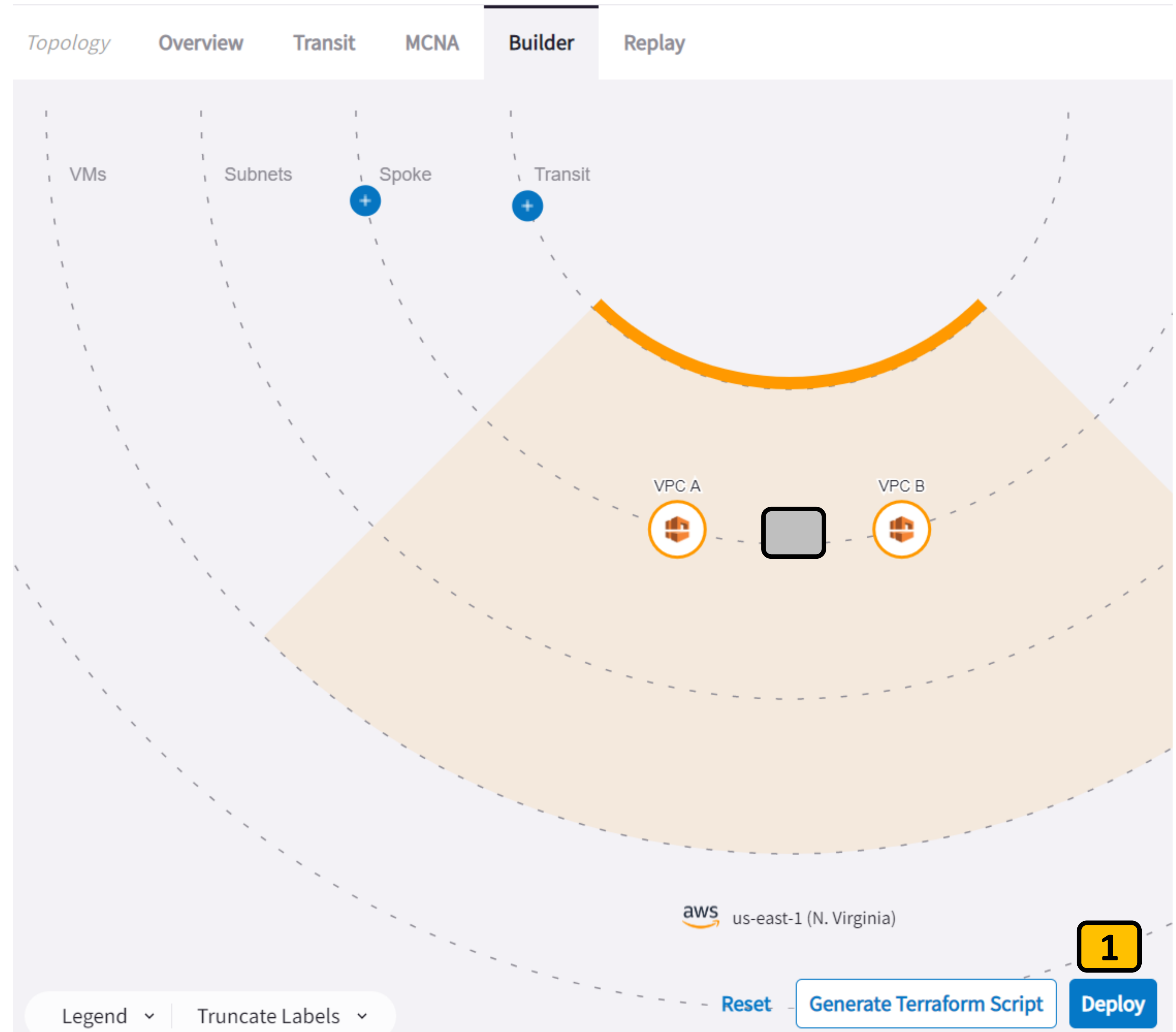
Additional configuration options visible include High Performance Encryption (Off) and a button to Add Gateway Instance.

## Lab 2: Cloud Backbone: Step 2.17

Use the Topology Builder to launch Aviatrix Spoke Gateways and connect them to Transit

**Optional:** If you want to double-check or change details of your Spoke gateway deployment settings, click the VPC A or VPC B icons 

Click Deploy to begin your Spoke VPC deployment in us-east-1 **1**





## Lab 2: Cloud Backbone: Step 2.18

Use the Topology Builder to launch Aviatrix Spoke Gateways and connect them to Transit

Observe the Deployment Progress of your Aviatrix Spoke Gateways in us-east-1. **1**

After the Spoke gateways are deployed, CoPilot will connect them to the Aviatrix Transit gateway you created earlier. **2**

The total deployment time should take about 5 minutes.



## Lab 2: Cloud Backbone: Step 2.19

Enable Connected Transit

Go to AirSpace > Gateways > Transit Gateways

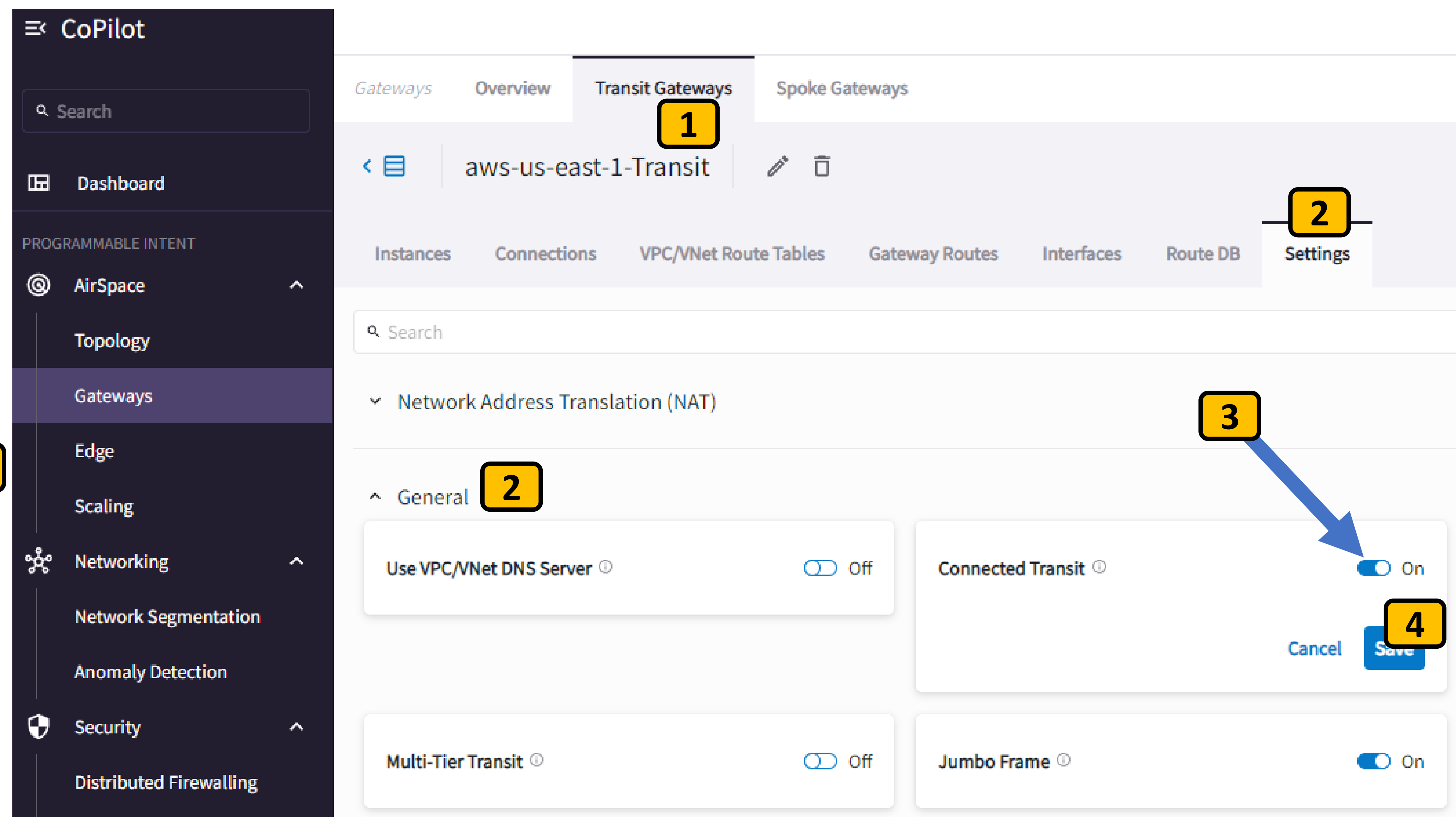
Click on the Transit Gateway **aws-us-east-1-Transit** **1**

Select **Settings** **2**

Toggle **Connected Transit** to On **3**

Click **Save** **4**

This tells the Transit Gateway that it can forward traffic between Spoke Gateways

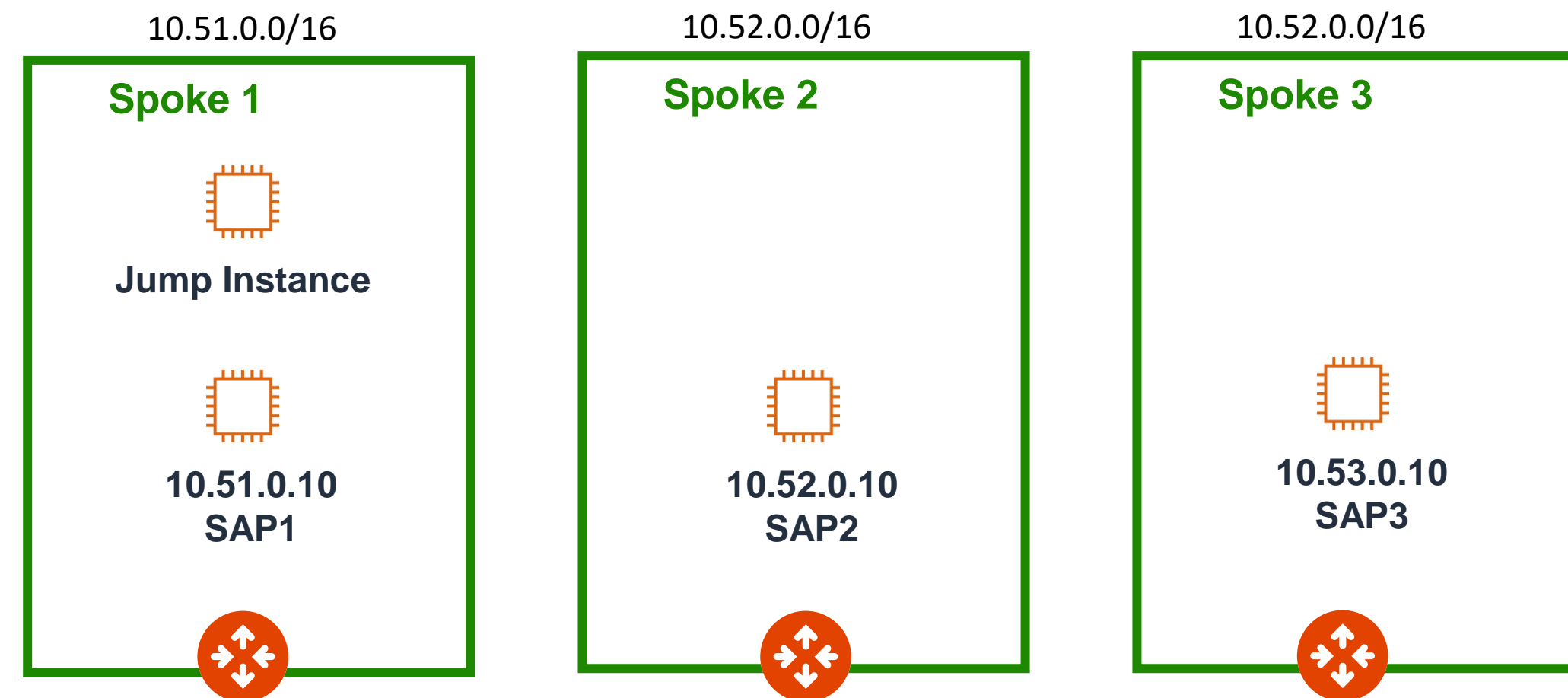


The screenshot shows the AWS CloudFormation console interface for managing a Transit Gateway. The left sidebar, labeled 'CoPilot', contains a search bar and a navigation menu with options like Dashboard, AirSpace, Topology, Gateways, Edge, Scaling, Networking, Network Segmentation, Anomaly Detection, Security, and Distributed Firewalling. The main content area has a top navigation bar with tabs for Gateways, Overview, Transit Gateways (selected), and Spoke Gateways. Below this, there's a breadcrumb trail showing 'aws-us-east-1-Transit' and a list of tabs: Instances, Connections, VPC/VNet Route Tables, Gateway Routes, Interfaces, Route DB, and Settings (selected). The Settings tab displays a search bar and a section for 'Network Address Translation (NAT)'. Under the 'General' section, there are two toggle switches: 'Use VPC/VNet DNS Server' (Off) and 'Connected Transit' (On). A blue arrow points from the '3' label to the 'Connected Transit' toggle. At the bottom right, there are 'Cancel' and 'Save' buttons, with the 'Save' button highlighted by a blue box and labeled '4'.

