



# AWS Immersion Day

## LAB 2

DISTRIBUTED FIREWALL FOR SECURE EGRESS



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

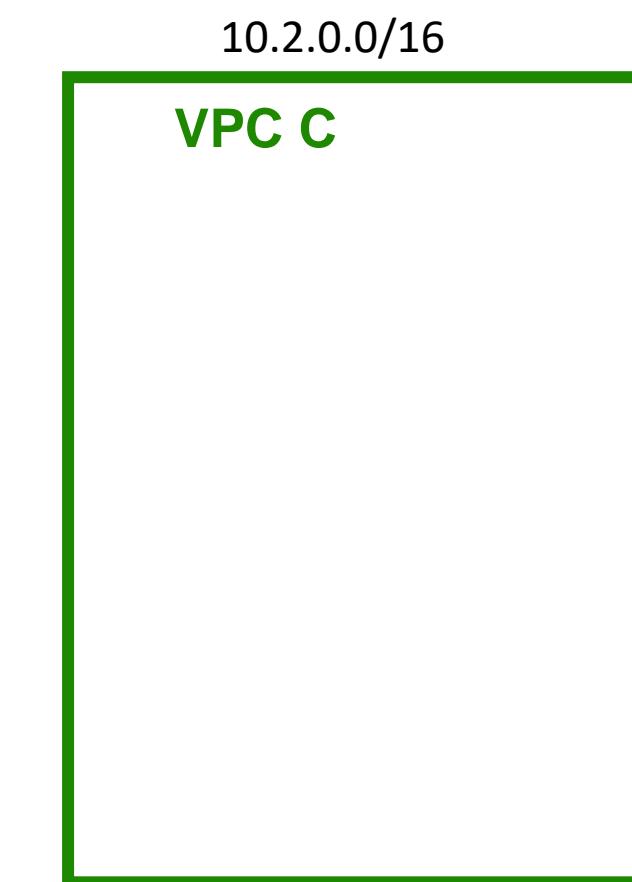
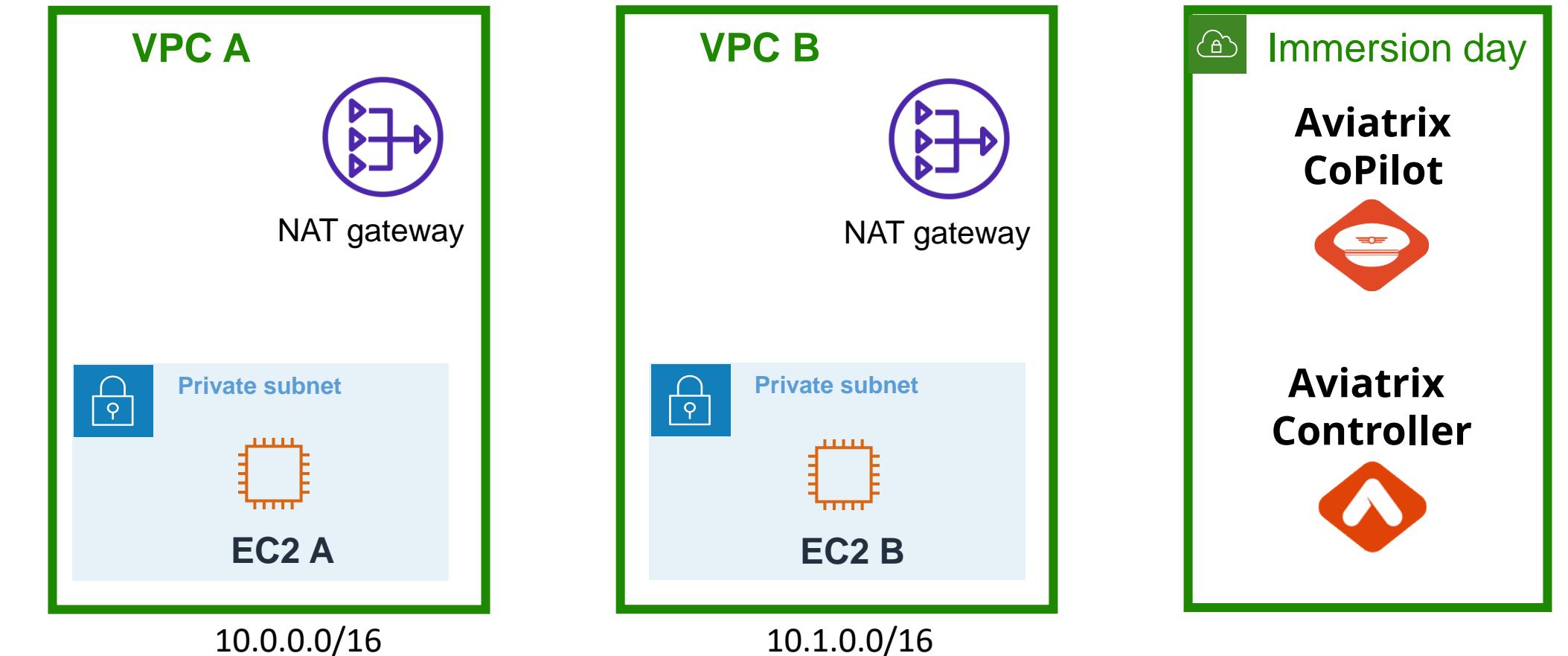
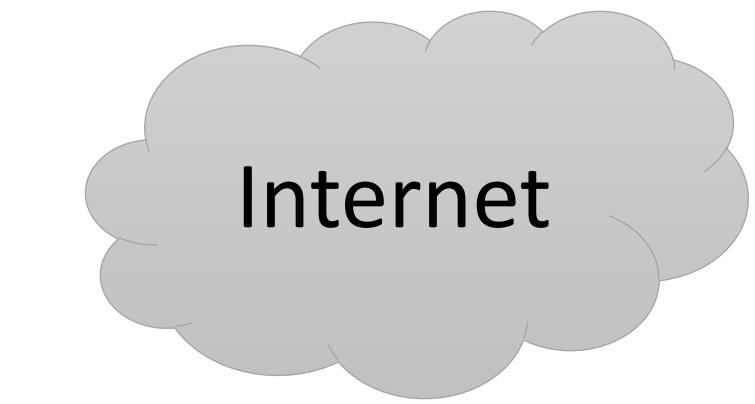
## Lab 1 Recap

In Lab 1 you created (3) AWS VPCs in the us-east-1 region.

VPCs A and B have EC2 instances on a private subnet and AWS NAT Gateway providing internet egress.

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

In this lab you will deploy Aviatrix gateways to provide Secure Egress using the Aviatrix Distributed Firewall



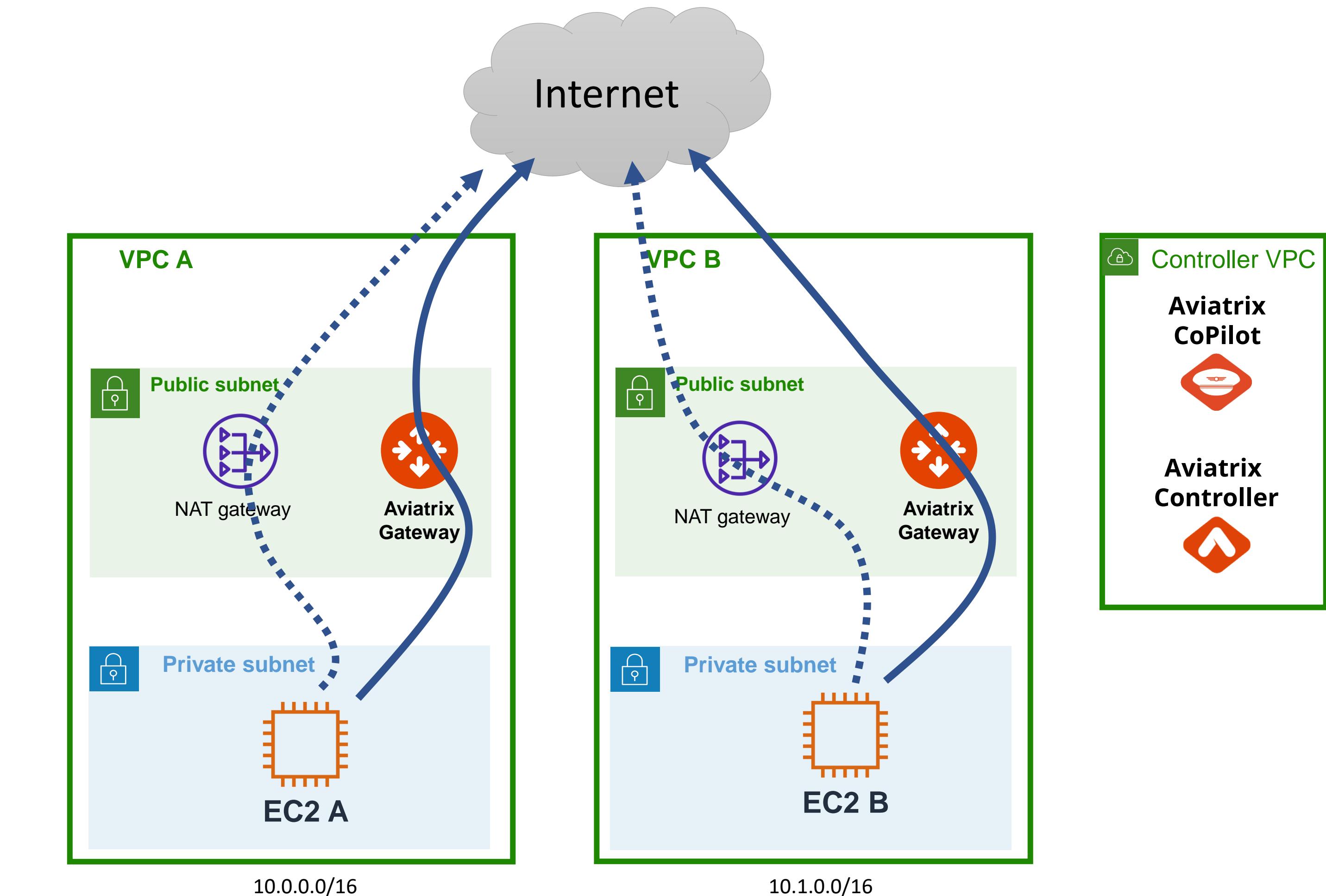
**AWS us-east-1**

## Lab 2 Intro

In Lab 2 you are going to deploy Aviatrix Spoke Gateways in VPCs A & B.