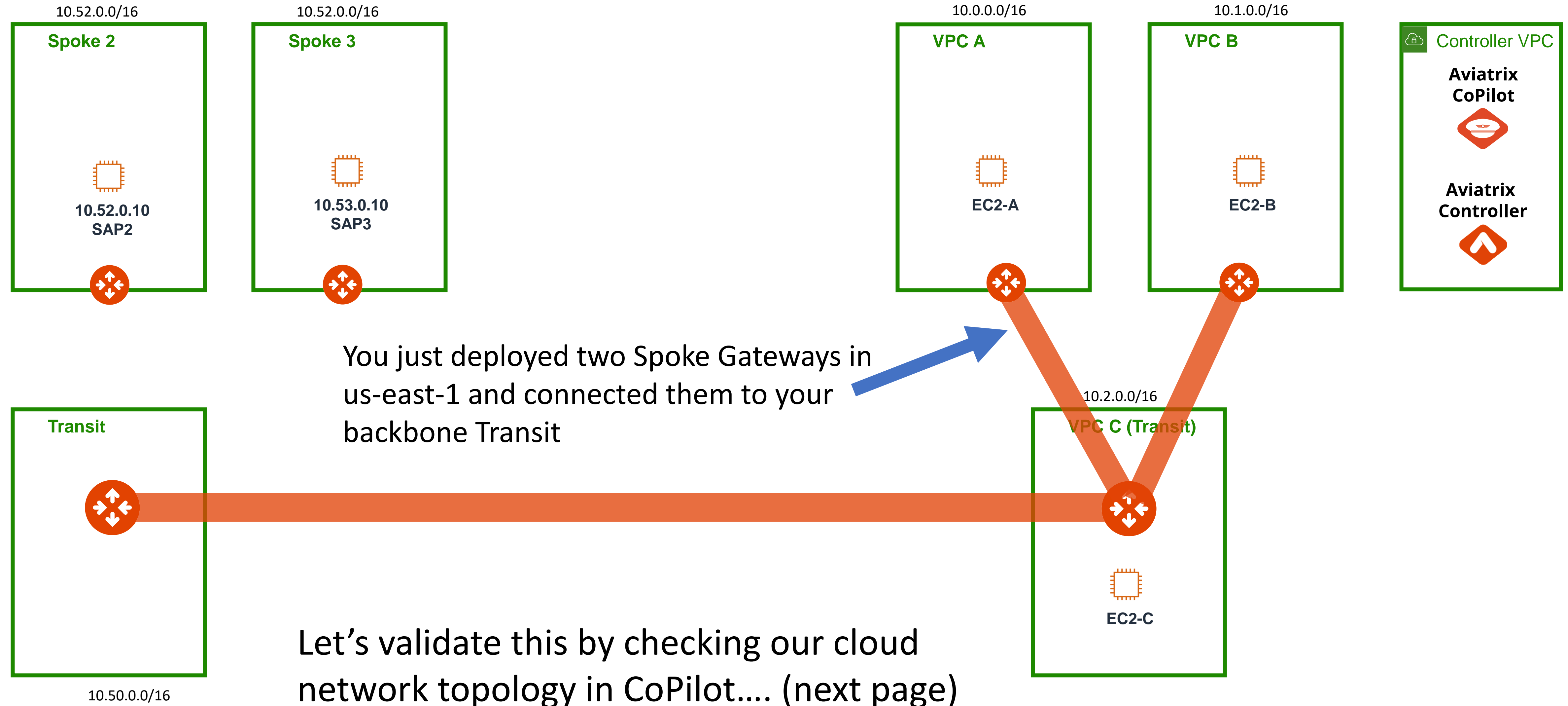
# Lab 2: Cloud Backbone: Progress Check

Your backbone architecture now looks like *this*

## AWS us-west-2



## AWS us-east-1





## Lab 2: Cloud Backbone: Step 2.20

Check your cloud network topology in Aviatrix CoPilot

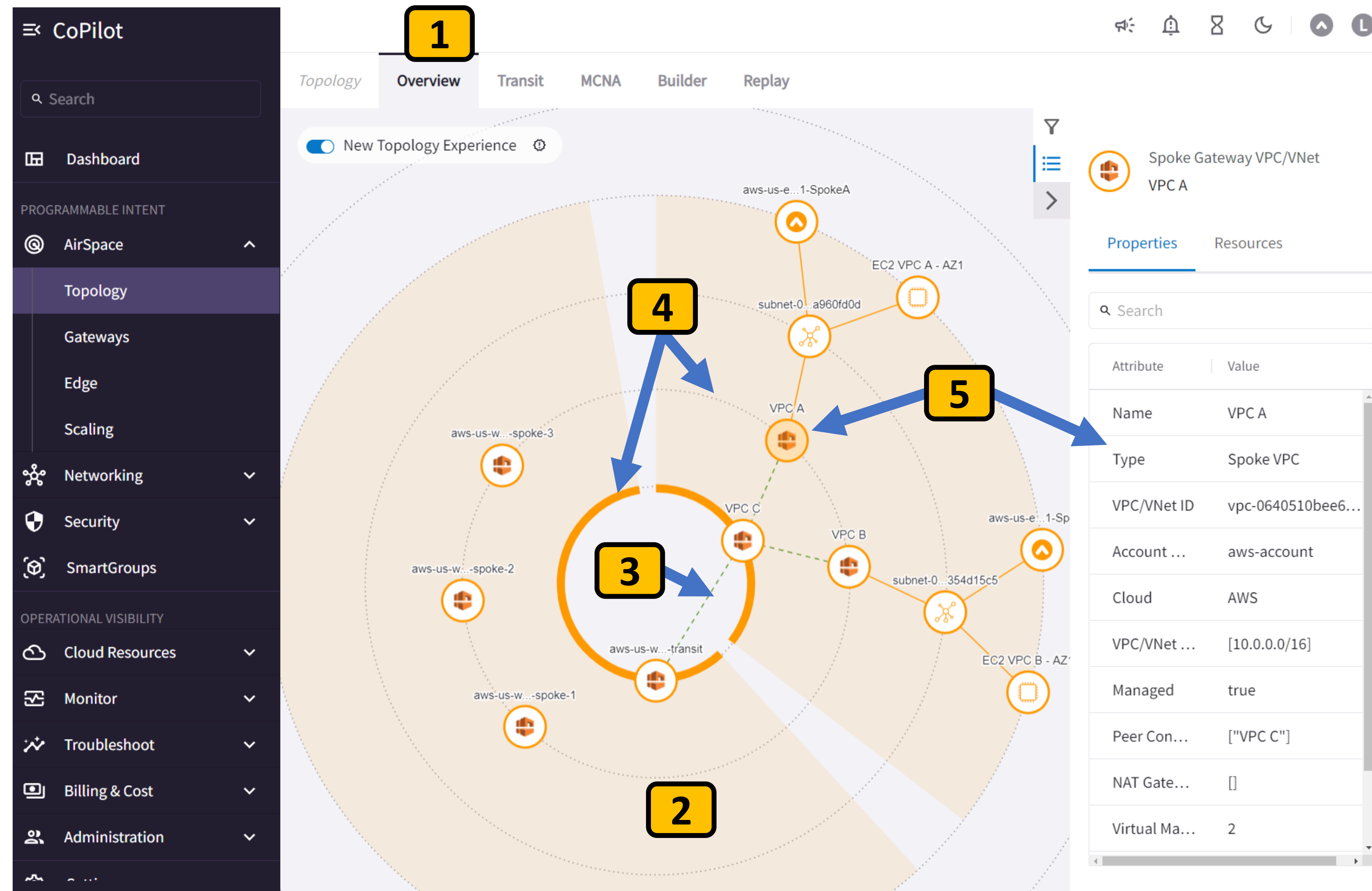
Go to AirSpace > Topology > Overview **1**

Each piece of orange pie represents an AWS region and the resources inside it **2**

Your backbone is represented by the dashed line **3**

The inner most circle of your topology represents Transit VPCs. The second circle represents Spoke VPCs. **4**

Click on any VPC to expand (or hide) its contents and see details in the right-side panel. **5**



**NOTE:** If you don't see your EC2 instances from Lab 1 in your topology, see the next slide for steps to resolve that.



## Lab 2: Cloud Backbone: Step 2.20b

Set your Fetch Instance task frequency in CoPilot

If you are not able to see your EC2 instances from Lab 1 in the CoPilot topology above...

Navigate to Settings **1**

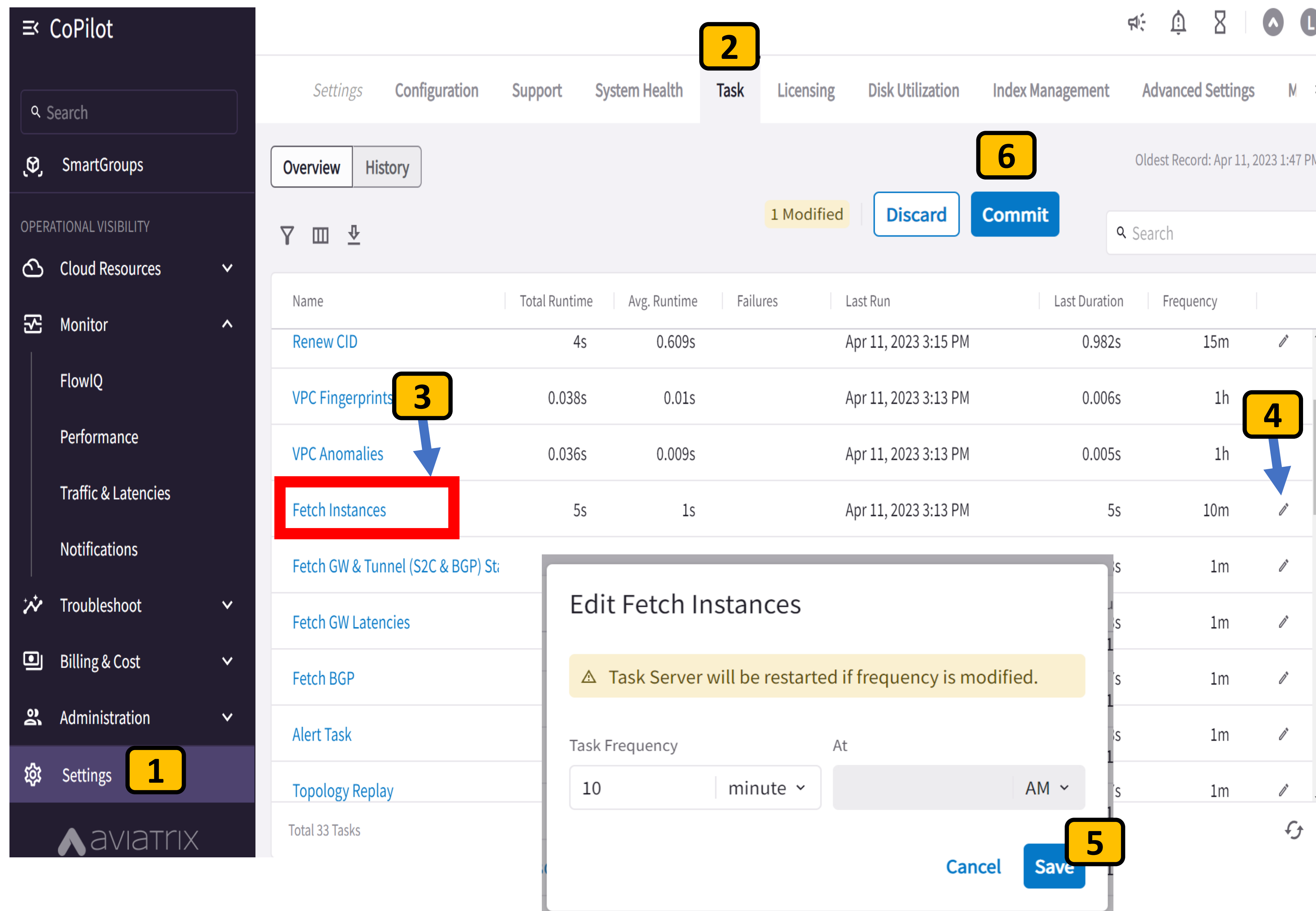
Select Task **2**

Find the Fetch Instances task **3**

Click the Pencil to edit this task **4**

Set the task to run every 10 minutes and click Save **5**

Be sure to Commit your change **6**



The screenshot shows the AviaMatrix CoPilot interface. On the left is a dark sidebar with a 'Settings' option at the bottom, marked with a yellow box labeled '1'. The main panel has a top navigation bar with 'Task' selected, marked with a yellow box labeled '2'. Below this is a table of tasks. The 'Fetch Instances' task is highlighted with a red box, marked with a yellow box labeled '3'. A blue arrow points from the pencil icon in the 'Fetch Instances' row to an 'Edit Fetch Instances' modal window, marked with a yellow box labeled '4'. The modal window shows a 'Task Frequency' of '10' minutes, marked with a yellow box labeled '5'. At the bottom of the modal are 'Cancel' and 'Save' buttons. In the background, the 'Task' tab shows a 'Commit' button, marked with a yellow box labeled '6'.