You will create domain name filtering rules in the Aviatrix Distributed Firewall to secure your egress internet traffic.

After that you will seamlessly switch your egress traffic flows from AWS NAT Gateway to Aviatrix Gateways.



AWS us-east-1

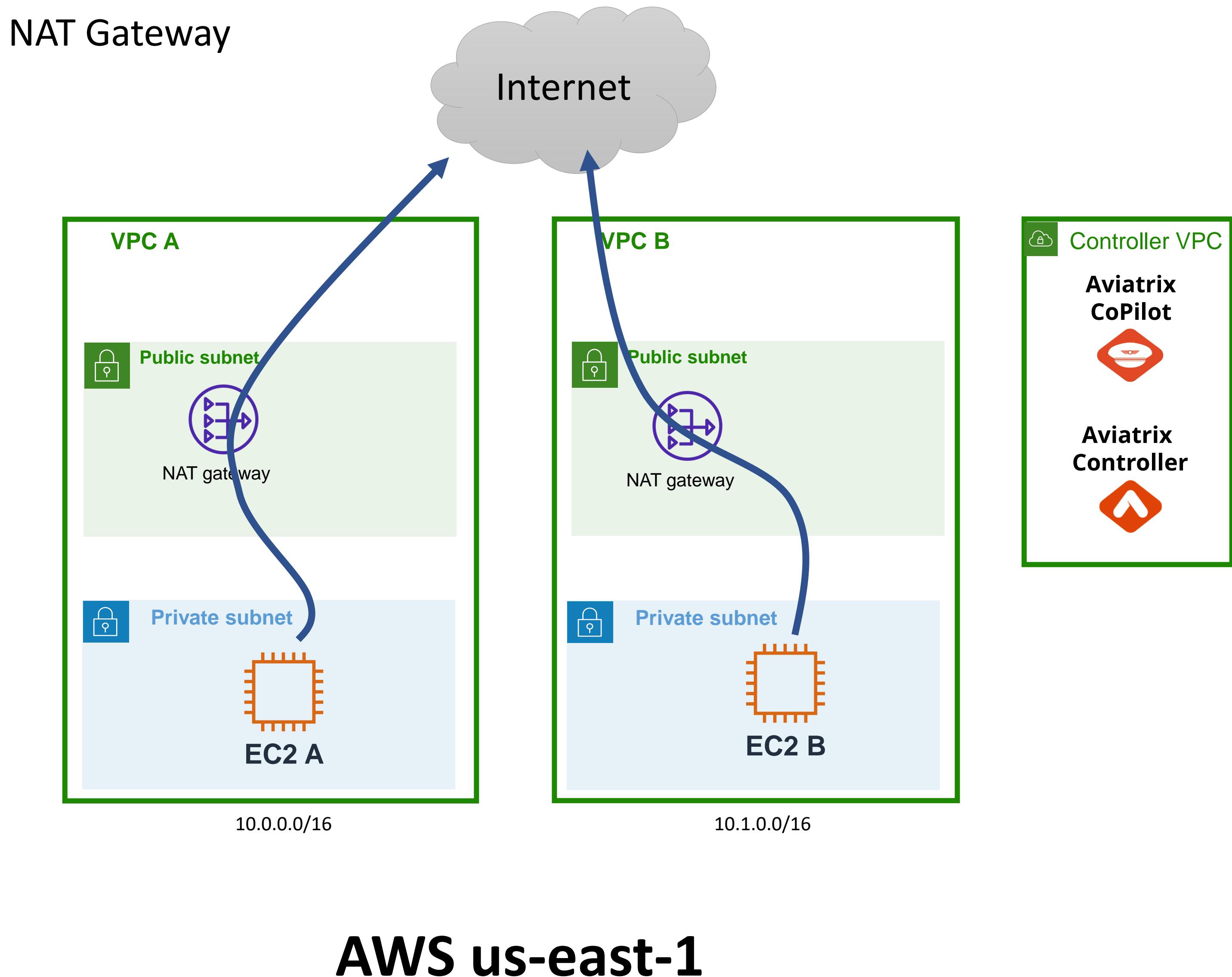
## Lab 2: Step 2

Test egress with AWS NAT Gateway

Before we do anything with Aviatr ix, let's test the internet egress with AWS NAT Gateway.

In the following steps you will verify the default route is using AWS NAT Gateway.

After that you will connect to the console of instance EC2 A and test internet egress.



## Lab 2: Step 2.1

Observe the default route of your VPC A Private Route Table

Make sure your AWS Console is in the us-east-1 N. Virginia region.

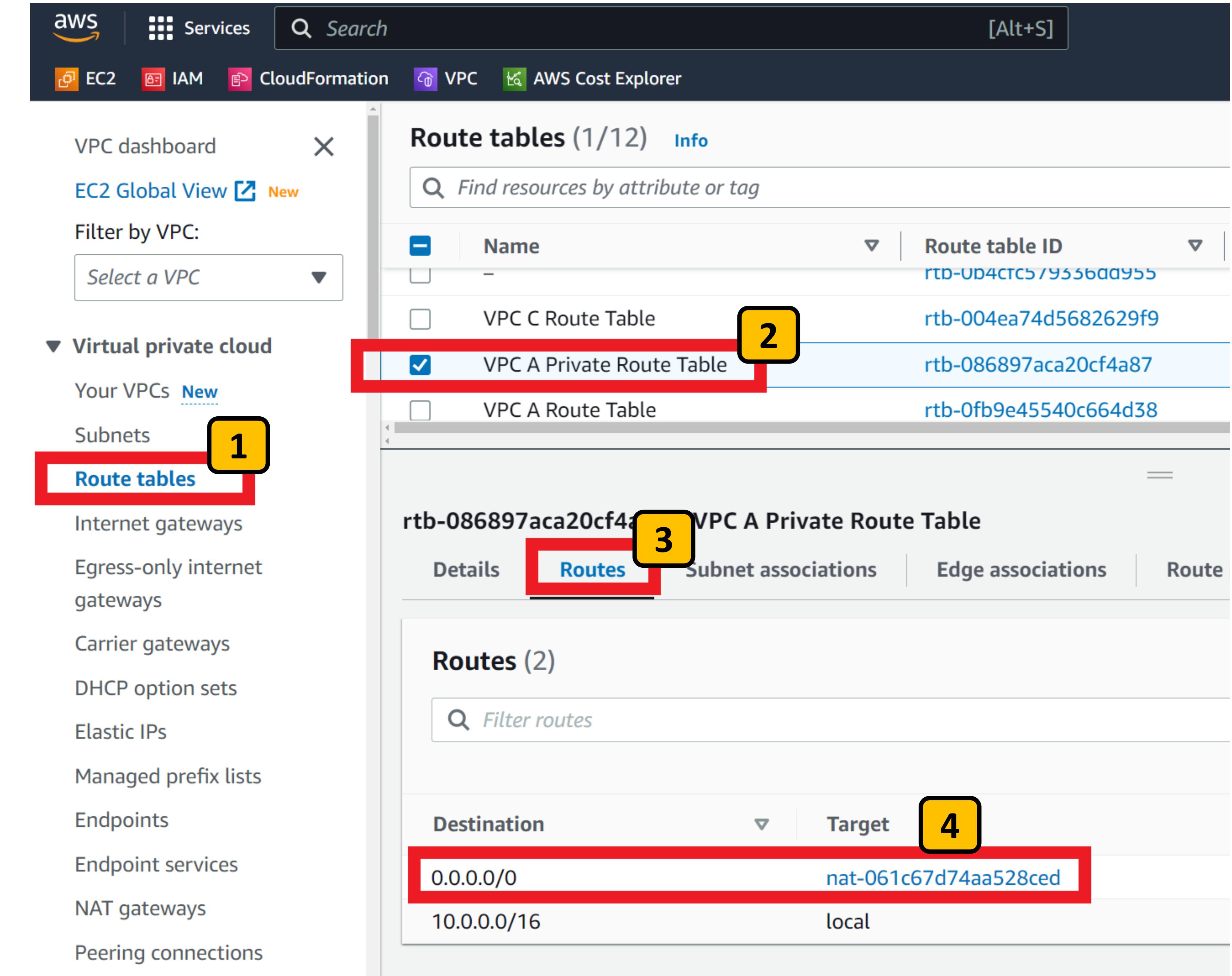
From the AWS Console go to the VPC section of the console and select

**Route tables** 1

Select the **VPC A Private Route Table** 2

Select the **Routes** tab 3

Observe the 0.0.0.0/0 default route directing traffic to AWS NAT Gateway 4



## Lab 2: Step 2.2

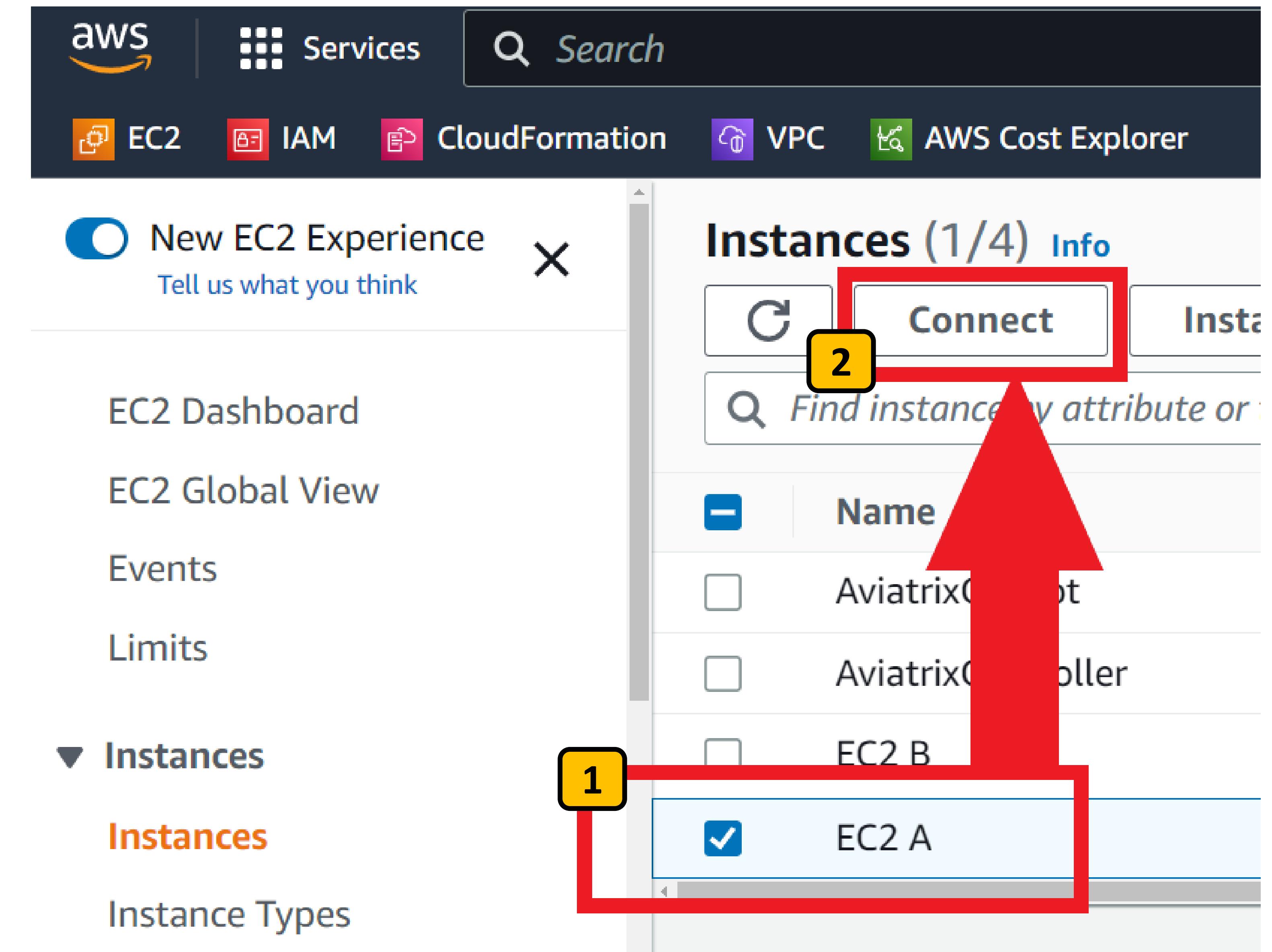
Connect to EC2 A instance console

Go the EC2 section of the AWS Console.

Select Instances

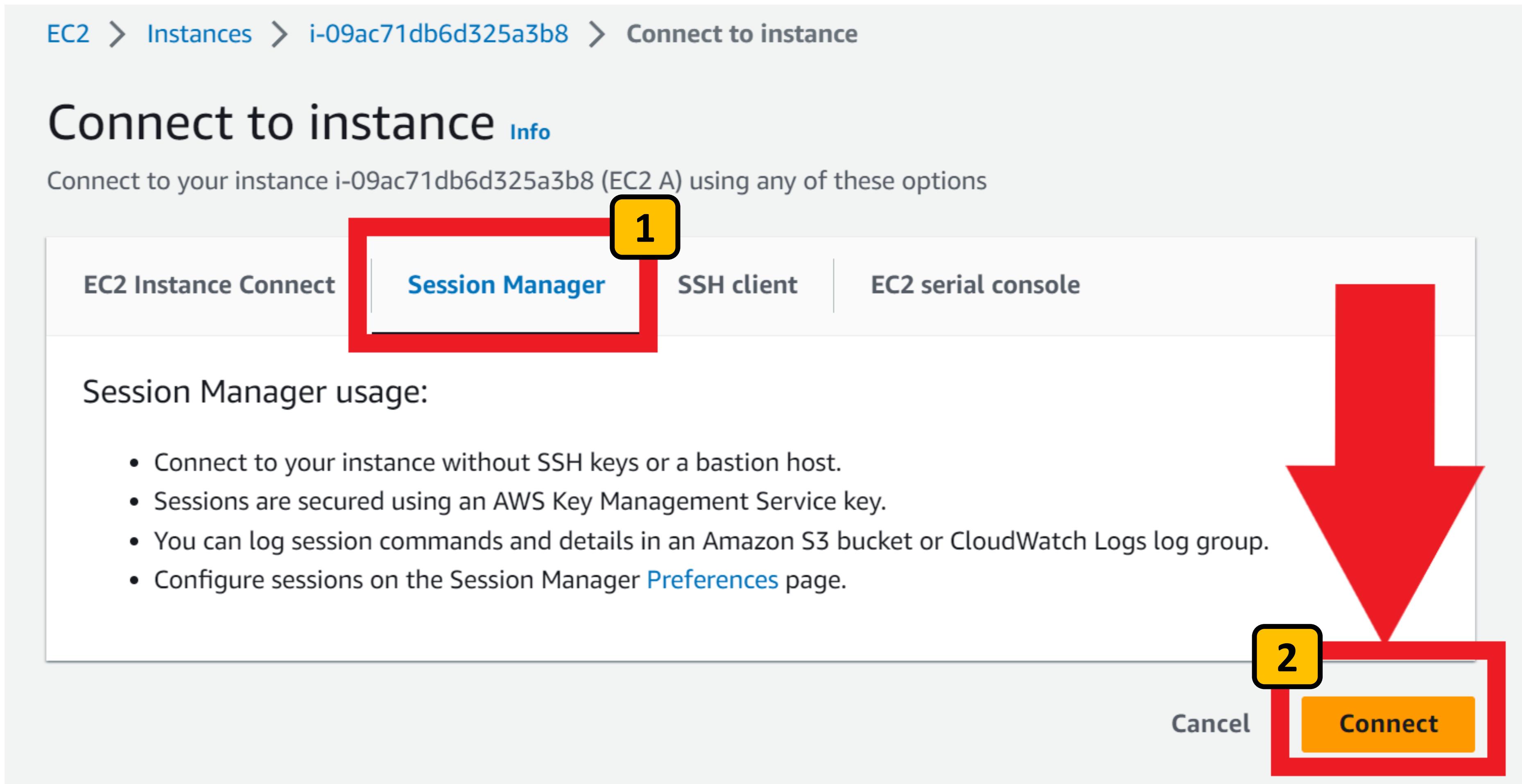
Find the EC2 A instance and select it. **1**

Click the Connect button **2**



## Lab 2: Step 2.3

Connect to EC2 A instance console



Select the Session Manager tab **1**

Click the Connect button **2**

## Lab 2: Step 2.4

Connect to EC2 A instance console

Your browser should open a new tab giving you a CLI session.

Type the command:

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

*(dash lower case L)*

You are now logged on as the ec2-user and should see your private IP address in the hostname.

```
sh-4.2$  
sh-4.2$  
sh-4.2$  
sh-4.2$ sudo su -l ec2-user1  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$
```

Next, let's test internet egress.

## Lab 2: Step 2.5

Test connection to google.com

Let's test a connection to google.com

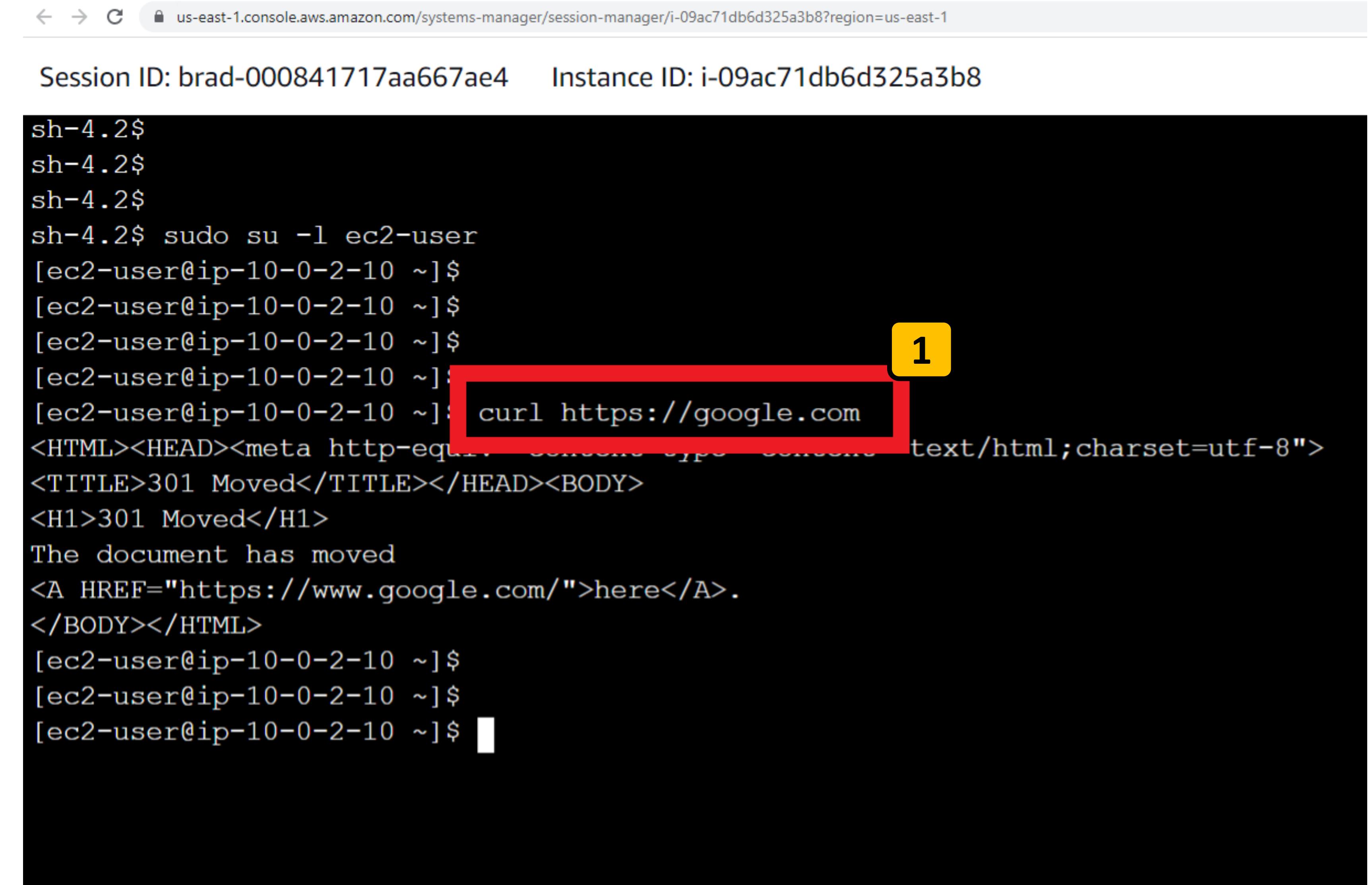
Type the command:

**curl https://google.com** 1

You should see your CLI return HTML code from google.com

Great! Internet is working..