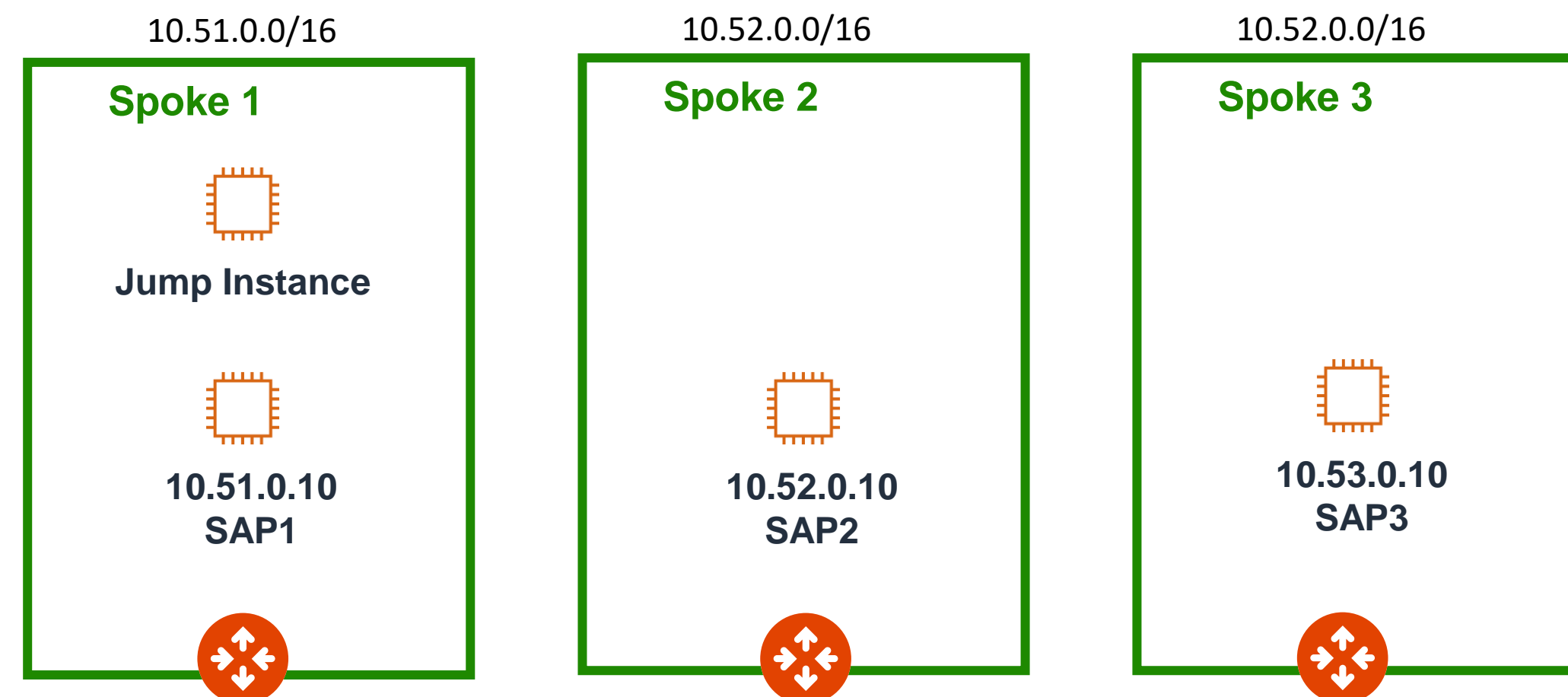
Name	Total Runtime	Avg. Runtime	Failures	Last Run	Last Duration	Frequency
Renew CID	4s	0.609s		Apr 11, 2023 3:15 PM	0.982s	15m
VPC Fingerprints	0.038s	0.01s		Apr 11, 2023 3:13 PM	0.006s	1h
VPC Anomalies	0.036s	0.009s		Apr 11, 2023 3:13 PM	0.005s	1h
Fetch Instances	5s	1s		Apr 11, 2023 3:13 PM	5s	10m
Fetch GW & Tunnel (S2C & BGP) St						1m
Fetch GW Latencies						1m
Fetch BGP						1m
Alert Task						1m
Topology Replay						1m

This will cause CoPilot to fetch your instances every 10 minutes, starting Now.

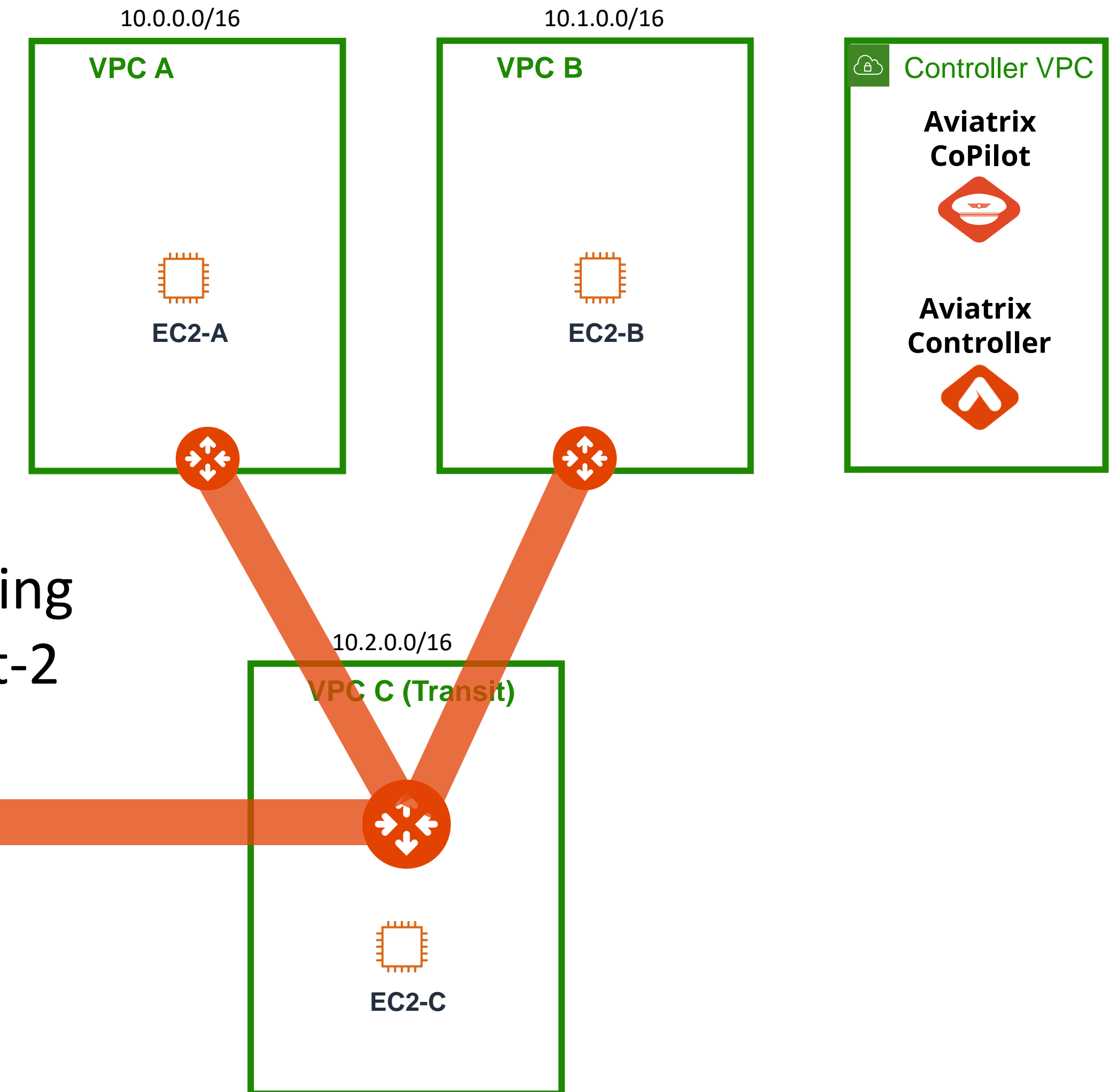
# Lab 2: Cloud Backbone: Progress Check

Your backbone architecture now looks like *this*

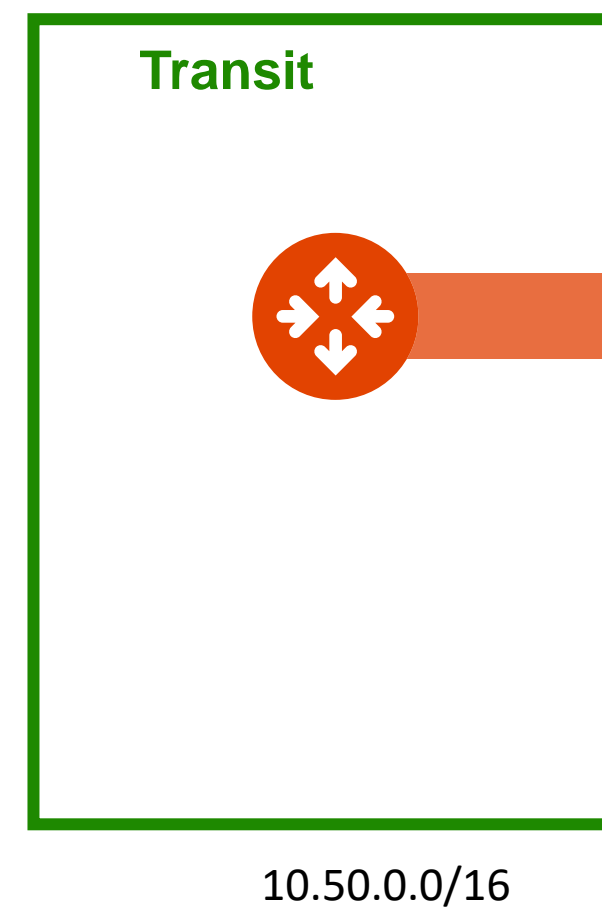
## AWS us-west-2



## AWS us-east-1

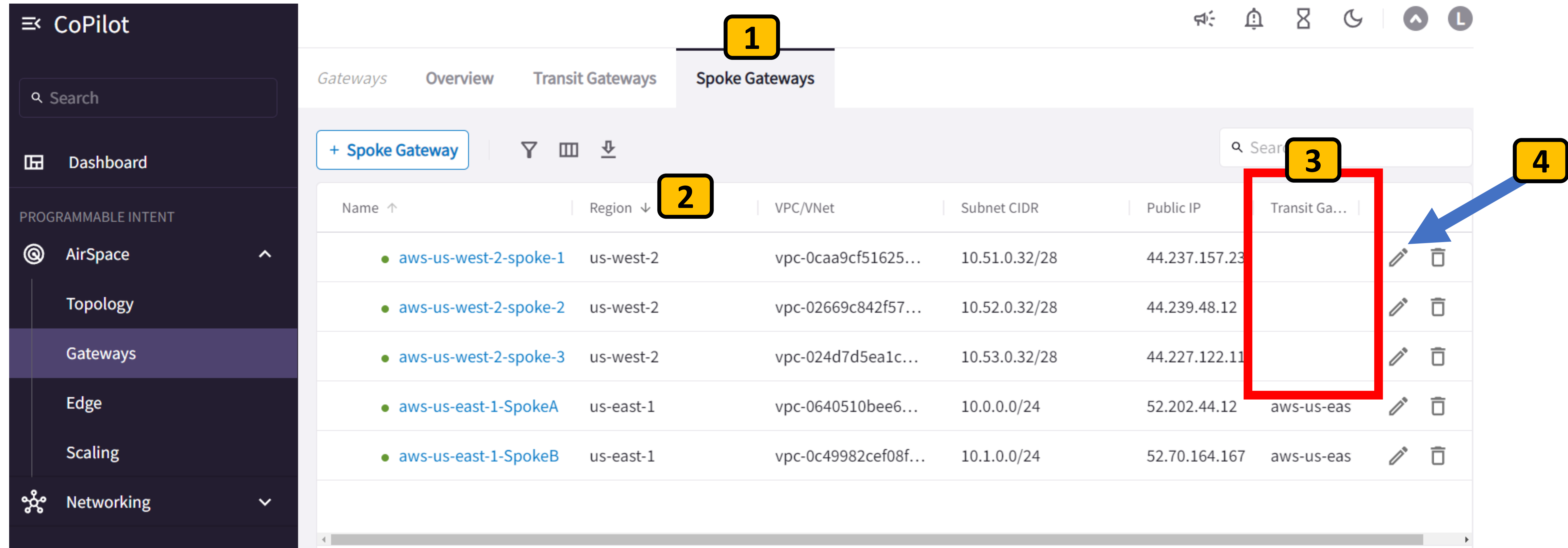


Next, we will connect the pre-existing Aviatrix Spoke Gateways in us-west-2 to our cloud backbone.



# Lab 2: Cloud Backbone: Step 2.21

Connect Spoke Gateways to Transit using CoPilot



Name ↑	Region ↓	VPC/VNet	Subnet CIDR	Public IP	Transit Ga...
aws-us-west-2-spoke-1	us-west-2	vpc-0caa9cf51625...	10.51.0.32/28	44.237.157.23	
aws-us-west-2-spoke-2	us-west-2	vpc-02669c842f57...	10.52.0.32/28	44.239.48.12	
aws-us-west-2-spoke-3	us-west-2	vpc-024d7d5ea1c...	10.53.0.32/28	44.227.122.11	
aws-us-east-1-SpokeA	us-east-1	vpc-0640510bee6...	10.0.0.0/24	52.202.44.12	aws-us-eas
aws-us-east-1-SpokeB	us-east-1	vpc-0c49982cef08f...	10.1.0.0/24	52.70.164.167	aws-us-eas