Next, let's do a software update for this instance from AWS...



The screenshot shows a terminal window titled "Session ID: brad-000841717aa667ae4" and "Instance ID: i-09ac71db6d325a3b8". The URL "https://google.com" is highlighted with a red box and a yellow number "1" in the top right corner. The terminal output shows the command being typed and the resulting HTML response from Google.

```
sh-4.2$  
sh-4.2$  
sh-4.2$  
sh-4.2$ sudo su -l ec2-user  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$ curl https://google.com1  
<HTML><HEAD><meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<TITLE>301 Moved</TITLE></HEAD><BODY>  
<H1>301 Moved</H1>  
The document has moved  
<A HREF="https://www.google.com/">here</A>.  
</BODY></HTML>  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$
```

## Lab 2: Step 2.6

Run a software update

Let's run a software update from the AWS repositories

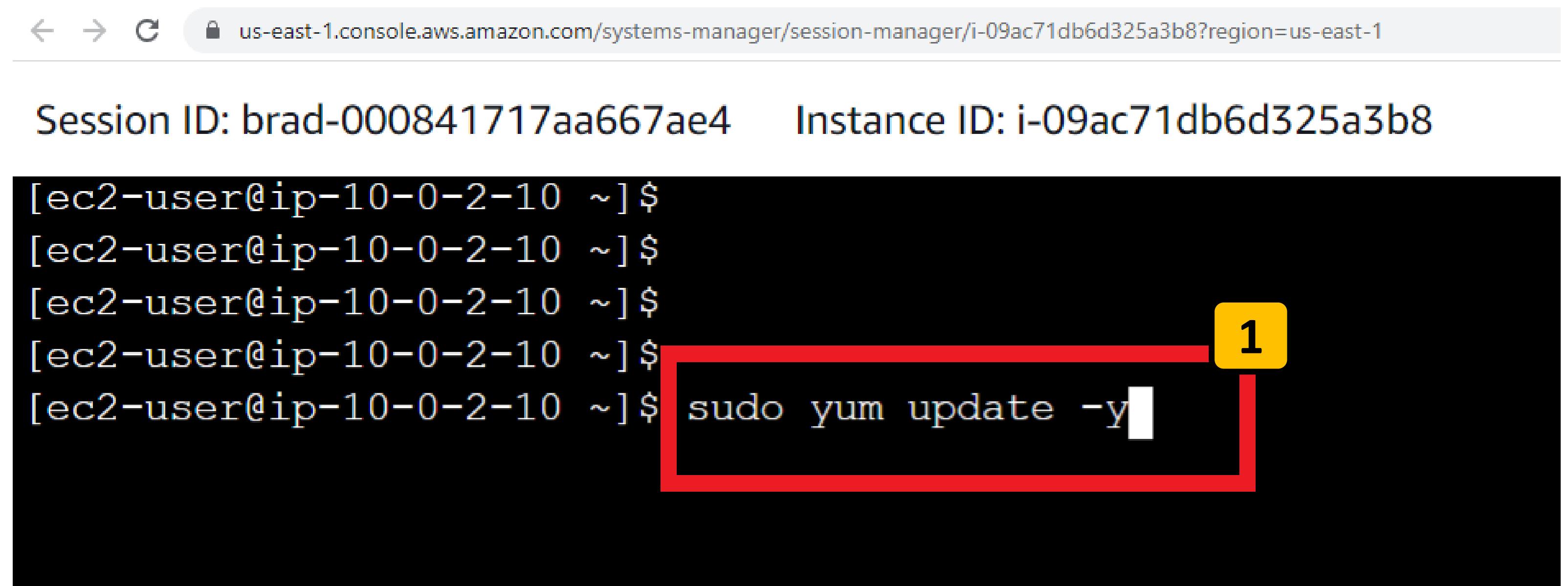
Run the command:

**sudo yum update -y** 1

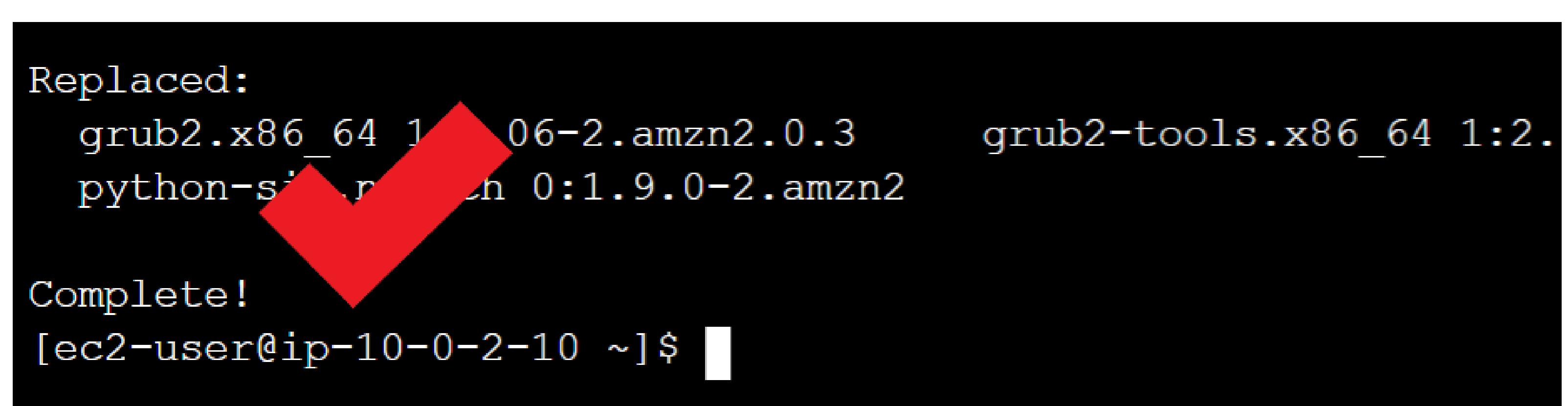
This will connect to **amazonaws.com** domains to download software updates.

Great! Software updates are working..

Next, let's see if this instance can connect to unwanted or potentially harmful domains...



```
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$ sudo yum update -y1
```



```
Replaced:  
grub2.x86_64 1:06-2.amzn2.0.3      grub2-tools.x86_64 1:2.  
python-setuptools 0:1.9.0-2.amzn2  
  
Complete!  
[ec2-user@ip-10-0-2-10 ~]$
```

# Lab 2: Step 2.7

# Test connections to potentially harmful domains

Test that your EC2 instance has unfiltered internet access, to even potentially harmful domains.

# Run the commands:

**curl https://ransomware.org** 1

**curl https://malware.net** 

**curl https://botnet.com** 3

For each command you should see  
the CLI return HTML code from those  
domains ... a successful connection

```
Session ID: brad-000841717aa667ae4    Instance ID: i-09ac71db6d325a3b8

[ec2-user@ip-10-0-2-10 ~] $ curl https://ransomware.org
```

```
Session ID: brad-000841717aa667ae4      Instance ID: i-09ac71db6d325a3b8

[ec2-user@ip-10-0-2-10 ~] $ curl https://malware.net
```

```
Session ID: brad-000841717aa667ae4    Instance ID: i-09ac71db6d325a3b8

[ec2-user@ip-10-0-2-10 ~] $ curl https://botnet.com
```

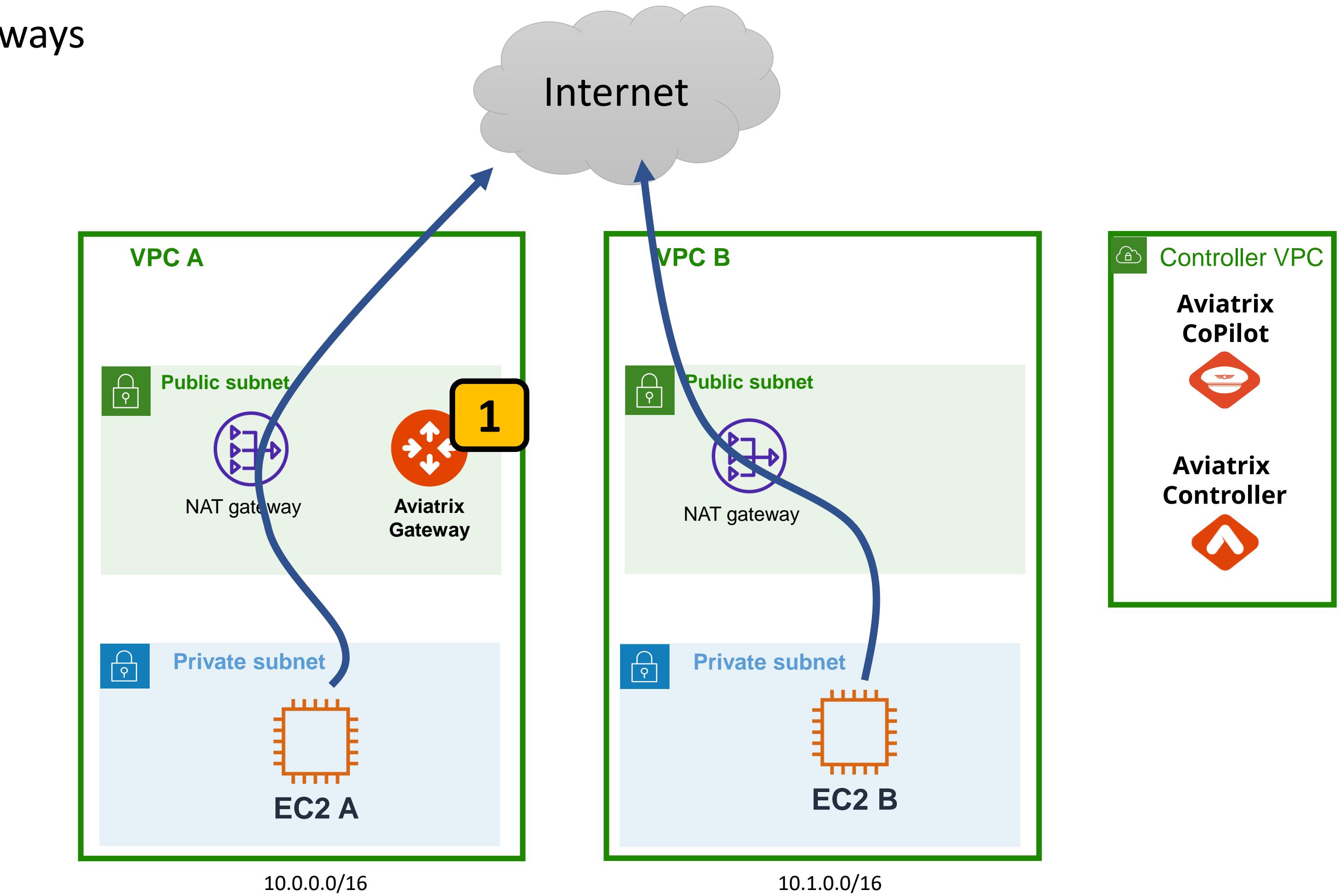
```
com/ , title : Botnet , description : we build great games and  
ba90d8497998ad39721157a5a8", "uri": "/", "title": "Botnet", "icon": "https://d73/images/9d7b3e49-114b-4672-9757-c3196f0cca7f/PNG-1.png"},  
79a74b409d73/uploads/cover/27aacf5f-23c1-4c4b-1  
fb-a32f-79a74b409d73/uploads/favicon/75b8116-9fe7-80a  
idPage": true, "pageId": "b88f83ba90d8  
, "query": {}, "buildId": "39I4gFdmEgLRyw", "isFallback": false  
] }</script></body></html>[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$
```

## Lab 2: Checkpoint 1

Deploy Aviatrix Gateways

At this point you've verified unfiltered egress through the AWS NAT Gateway

In the next steps you will log on to the Aviatrix UI and deploy an Aviatrix Gateway in the public subnet of VPC A



AWS us-east-1

## Lab 2: Step 2.8

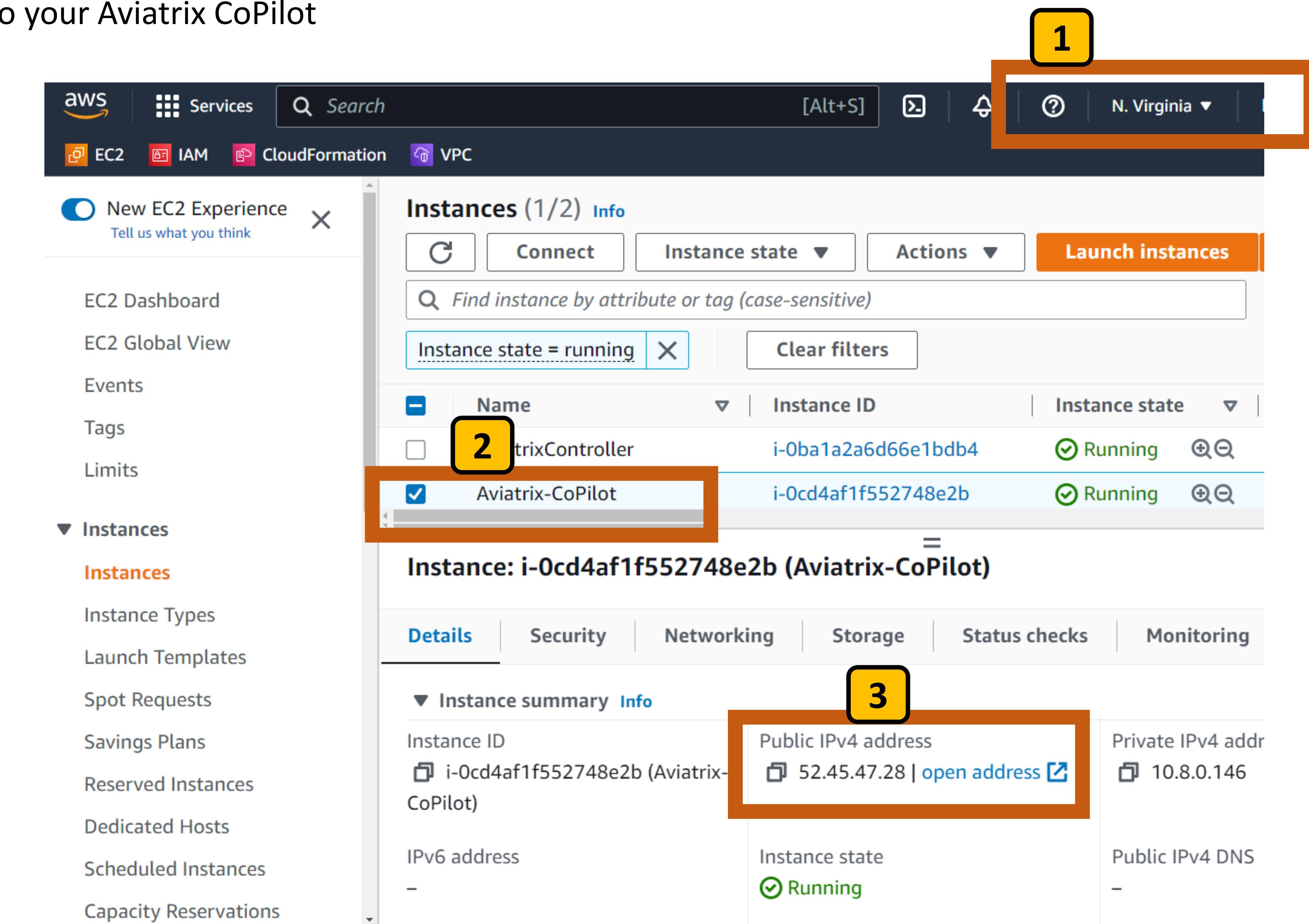
Log in to your Aviatrix CoPilot

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



## Lab 2: Step 2.9

Log in to your Aviatrix CoPilot

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

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

Username = lab\_student

Password = ImmersionDay123# **3**

The screenshot shows a Chrome browser window with two tabs: 'Instances | EC2 Management Con...' and 'Aviatrix CoPilot'. The address bar indicates a 'Not secure' connection to https://52.45.47.28. A red warning icon with an exclamation mark is visible in the top right corner of the browser. The main content area displays the Aviatrix logo and the CoPilot login form. The login form fields are labeled 'Username' and 'Password', with the respective values 'lab\_student' and 'ImmersionDay123#'. Below the form are 'Log In', 'Remember Me', and 'Forgot Password' buttons. To the right of the login form, a detailed description of the certificate warning is shown, including a callout pointing to the 'Advanced' button in the browser's security dialog.

Not secure | https://52.45.47.28

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

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 configuration problem with the server accepting your connection.

Proceed to 34.239.54.147 (unsafe)