Go to AirSpace > Gateways > Spoke Gateways **1**

Sort by Region **2**

Select the edit Pencil for the Spoke Gateway aws-us-west-2-spoke-1 **4**

Notice your us-west-2 Spoke gateways are not connected to a Transit **3**



## Lab 2: Cloud Backbone: Step 2.22

Connect Spoke Gateways to Transit using CoPilot

On the Edit pop-up window select:  
Attach to Transit Gateway **1**

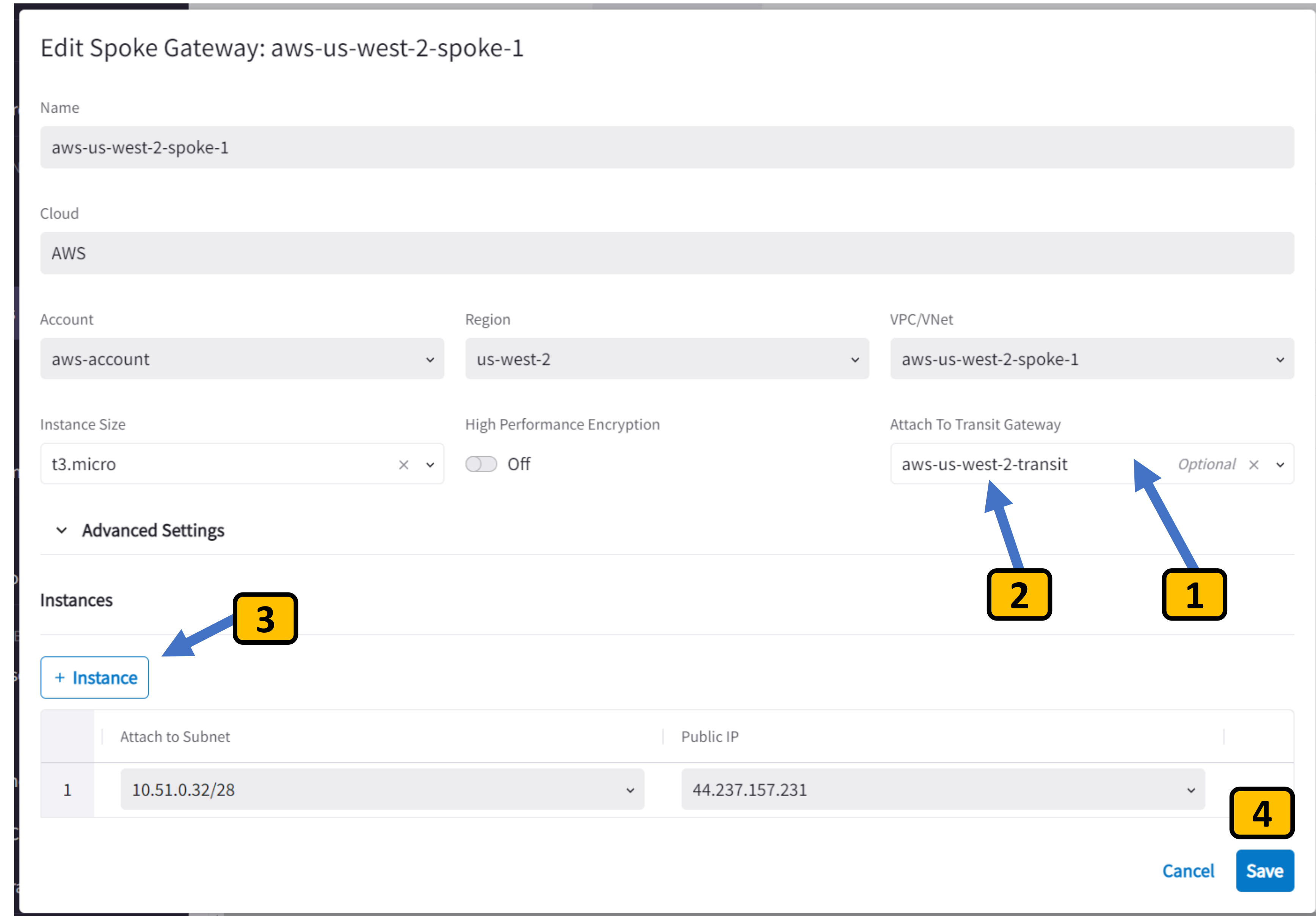
Select the **aws-us-west-2-transit** **2**

FYI: If you wanted to add an active-active HA Spoke Gateway, you can do that here. **3**

(don't do that now as our lab EIP limits are already used up)

Click Save **4**

Repeat this process for the remaining Spoke Gateways in us-west-2



Edit Spoke Gateway: aws-us-west-2-spoke-1

Name: aws-us-west-2-spoke-1

Cloud: AWS

Account: aws-account

Region: us-west-2

VPC/VNet: aws-us-west-2-spoke-1

Instance Size: t3.micro

High Performance Encryption: Off

Attach To Transit Gateway: aws-us-west-2-transit

Advanced Settings

Instances

	Attach to Subnet	Public IP
1	10.51.0.32/28	44.237.157.231

Cancel Save

Easy, right?

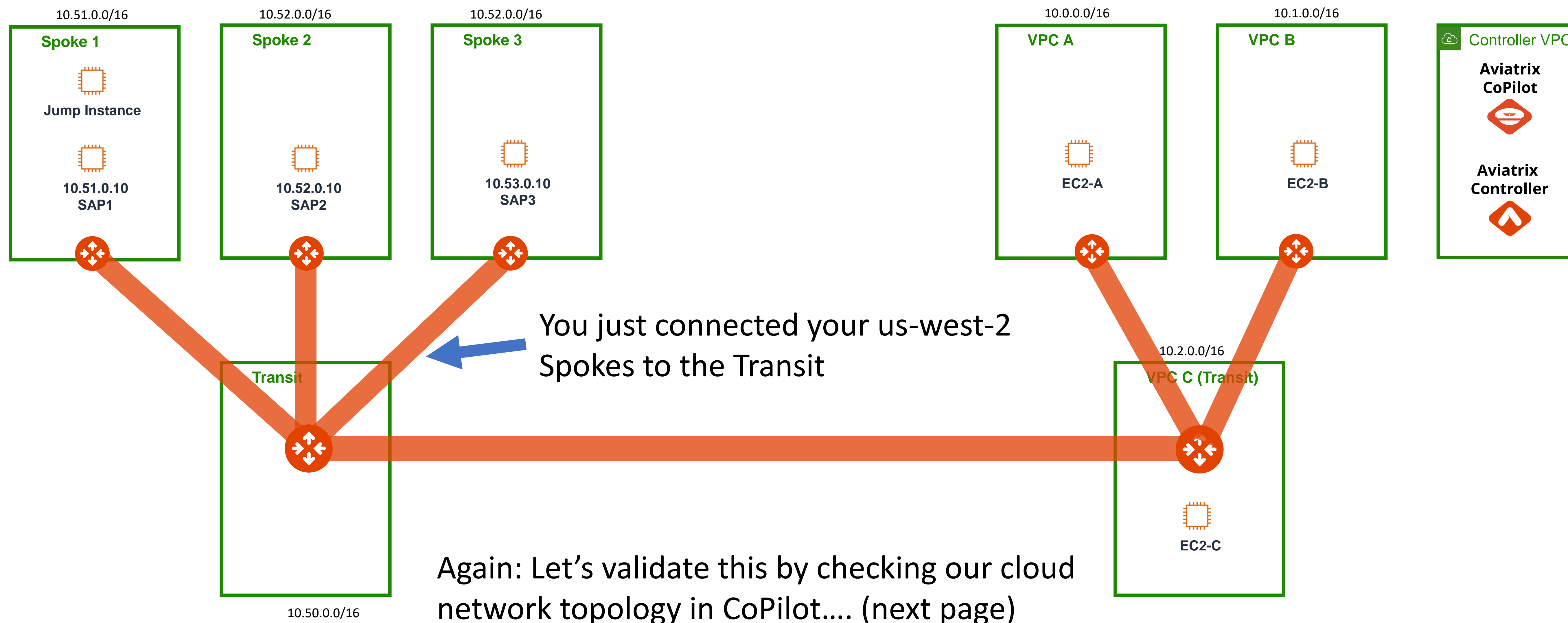


# Lab 2: Cloud Backbone: Progress Check

Your backbone architecture now looks like *this*

## AWS us-west-2

## AWS us-east-1



# Lab 2: Cloud Backbone: Step 2.23

Check your cloud network topology in CoPilot

Go to AirSpace > Topology > Overview

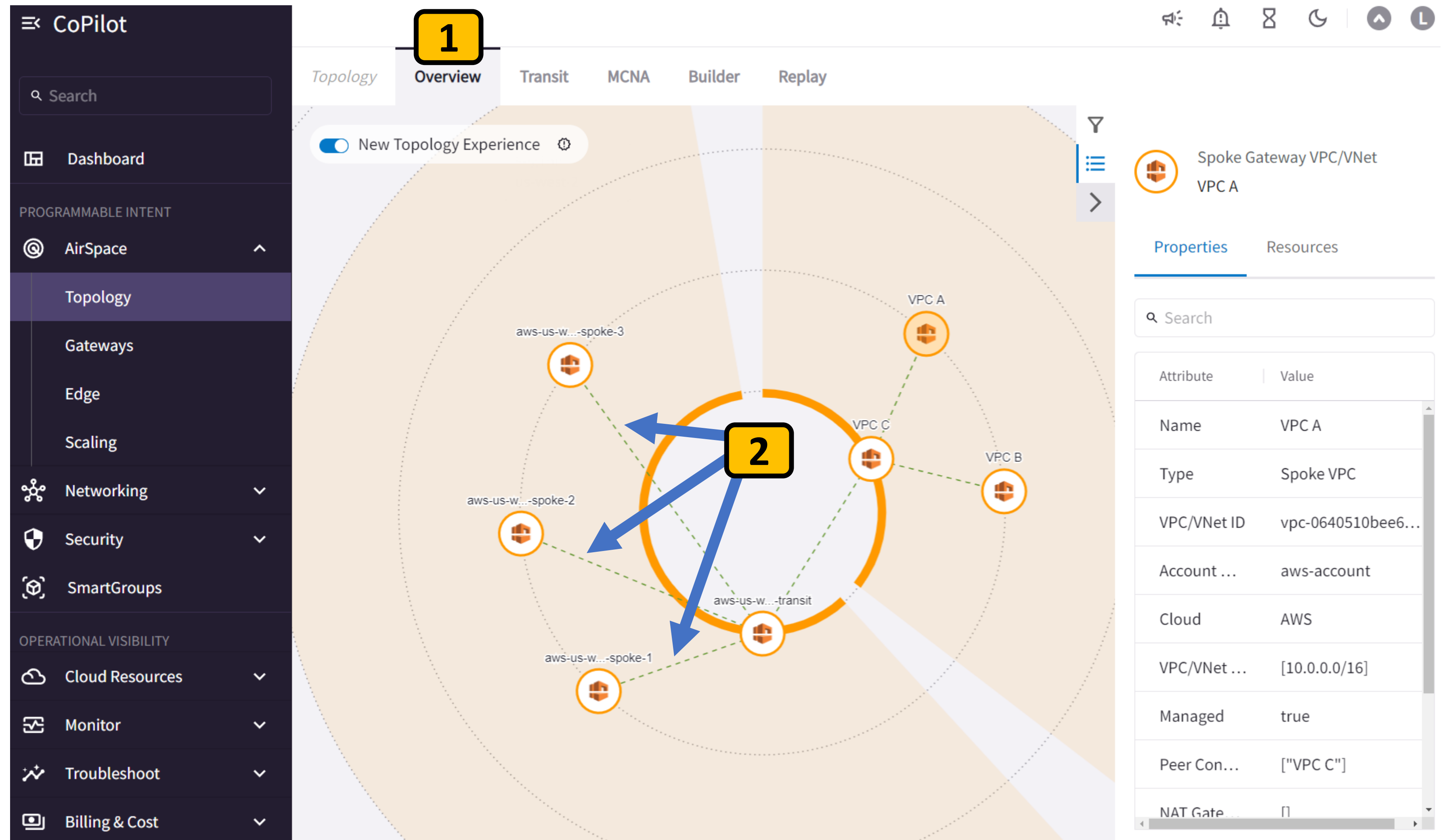
1

Notice your us-west-2 Spokes are now connect to your Transit in us-west-2

2

**Success!** We have our multi-region cloud network backbone!

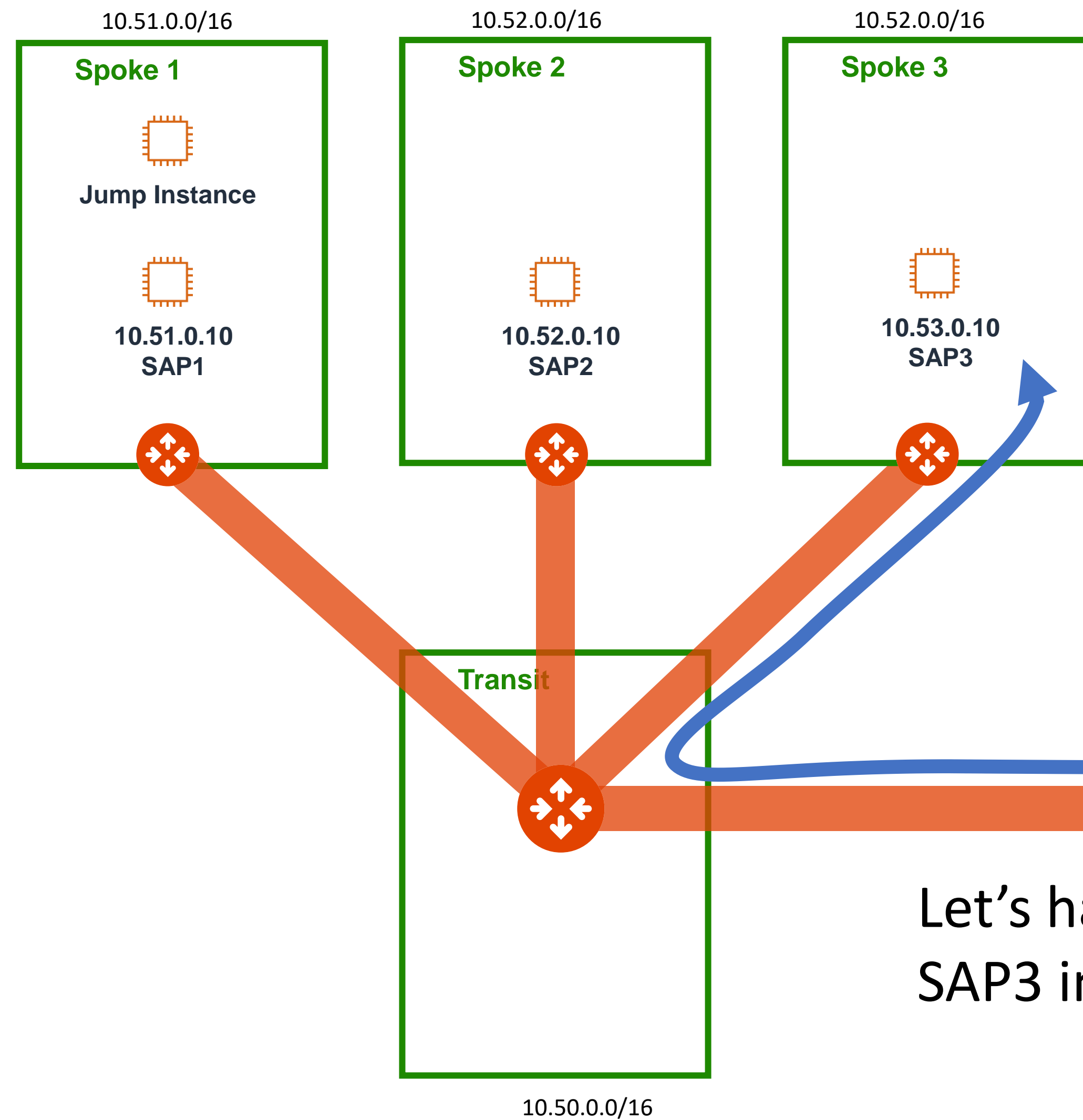
Now let's test the connectivity and observe network traffic ...(next)