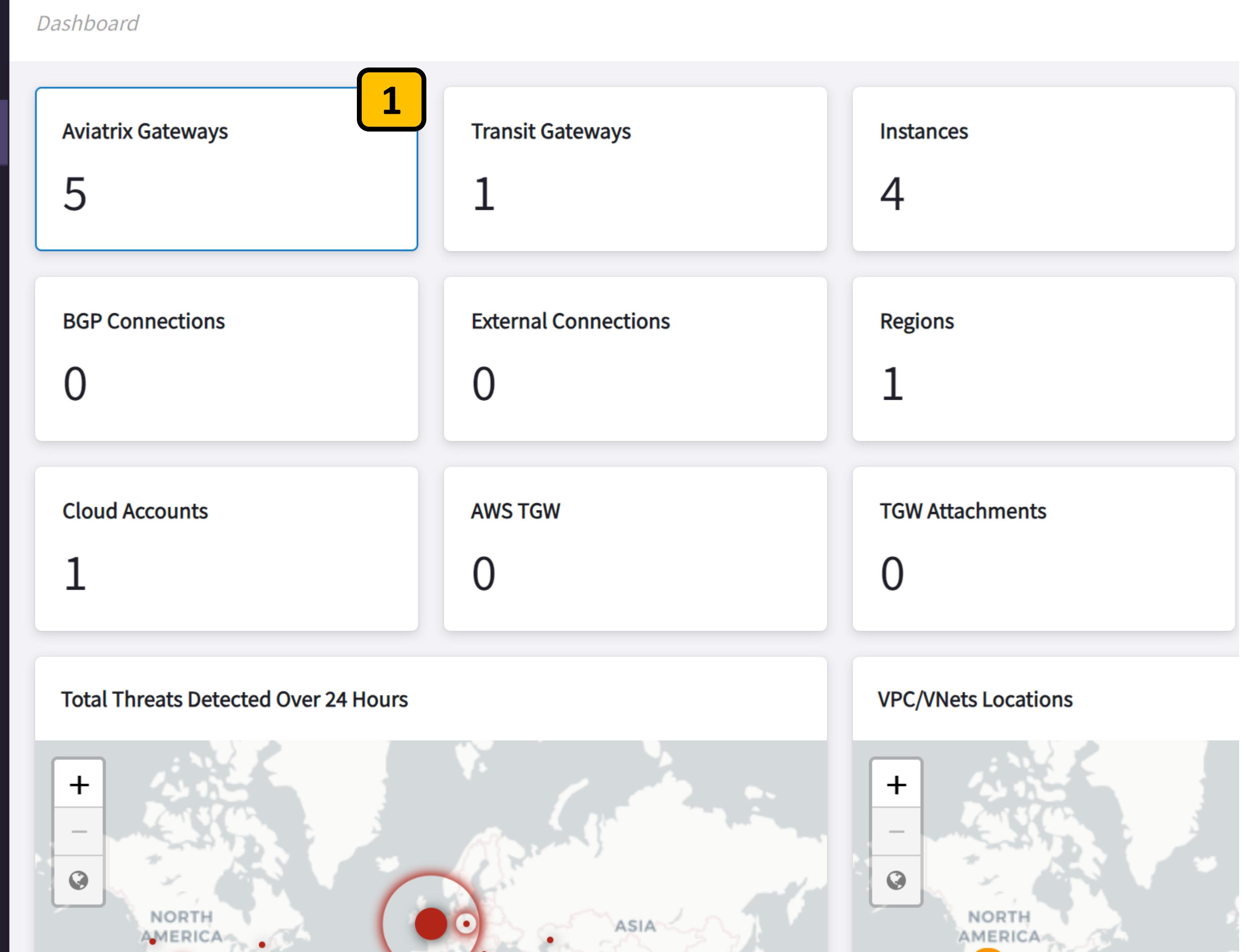
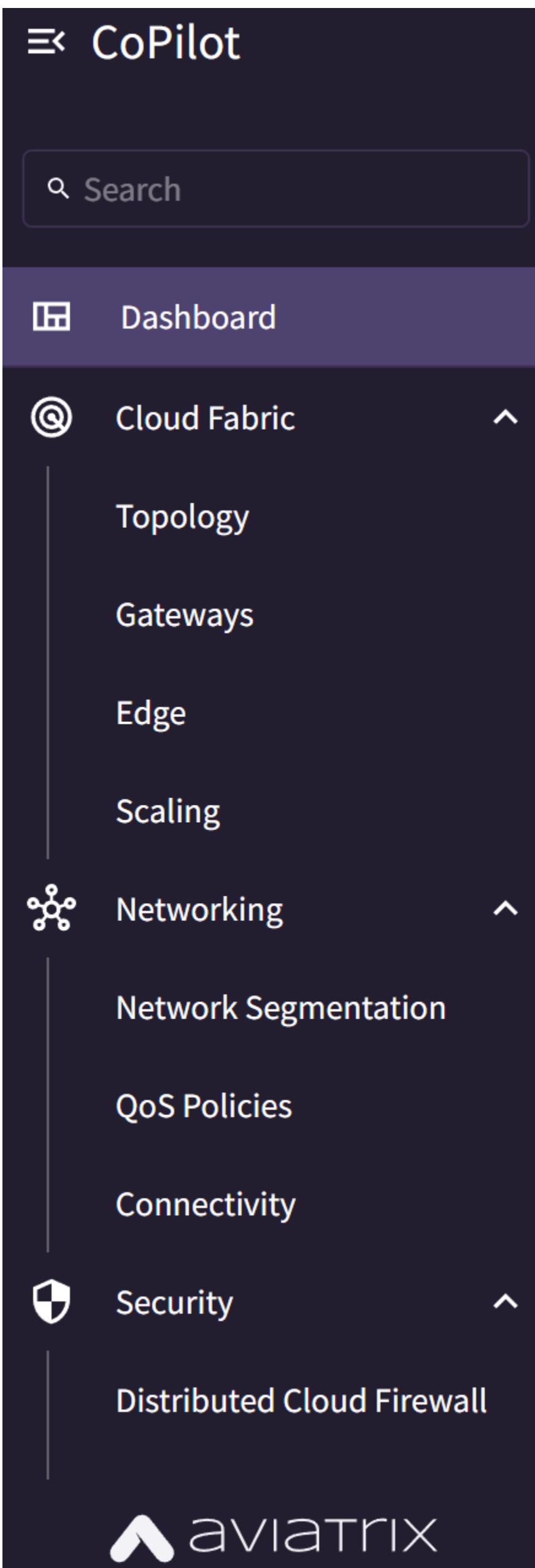
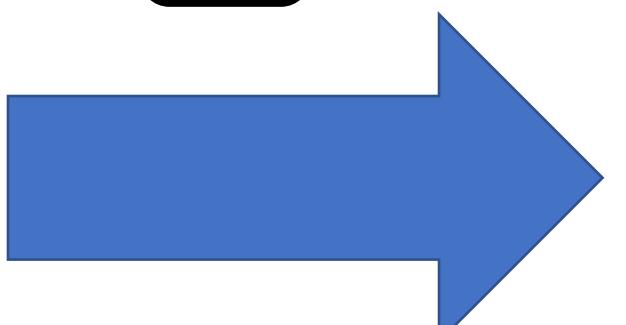
## Lab 2: Step 2.10

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.

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



## Lab 2: Step 2.11

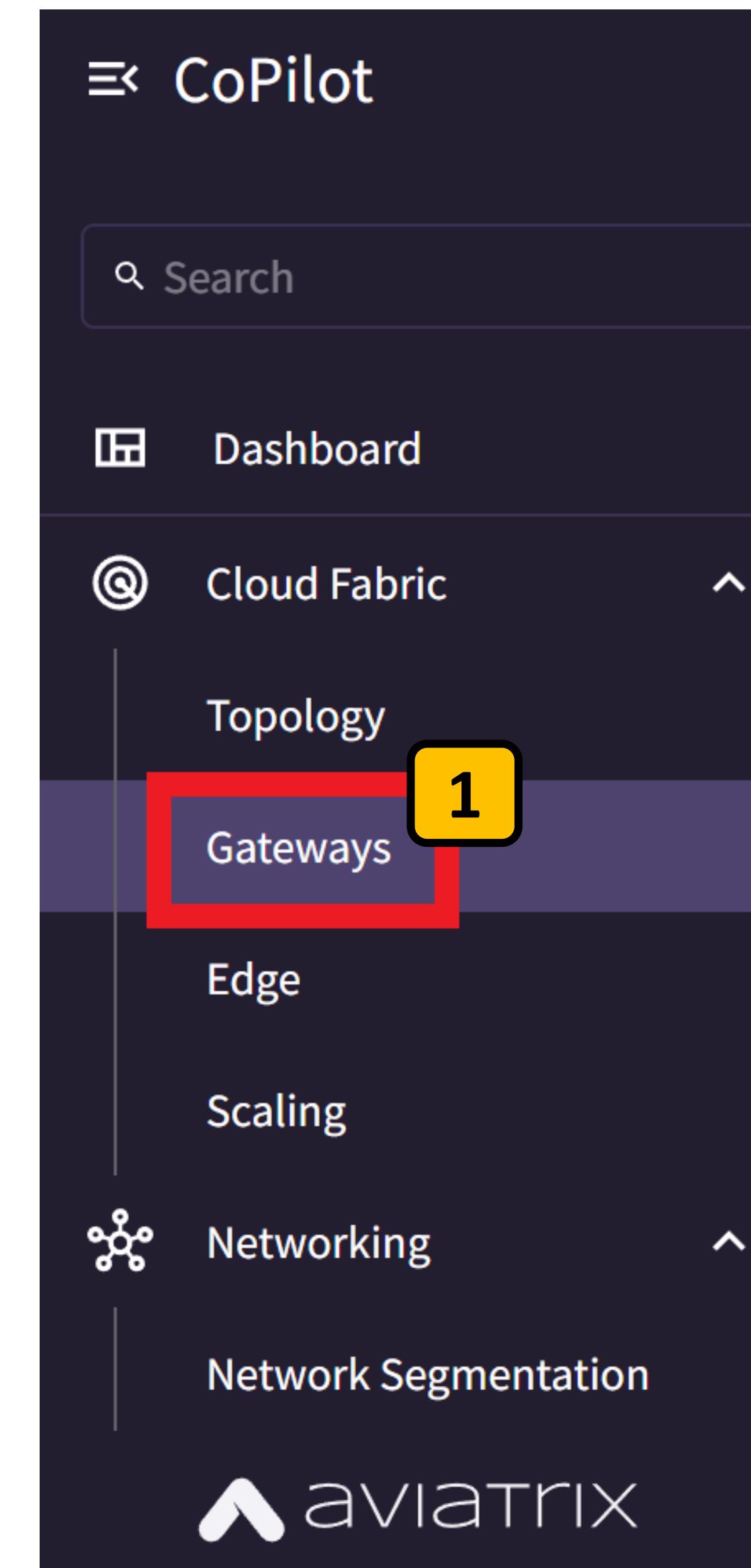
Deploy Aviatrix Spoke Gateway in VPC A Public Subnet

From the left-hand navigation in Aviatrix CoPilot...

Under Cloud Fabric select **Gateways** 1

Select the **Spoke Gateways** tab 2

Click the **+Spoke Gateway** 3 button to deploy a new Spoke Gateway.



The image shows the Aviatrix CoPilot interface with the 'Spoke Gateways' tab selected (highlighted with a red box and a yellow box containing '2'). The main area displays a table of existing Spoke Gateways:

Name	Region	VPC/VNet	Subnet CIDR
aws-us-west-2-spoke-1	us-west-2	vpc-0dff...	10.51.0.32/28
aws-us-west-2-spoke-2	us-west-2	vpc-04fd9...	10.52.0.32/28
aws-us-west-2-spoke-3	us-west-2	vpc-0270...	10.53.0.32/28

Total 3 Spoke Gateways

In the top right corner of the interface, there are two small icons: a speaker and a bell, each with a yellow box containing the number '2'. In the center of the top bar, there are three tabs: 'Gateways' (disabled), 'Overview', and 'Spoke Gateways' (selected). Below the tabs is a search bar and a row of filter icons.

## Lab 2: Step 2.12

Deploy Aviatrix Spoke Gateway in VPC A Public Subnet

Create Spoke Gateway

1 Name: aws-us-east-1-SpokeA

2 Cloud: AWS Standard

3 Account: aws-account

4 Region: us-east-1 (N. Virginia)

5 VPC/VNet: VPC A

6 Instance Size: t3.medium

High Performance Encryption: Off

Attach To Transit Gateway: Optional

The screenshot shows the 'Create Spoke Gateway' interface. The 'Name' field contains 'aws-us-east-1-SpokeA'. The 'Cloud' section shows 'AWS Standard' selected. The 'Account' field contains 'aws-account'. The 'Region' dropdown is set to 'us-east-1 (N. Virginia)'. The 'VPC/VNet' dropdown is set to 'VPC A'. The 'Instance Size' dropdown is set to 't3.medium'. A 'High Performance Encryption' toggle switch is set to 'Off'. An 'Optional' dropdown for 'Attach To Transit Gateway' is visible. Step numbers 1 through 6 are overlaid on the interface: 1 is on the 'Name' field, 2 is on the 'Cloud' section, 3 is on the 'Account' field, 4 is on the 'Region' dropdown, 5 is on the 'VPC/VNet' dropdown, and 6 is on the 'Instance Size' dropdown.

Name the gateway **aws-us-east-1-SpokeA** 1

Select the cloud AWS standard 2

Select the account **aws-account** 3

Select the region **us-east-1** 4

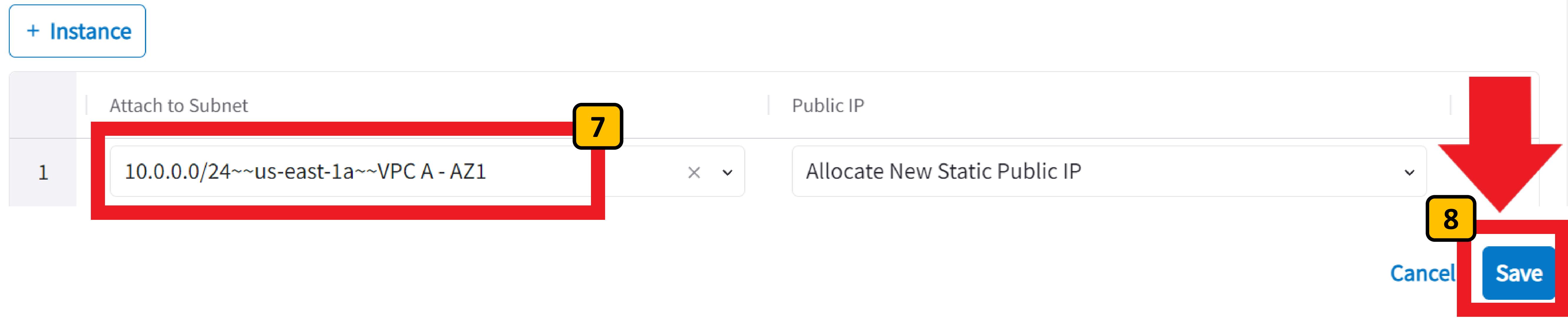
Select VPC A 5

Select the **t3.medium** instance size 6

## Lab 2: Step 2.13

Deploy Aviatrix Spoke Gateway in VPC A Public Subnet

Instances



Select the 10.0.0.0/24 subnet in AZ1 **7**

Leave the Public IP at the default selection of Allocate New

Click **Save** **8**

Upon clicking Save, Aviatrix CoPilot will begin the gateway deployment

## Lab 2: Step 2.14

Monitor the gateway deployment Task

The screenshot shows the AviatrIX CoPilot interface. On the left is a sidebar with the following items:

- CoPilot (selected)
- task (highlighted with a red box and yellow box labeled 1)
- Monitor (highlighted with a red box and yellow box labeled 2)
- Notifications / Tasks (highlighted with a red box and yellow box labeled 2)
- Settings
- Resources
- Task

The main area has tabs: Notifications, Alerts, Alerts Configuration, System Messages, Tasks (selected), and Recipients. Below is a search bar and a table titled "Active Gateway Operations". The table has columns: Name, Entity, Status, and Progress. It contains two rows:

Name	Entity	Status	Progress
Create spoke gateway: aws-us-east-1-SpokeA		Completed	[Progress Bar]
Create primary gateway		Completed	[Progress Bar]

Total 1 task

From the CoPilot search bar type **task** 1

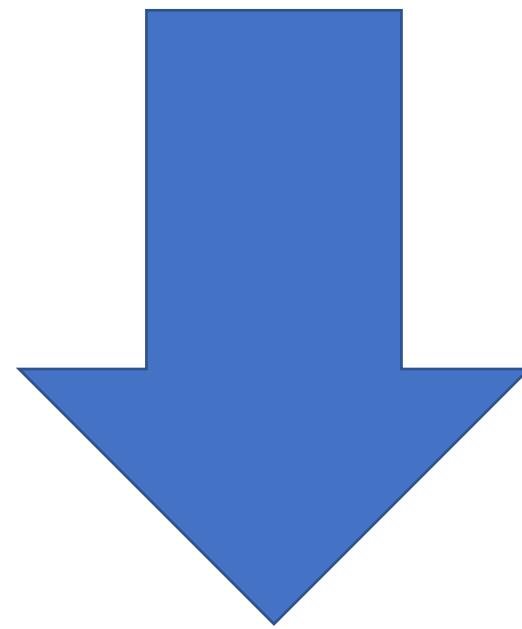
Click the search result **Notifications / Tasks** 2

Observe the spoke gateway creation Task and wait for it to complete 3

## Lab 2: Step 2.15

### Enable Egress on the AviatrIX Spoke Gateway

Now let's configure your new AviatrIX Spoke Gateway to do Local Egress for VPC A, so we can use it for egress NAT from our private subnets.



The screenshot shows the AviatrIX CoPilot interface. On the left, there is a search bar labeled "CoPilot" with a red box around it and a yellow box with the number "1" over the word "egress". Below the search bar is a "Security" section with a shield icon, followed by "Egress" and "Egress VPC/VNets", both highlighted with red boxes and yellow boxes with numbers "2" and "3" respectively. On the right, there is a navigation bar with tabs: "Egress" (highlighted with a red box and yellow box with number "3"), "Overview", "Monitor", and "Egress VPC/VNets" (also highlighted with a red box). Below the navigation bar is a button labeled "+ Local Egress on VPC/VNets". The main area displays a table with four rows of data:

Name	Point of Egress	Transit Attachment
aws-us-west-2-spoke-1	Local Egress	aws-us-west-2-transit
aws-us-west-2-spoke-2	Local Egress	aws-us-west-2-transit
aws-us-west-2-spoke-3	Local Egress	aws-us-west-2-transit

Below the table, it says "Total 4 VPC/VNets". A blue arrow points from the "Native Cloud Egress" entry in the table to a callout box.

From the CoPilot search bar type **egress** **1**

Click the search result **Egress VPC/VNets** **2**

Click the **+ Local Egress on VPC/VNets** button **3**

Notice how CoPilot is telling us that VPC A is currently using “Native Cloud Egress”

Because this VPC is currently using AWS NAT Gateway

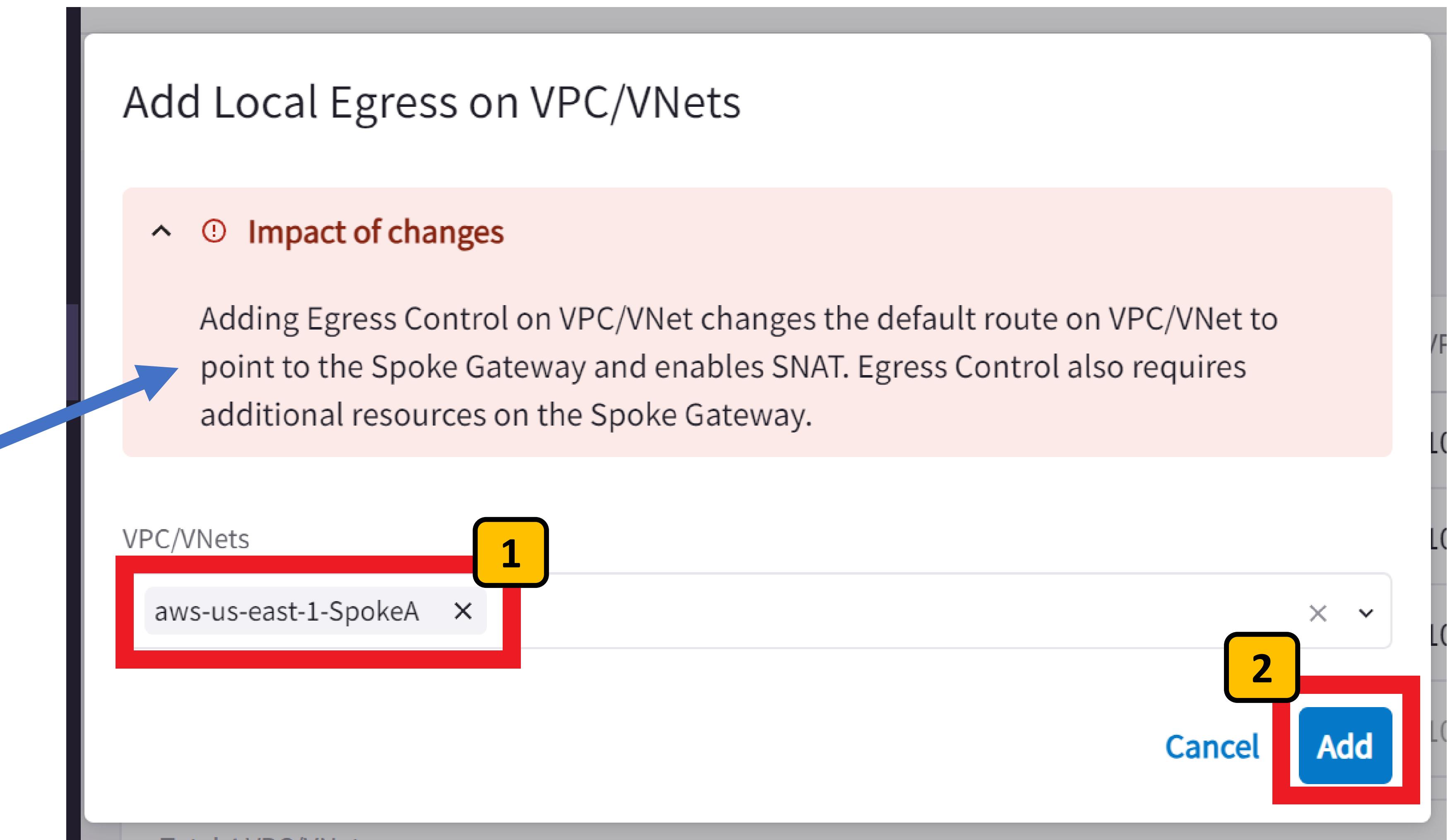
## Lab 2: Step 2.16

Enable Egress on the Aviatrix Spoke Gateway

In the Add Local Egress on VPC/VNets pop-up, select your new **aws-us-east-1-SpokeA** gateway from the VPC/VNets drop down **1**

Click **Add** **2**

After you click Add, Copilot will change the VPC default route associated to all private subnets in VPC A to point to your new Aviatrix Gateway



## Lab 2: Step 2.17

Observe the default route of your VPC A Private Route Table

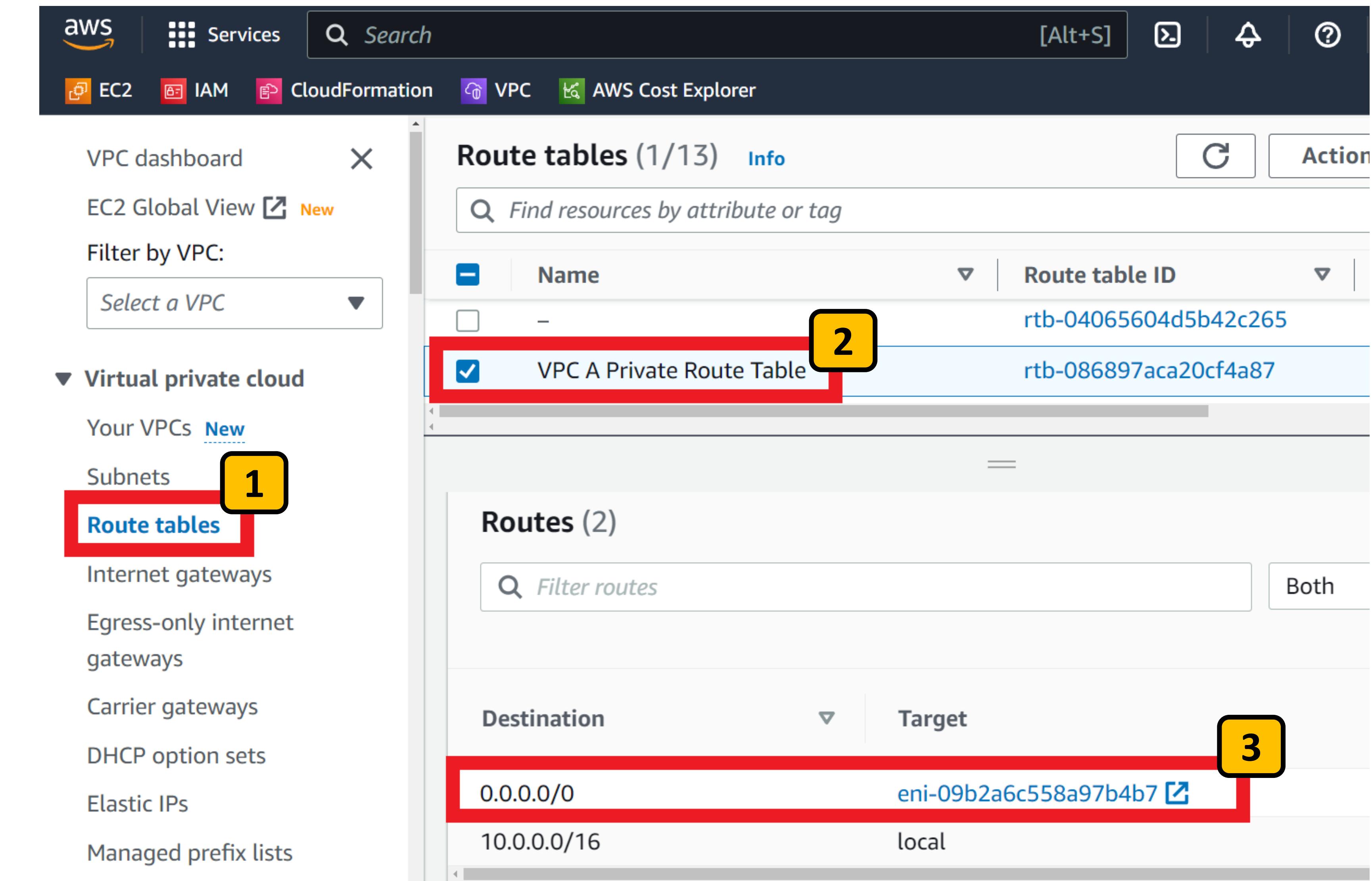
After enabling Egress on the Aviatrix Gateway, the Aviatrix Controller changed the default route in your VPC private route table.

From the AWS Console go to the VPC section of the console and select **Route tables** 1

Select the **VPC A Private Route Table** 2

Select the **Routes** tab

Observe the 0.0.0.0/0 default route directing traffic to Aviatrix Gateway 3



## Lab 2: Checkpoint 2

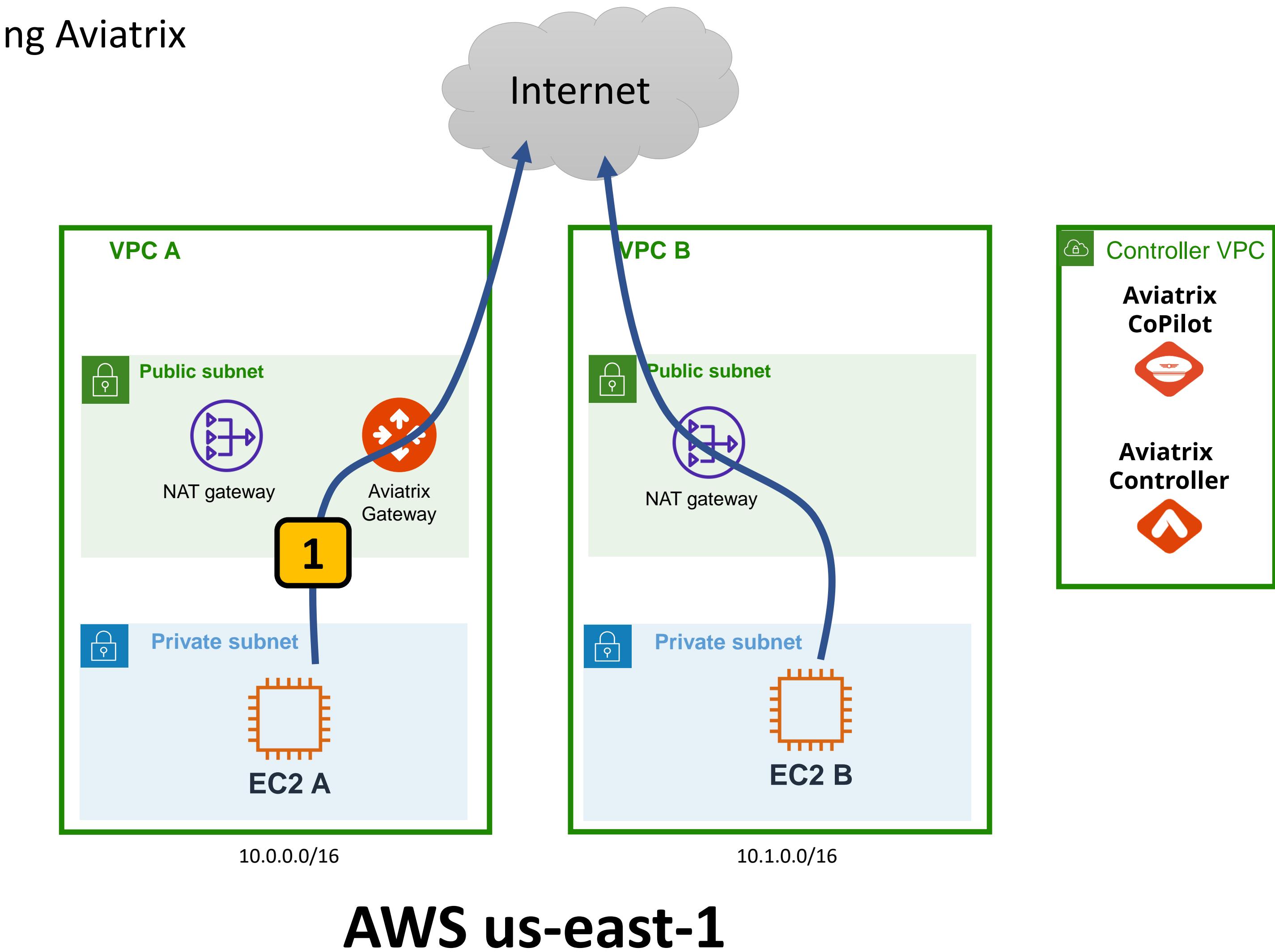
Simple egress NAT using Aviatrix

At this point you've deployed an Aviatrix Gateway in VPC A

You configured the Aviatrix Gateway for Egress

You observed the VPC A Private Route Table for private subnets has been changed to use the Aviatrix Gateway for egress. **1**

**Next, let's configure security rules for egress traffic using Aviatrix Distributed Firewall**



**AWS us-east-1**

**