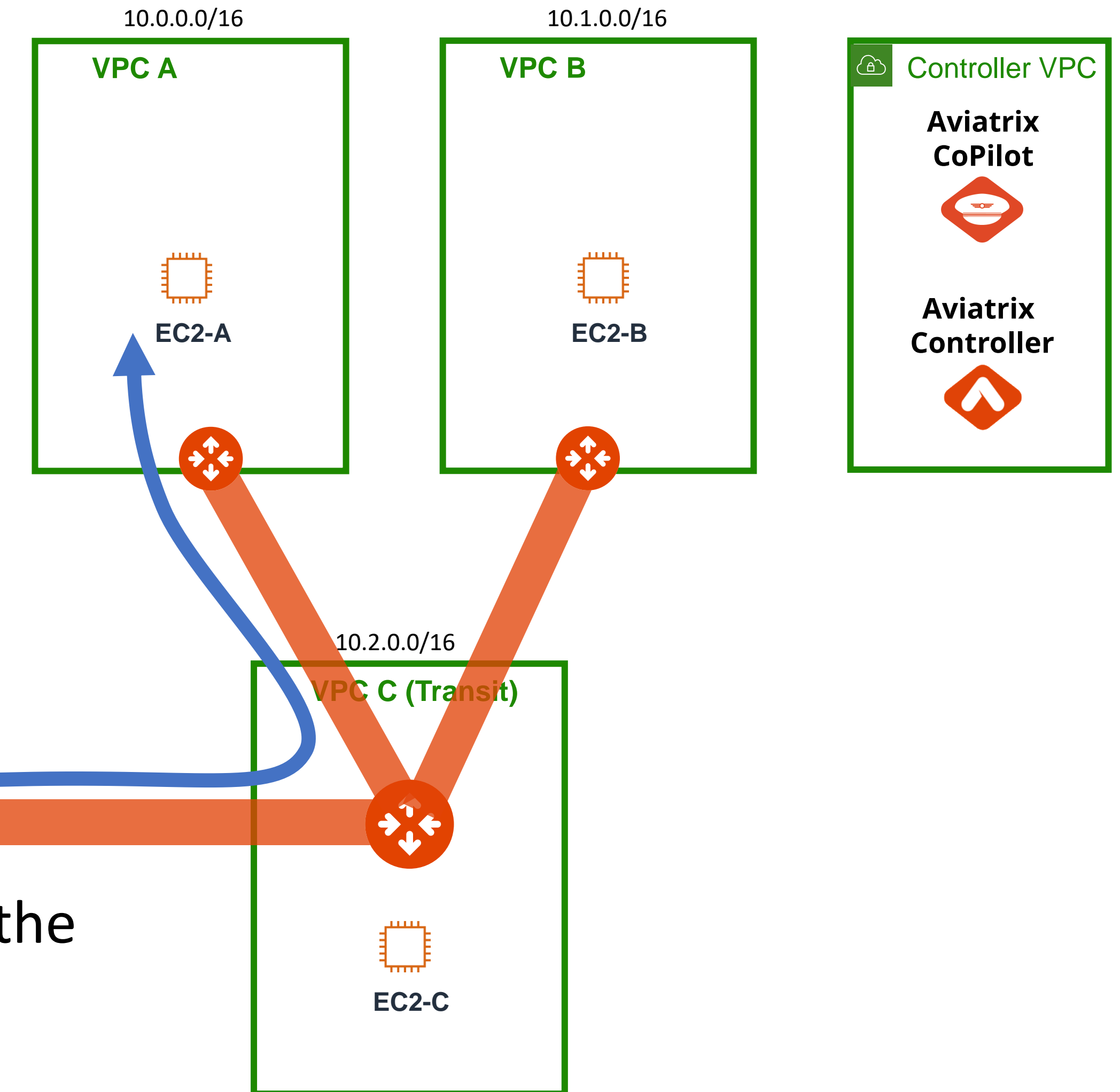
# Lab 2: Cloud Backbone: Test

## Test Connectivity

### AWS us-west-2



### AWS us-east-1



Let's have EC2-A in us-east-1 ping the SAP3 instance in us-west-2



## Lab 2: Cloud Backbone: Step 2.24

Access the console of instance EC2-A in us-east-1

From the AWS Console make sure you're in the correct region **1**

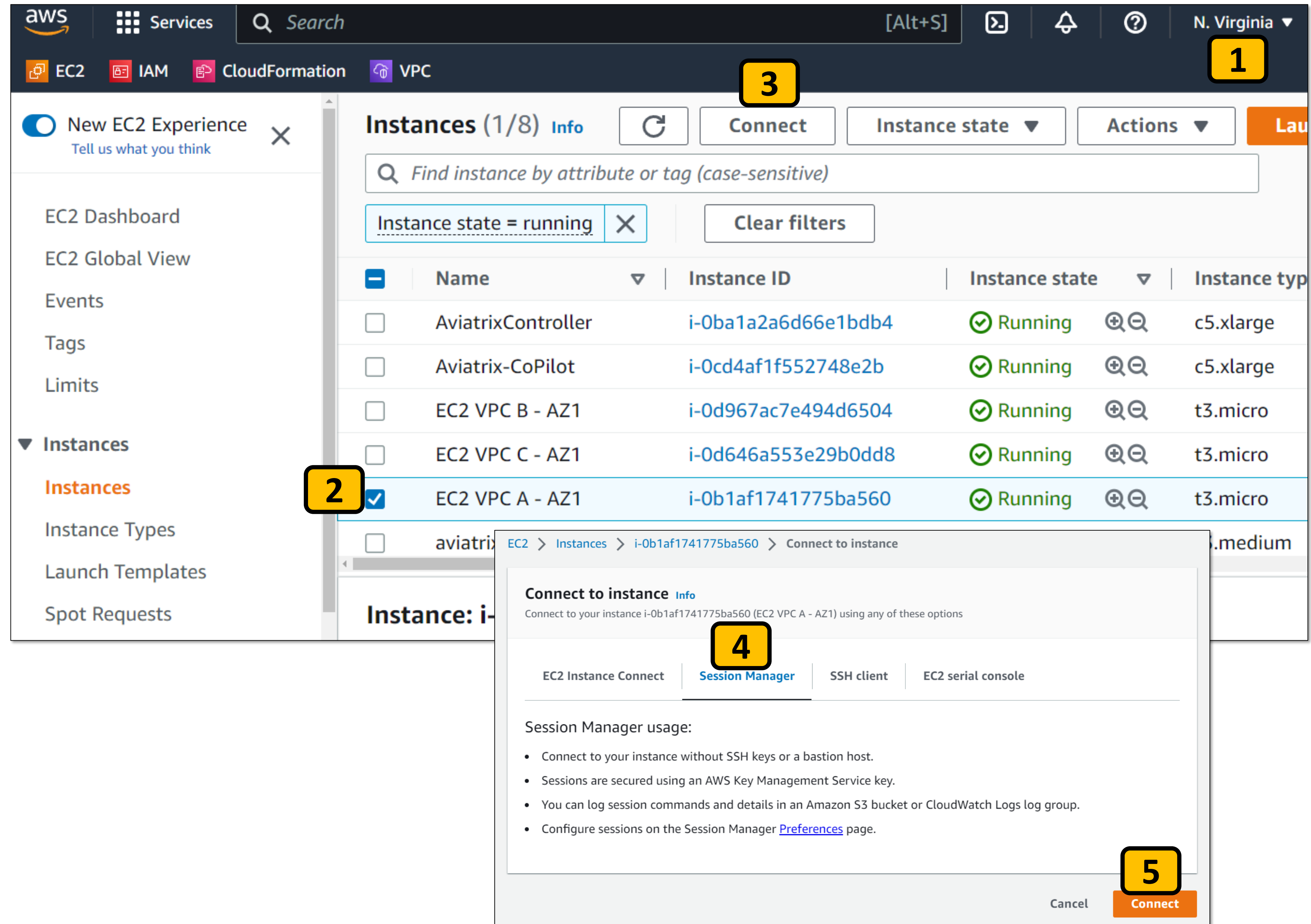
Select the **EC2 VPC A – AZ1** instance **2**

Click on **Connect** **3**

Select the **Session Manager** tab **4**

Click on **Connect** **5**

This will open a console on that instance where we can ping... (next)



The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and the region 'N. Virginia' (annotated with **1**). The left sidebar contains navigation links for EC2, IAM, CloudFormation, and VPC. The main content area displays the 'Instances (1/8)' page. A filter 'Instance state = running' is applied (annotated with **2**). The instance list table shows several instances, with 'EC2 VPC A - AZ1' (Instance ID: i-0b1af1741775ba560) selected (annotated with **2**). The 'Connect' button is visible above the table (annotated with **3**). Below the table, the 'Connect to instance' modal is open, showing the 'Session Manager' tab selected (annotated with **4**). The modal includes instructions on Session Manager usage and a 'Connect' button at the bottom right (annotated with **5**).

Name	Instance ID	Instance state	Instance type
AviaatrixController	i-0ba1a2a6d66e1bdb4	Running	c5.xlarge
Aviaatrix-CoPilot	i-0cd4af1f552748e2b	Running	c5.xlarge
EC2 VPC B - AZ1	i-0d967ac7e494d6504	Running	t3.micro
EC2 VPC C - AZ1	i-0d646a553e29b0dd8	Running	t3.micro
<b>EC2 VPC A - AZ1</b>	<b>i-0b1af1741775ba560</b>	Running	t3.micro



## Lab 2: Cloud Backbone: Step 2.25

Ping test from EC2-A to SAP3

From the instance console type:

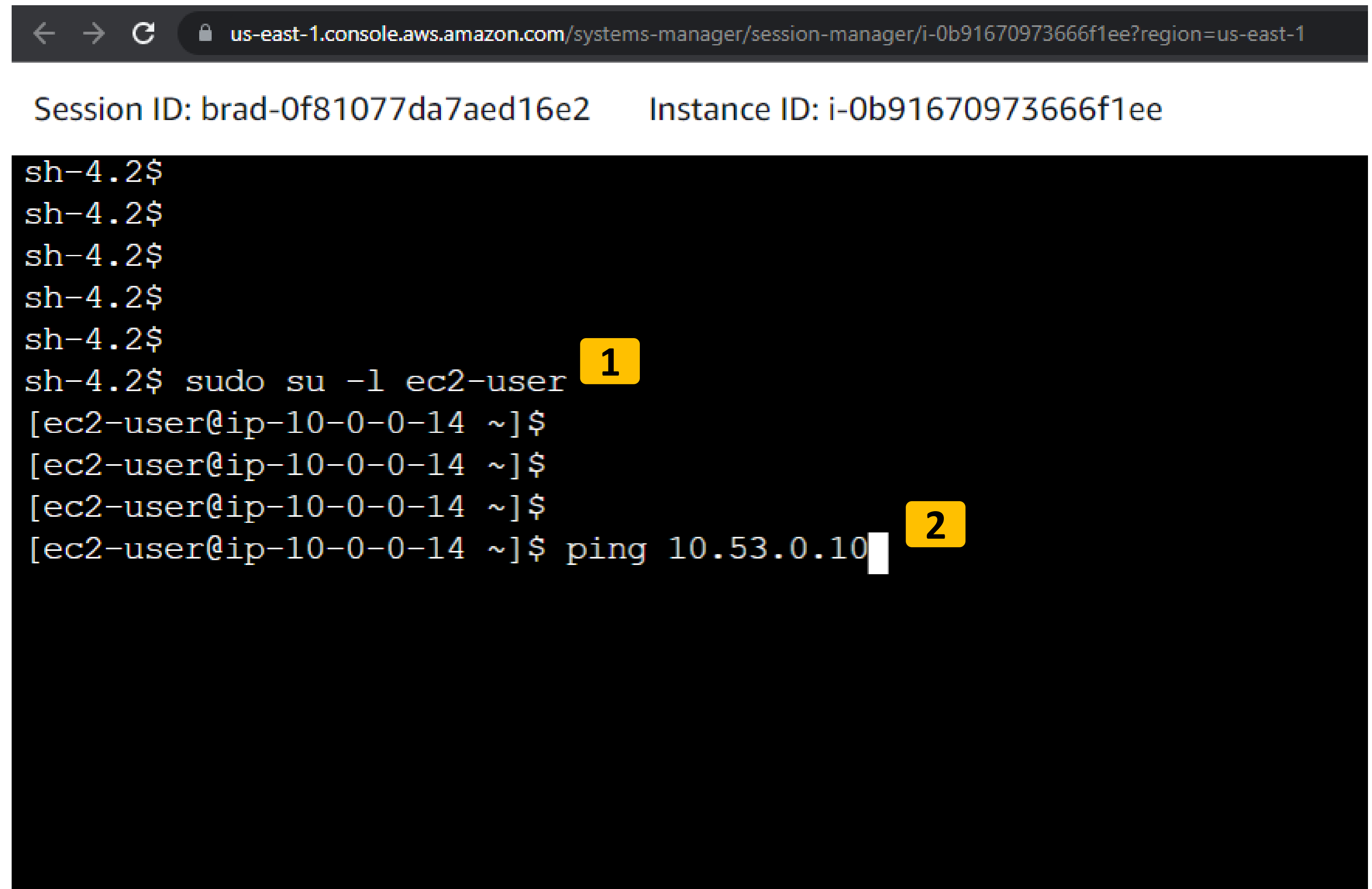
**sudo su -l ec2-user** **1**

(that's a dash lowercase L)

Now let's try to ping SAP3 instance in the other region...

Enter the command:

**ping 10.53.0.10** **2**



```
← → ↻ 🔒 us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-0b91670973666f1ee?region=us-east-1

Session ID: brad-0f81077da7aed16e2    Instance ID: i-0b91670973666f1ee

sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$ sudo su -l ec2-user 1
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$ ping 10.53.0.10 2
```

## Lab 2: Cloud Backbone: Step 2.26

Did the ping work?

OH SNAP! It's not working!?

1

This SHOULD be working

Major bummer. Now we must manually analyze every route table and security group configuration, right?

No, we don't! Let's use CoPilot to do the troubleshooting for us! (next)

```

us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-0b91670973666f1ee?region=us-east-1

Session ID: brad-0f81077da7aed16e2 Instance ID: i-0b91670973666f1ee

sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$ sudo su -l ec2-user
Last login: Mon Feb 27 05:54:11 UTC 2023 on pts/0
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$ ping 10.53.0.10
PING 10.53.0.10 (10.53.0.10) 56(84) bytes of data.

```

1

## Lab 2: Cloud Backbone: Step 2.27

Troubleshoot connectivity problems with ApplQ in CoPilot

Let's use the ApplQ troubleshooting tool in Copilot to find the problems.

Go to Troubleshoot > ApplQ > **FlightPath** <sup>1</sup>

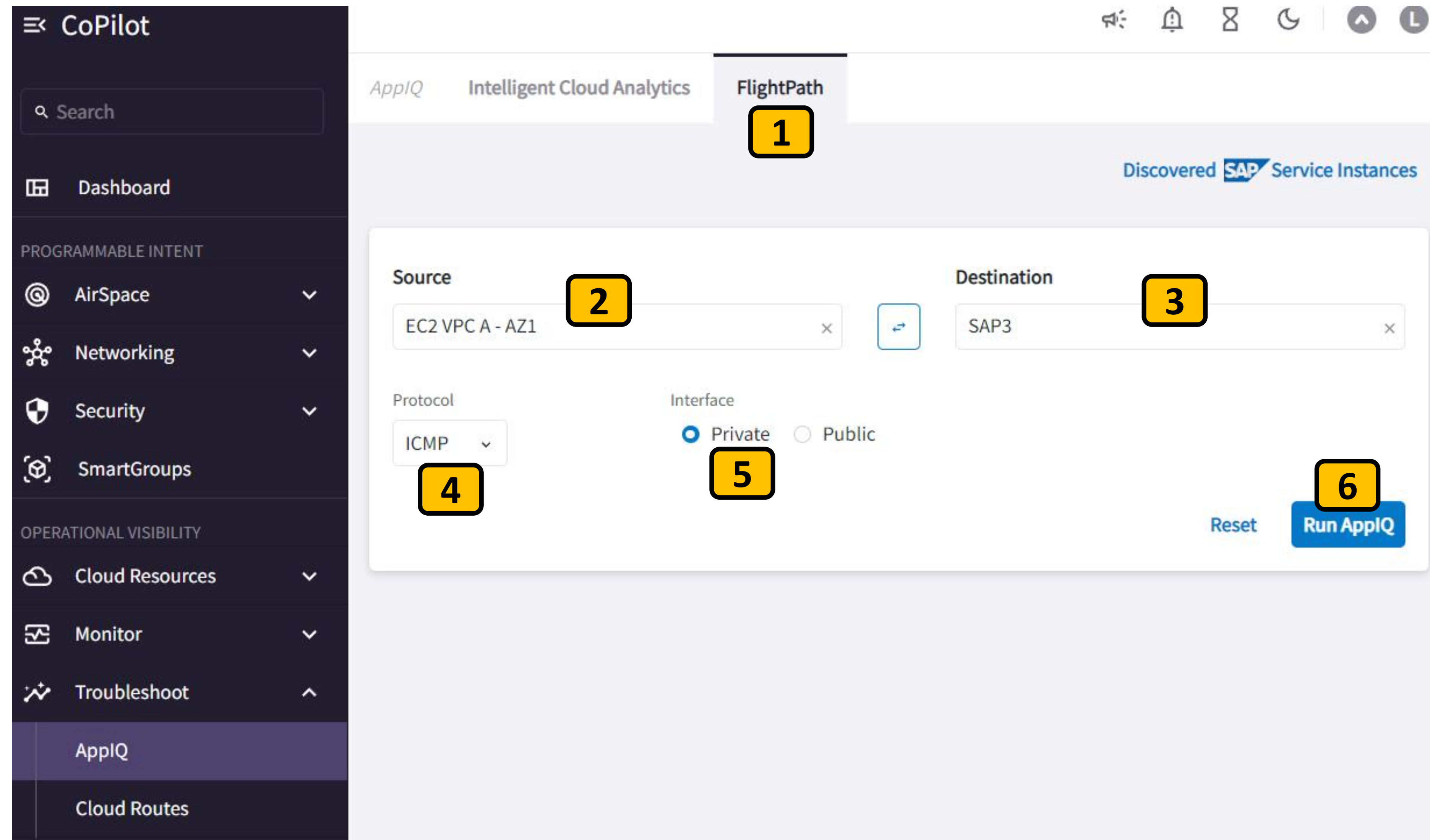
Select **EC2 VPC A – AZ1** as the *Source* instance <sup>2</sup>

Select **SAP3** as the *Destination* instance <sup>3</sup>

For Protocol select **ICMP** <sup>4</sup>

Select **Private** <sup>5</sup>

Click **Run ApplQ** <sup>6</sup>



The screenshot shows the AWS CoPilot interface with the 'FlightPath' tab selected. The 'Source' field is set to 'EC2 VPC A - AZ1' and the 'Destination' field is set to 'SAP3'. The 'Protocol' is set to 'ICMP' and the 'Interface' is set to 'Private'. The 'Run ApplQ' button is visible at the bottom right.

**CoPilot will now analyze all gateways, route tables, security groups, and NACLs for ICMP to succeed between these two instances**

## Lab 2: Cloud Backbone: Step 2.28a

View the ApplQ report to find the problem

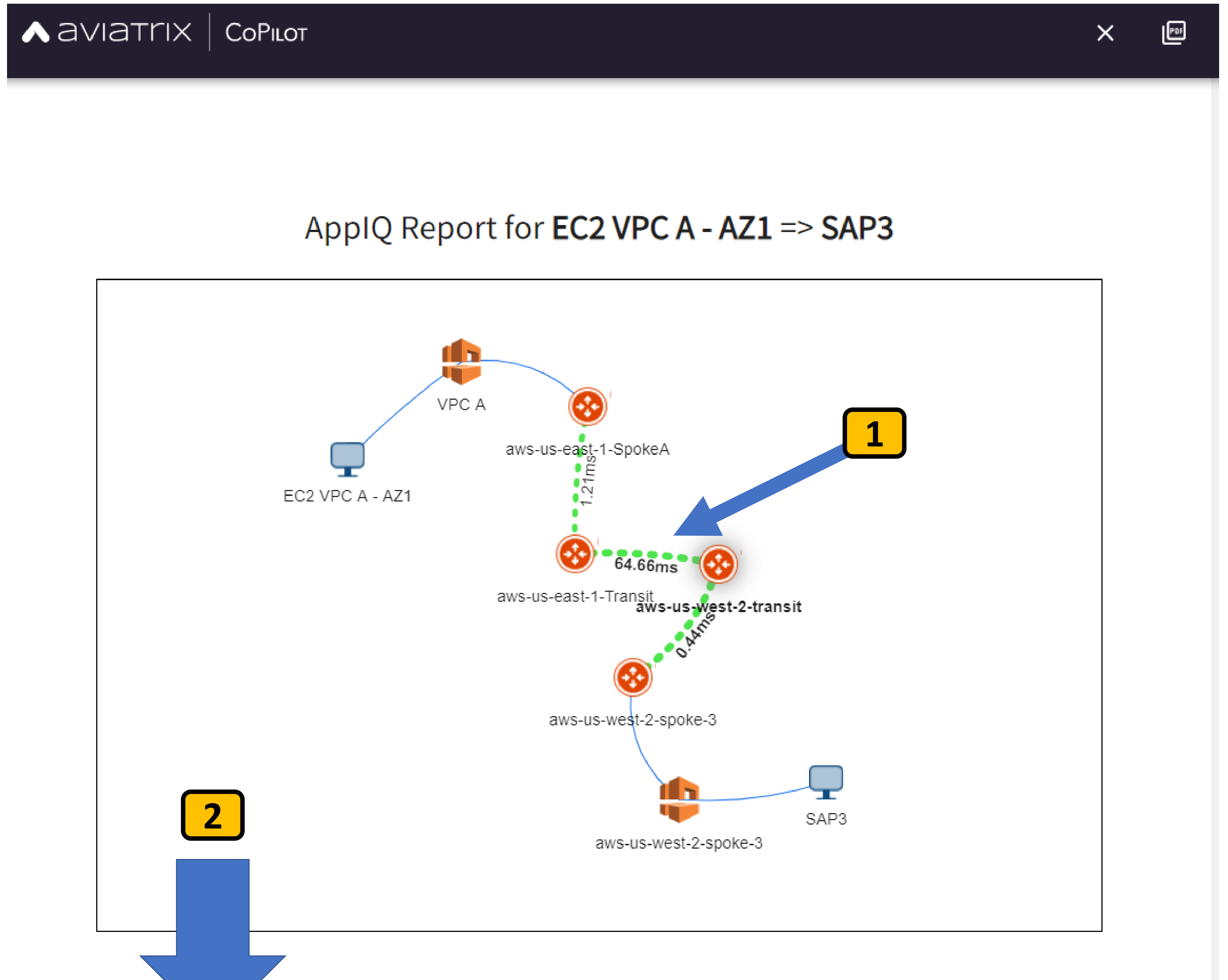
The ApplQ report gives a bunch of useful information.

The ApplQ report begins by showing us the network topology between these two instances, including link health and latency. **1**

Green lines between the Aviatrix Gateways means their connection to each other is good.

So far so good.

**Scroll down** to see the rest of the report **2**





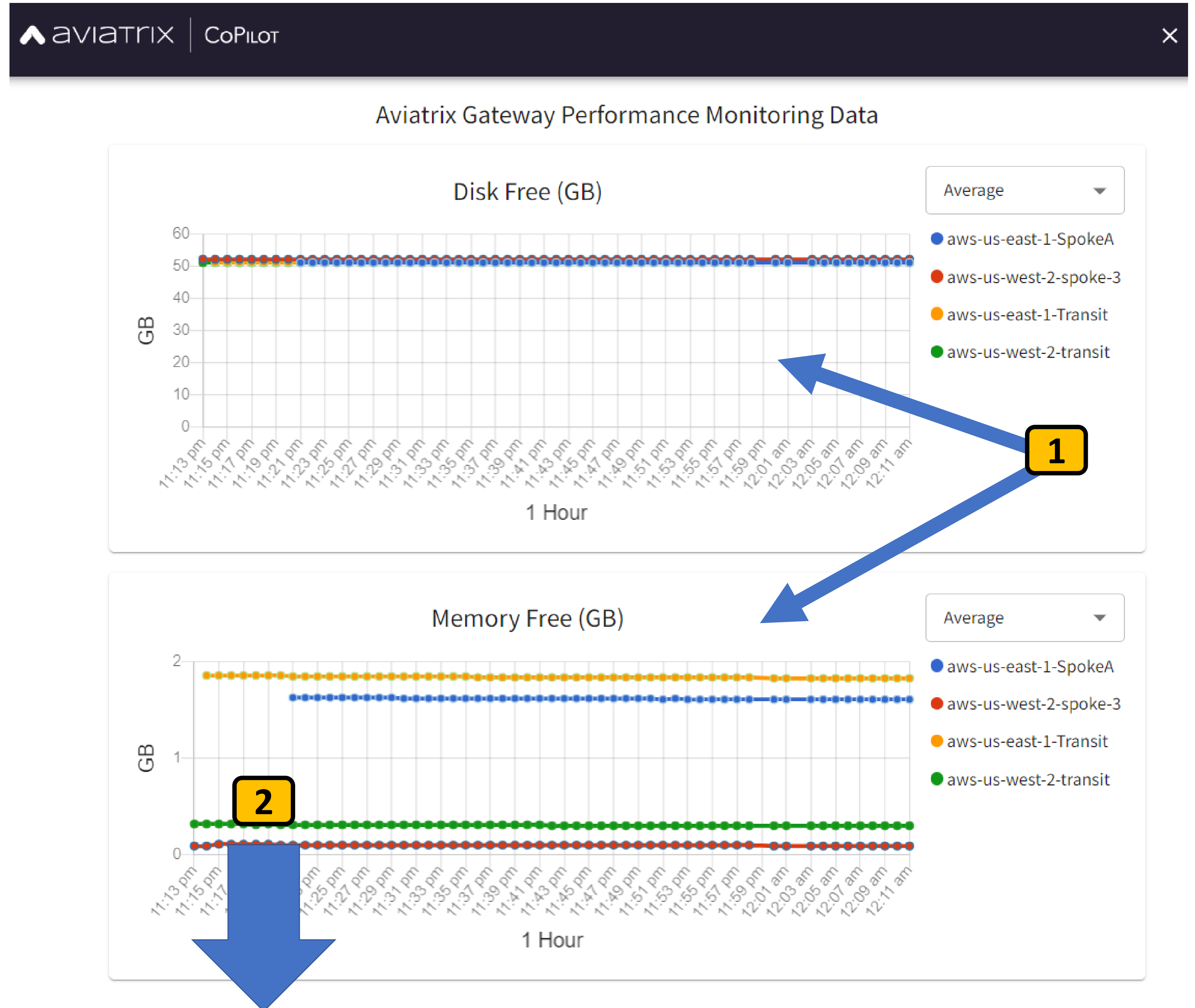
## Lab 2: Cloud Backbone: Step 2.28b

View the ApplQ report to find the problem

Next, ApplQ will give us a **ton of metrics concerning the health and utilization** of the AviaTrix Gateways carrying the traffic between these two instances.

So far so good..

**Scroll down** to see the rest of the report **2**



## Lab 2: Cloud Backbone: Step 2.28c

View the ApplQ report to find the problem

Next, ApplQ will **search the network traffic history** sent from the Source and Destination instances.

Have they been talking to ANYBODY? **1**

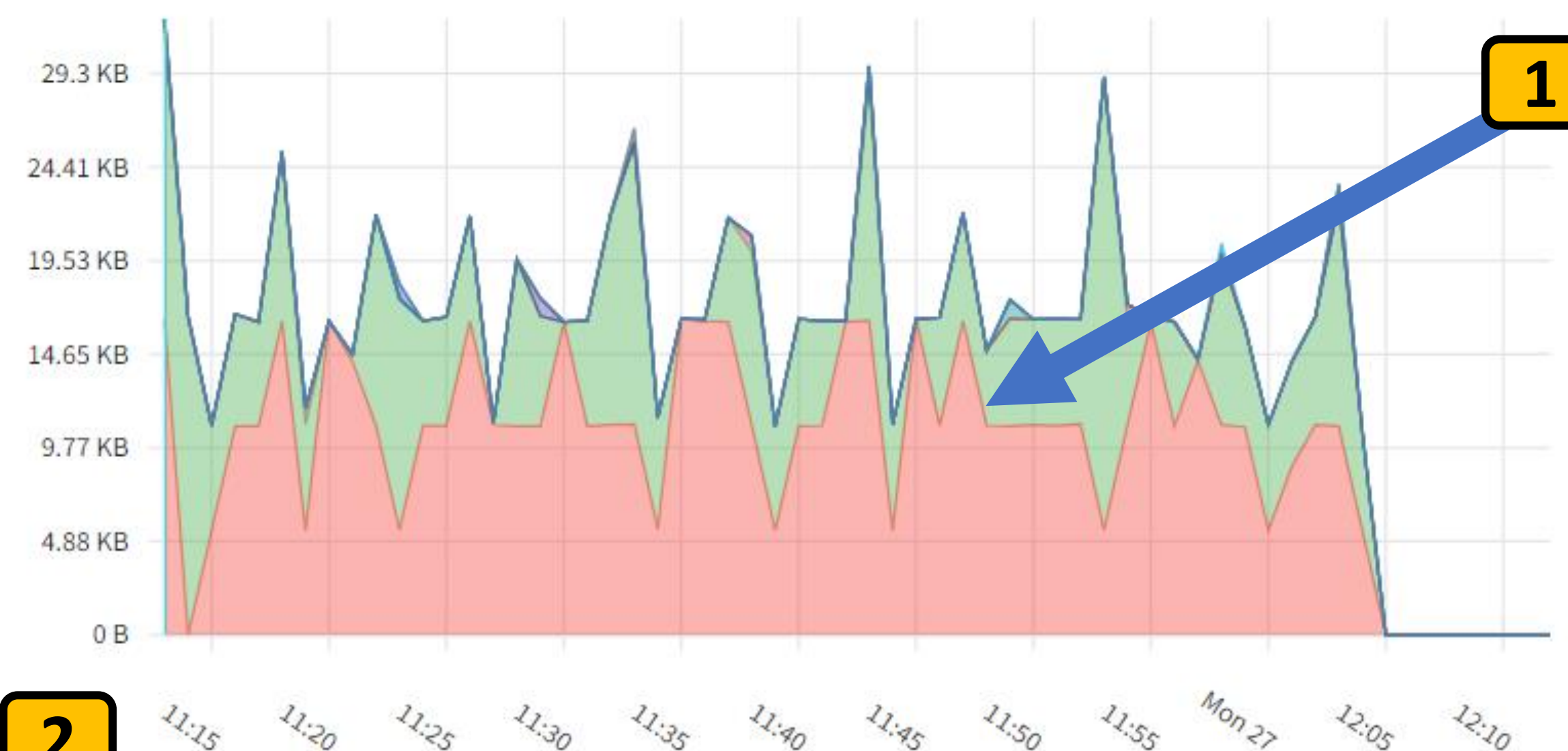
Yes they have ...  
So far so good..

**Scroll down** to see the rest of the report

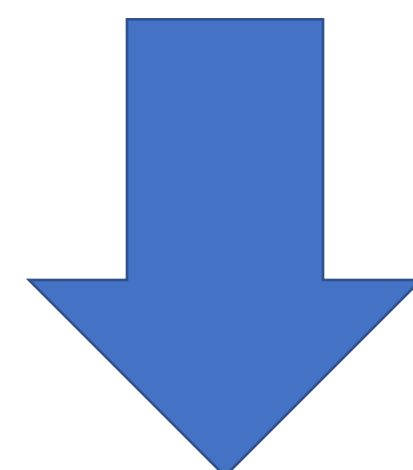


FlowIQ Data for EC2 VPC A - AZ1 and SAP3

Total Bandwidth Usage For Destination Instance (bytes)



**2**



## Lab 2: Cloud Backbone: Step 2.28d

View the ApplQ report to find the problem

Next, ApplQ will **search the network traffic history** to see if the two instances have ever talked to each other in the past.

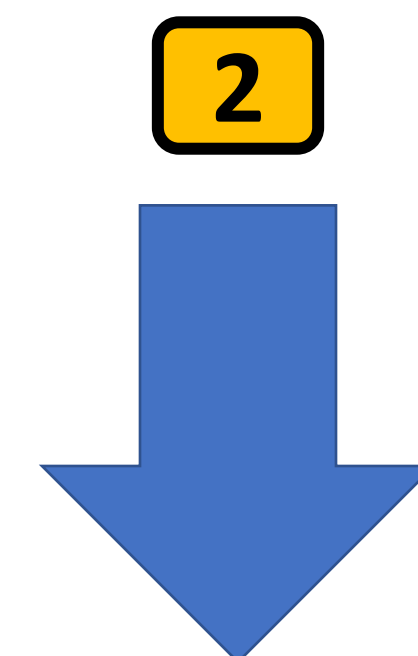
Have they ever talked to EACH OTHER? **1**

Yes they have ...  
So far so good..

This information really helps us to know that our cloud network and these instances are working properly.

So what's the problem?

**Scroll down** to see the rest of the report **2**



# Lab 2: Cloud Backbone: Step 2.28e

View the ApplQ report to find the problem

Next, ApplQ will **check for proper configuration** of all routing table and NACL and security groups in this path.

AHA! CoPilot found the problem! **1**

CoPilot found that there is no entry in the **inbound security group rules** for the destination instance SAP3 that would allow ICMP to work! **2**

CoPilot even gives us a direct link to the security group with the problem.

Click on the security group link.  
This will bring us to that security group in the AWS console to fix it. **3**

Flightpath Data for Destination Instance: SAP3

Network ACL: ✓ Pass

ACL ID: [acl-09bdab531c452d44a](#)

Direction	Protocol	Port	CIDR	Allow/Deny
Outbound	ALL	ALL	0.0.0.0/0	allow
Outbound	ALL	ALL	0.0.0.0/0	deny
Inbound	ALL	ALL	0.0.0.0/0	allow
Inbound	ALL	ALL	0.0.0.0/0	deny

Security Groups: ✗ Fail **1**

Group Name: immersion-sap-sg-3 ([sg-02989109d681b5007](#)) **3**

Inbound Rules				
Type	Protocol	Port Range	Source	Description
http	tcp	80	10.0.0.0/8	Allow local HTTP inbound
http	tcp	80	172.16.0.0/16	Allow local HTTP inbound
ssh	tcp	22	10.0.0.0/8	Allow local ssh inbound
ssh	tcp	22	172.16.0.0/16	Allow local ssh inbound
Outbound Rules				
Type	Protocol	Port Range	Destination	Description
All traffic	ALL	ALL	0.0.0.0/0	Allow all outbound



## Lab 2: Cloud Backbone: Step 2.29

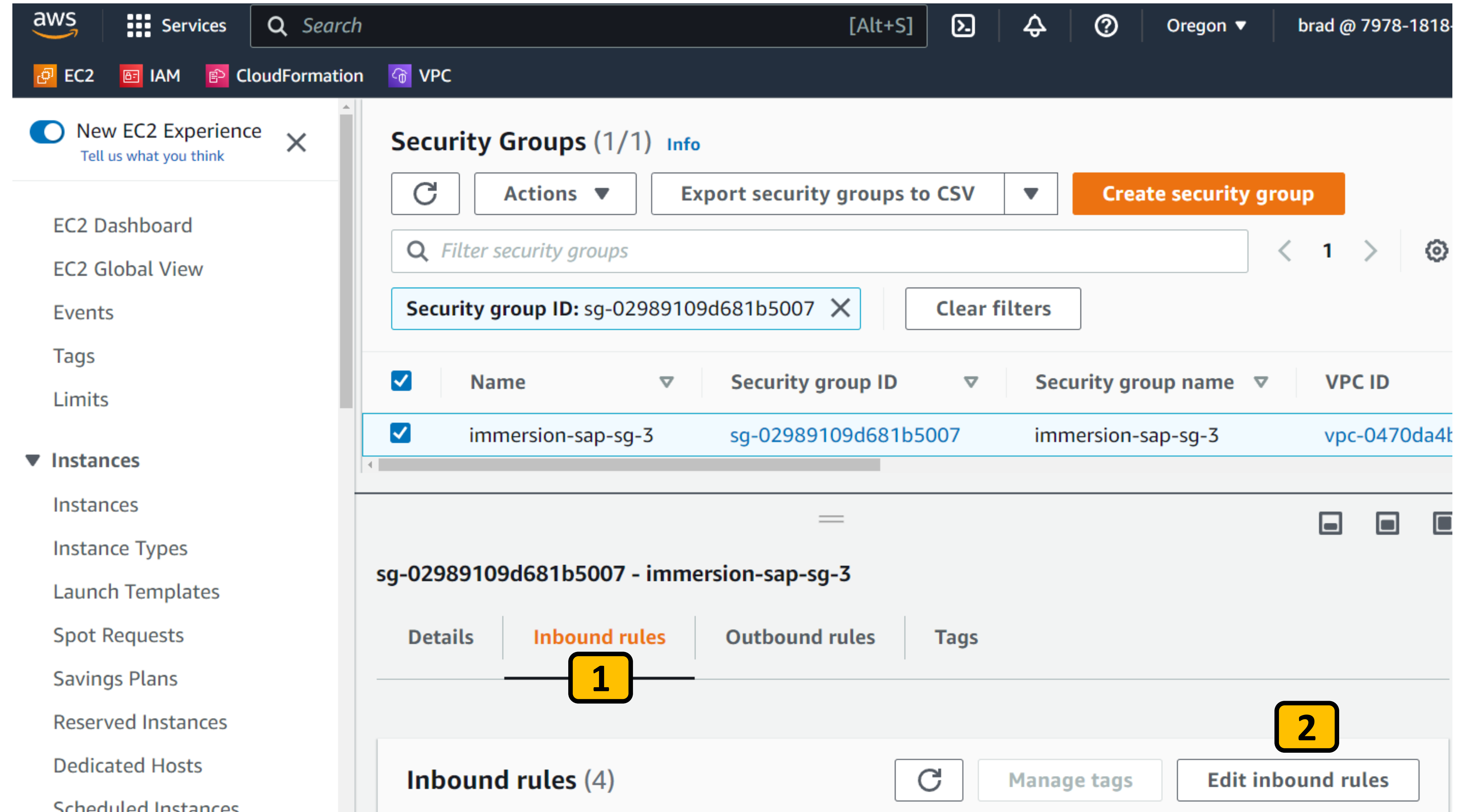
Resolve the connectivity problem

Clicking the link in ApplQ brought us right to the security group to fix it.

Let's fix it.

Click on **Inbound Rules** 1

Click on **Edit inbound rules** 2



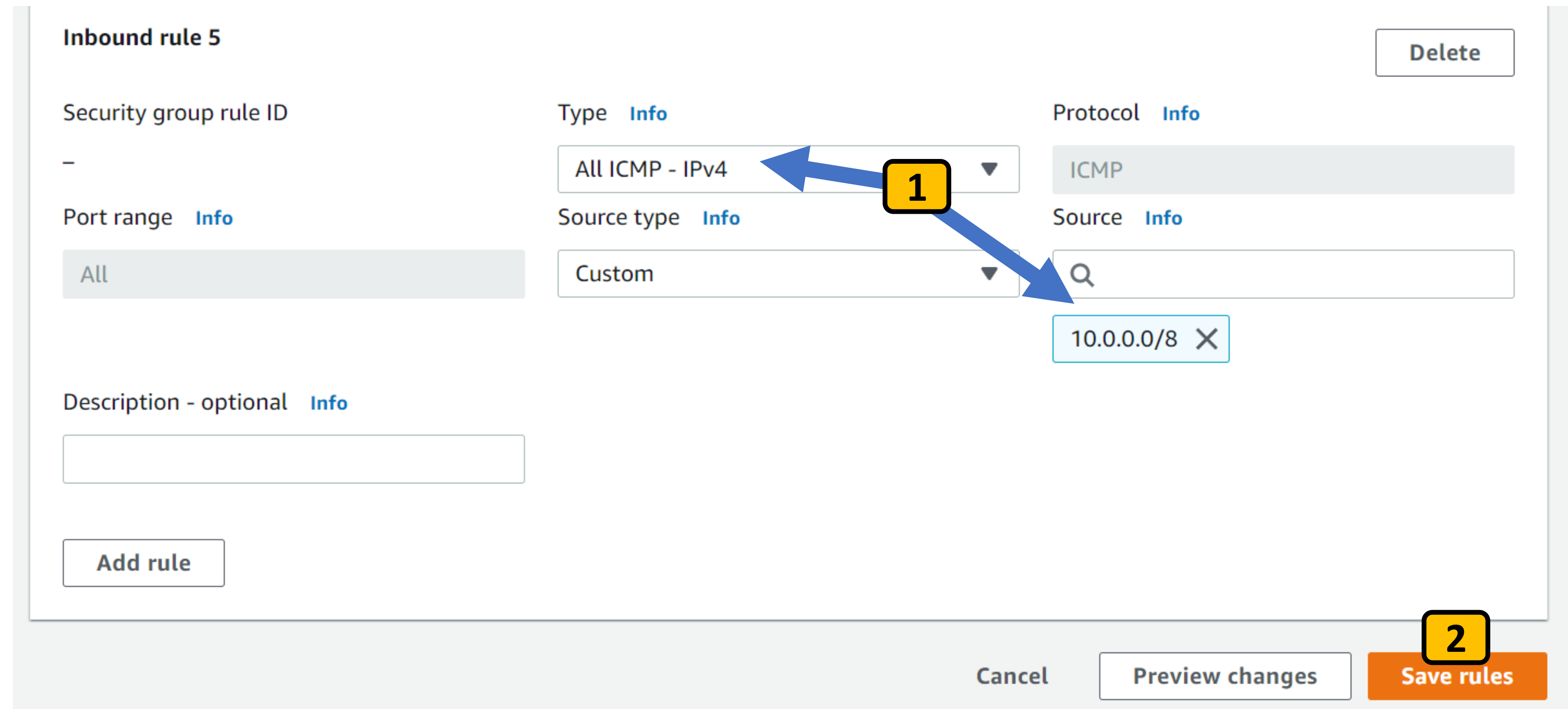
The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The left sidebar contains navigation links for EC2, IAM, CloudFormation, and VPC. The main content area displays the 'Security Groups (1/1)' page for the group 'immersion-sap-sg-3' (ID: sg-02989109d681b5007). The 'Inbound rules' tab is selected and highlighted with a yellow box labeled '1'. Below the tabs, the 'Inbound rules (4)' section is visible, with the 'Edit inbound rules' button highlighted with a yellow box labeled '2'.

## Lab 2: Cloud Backbone: Step 2.30

Resolve the connectivity problem

Add a rule that allows All ICMP v4 from the 10.0.0.0/8 IP range. **1**

Click on **Save rules** **2**



**Inbound rule 5** Delete

Security group rule ID: —

Port range [Info](#): All

Type [Info](#): All ICMP - IPv4

Source type [Info](#): Custom

Protocol [Info](#): ICMP

Source [Info](#): 10.0.0.0/8

Description - optional [Info](#):

Add rule

Cancel Preview changes **2 Save rules**

## Lab 2: Cloud Backbone: Step 2.31

Retest connectivity

Session ID: brad-059347c8e9174a00e

Instance ID: i-0b91670973666f1ee

Now let's try that ping again to see if we fixed the problem.

Go back to the console of the EC2-A instance and ping the SAP3 instance again at 10.53.10 **1**

**SUCCESS!!** **2**

Look how quickly we were able to solve that problem. It wasn't the network's fault. It was a misconfigured security group.

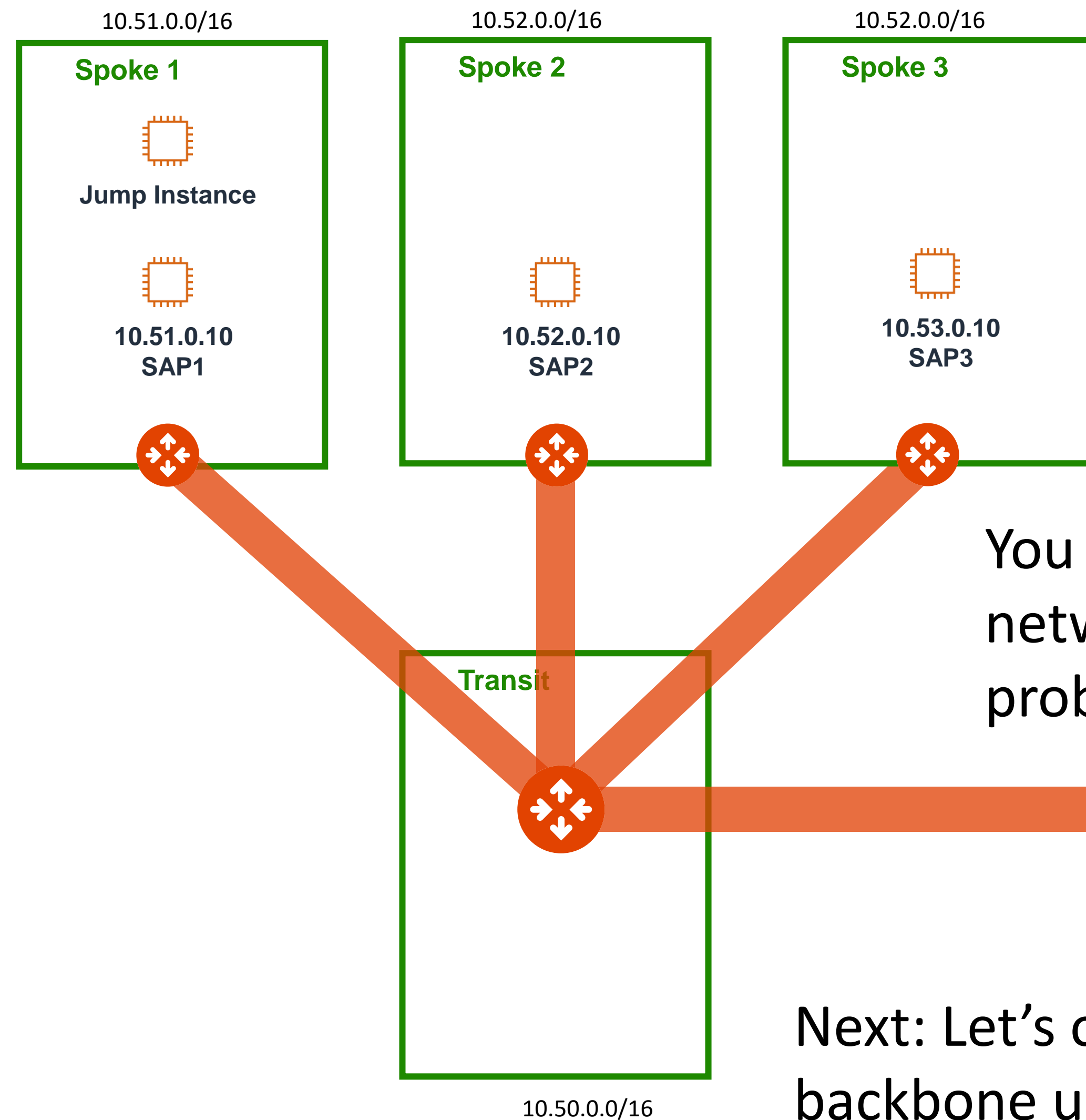
Feel free to ping around and continue testing your network.

```
sh-4.2$
sh-4.2$
sh-4.2$
sh-4.2$ sudo su -l ec2-user
Last login: Mon Feb 27 05:54:31 UTC 2023 on pts/0
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$
[ec2-user@ip-10-0-0-14 ~]$ 1
[ec2-user@ip-10-0-0-14 ~]$ ping 10.53.0.10
PING 10.53.0.10 (10.53.0.10) 56(84) bytes of data.
64 bytes from 10.53.0.10: icmp_seq=1 ttl=60 time=63.1 ms
64 bytes from 10.53.0.10: icmp_seq=2 ttl=60 time=63.0 ms
64 bytes from 10.53.0.10: icmp_seq=3 ttl=60 time=62.9 ms
64 bytes from 10.53.0.10: icmp_seq=4 ttl=60 time=63.3 ms
64 bytes from 10.53.0.10: icmp_seq=5 ttl=60 time=62.9 ms 2
64 bytes from 10.53.0.10: icmp_seq=6 ttl=60 time=62.9 ms
64 bytes from 10.53.0.10: icmp_seq=7 ttl=60 time=63.0 ms
64 bytes from 10.53.0.10: icmp_seq=8 ttl=60 time=63.0 ms
64 bytes from 10.53.0.10: icmp_seq=9 ttl=60 time=62.9 ms
64 bytes from 10.53.0.10: icmp_seq=10 ttl=60 time=63.1 ms
^C
--- 10.53.0.10 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9011ms
rtt min/avg/max/mdev = 62.911/63.054/63.365/0.258 ms
[ec2-user@ip-10-0-0-14 ~]$
```

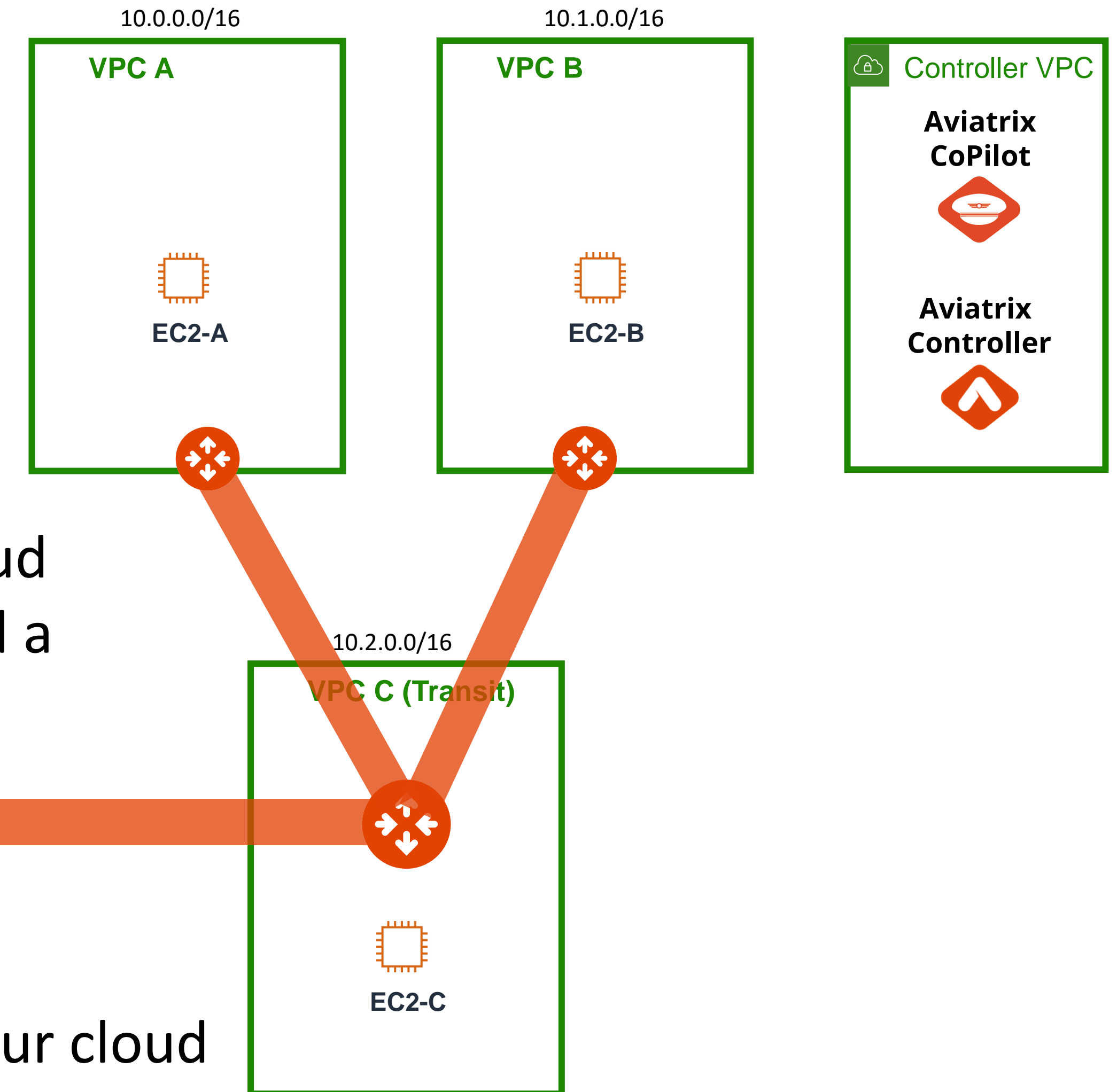
# Lab 2: Cloud Backbone: Progress Check

## Congratulations – You’ve completed Lab 2

### AWS us-west-2



### AWS us-east-1



You just built a multi-region cloud network backbone -- discovered a problem and fixed it quickly.

**Awesome!**

Next: Let's observe network traffic on our cloud backbone using Aviaatrix CoPilot

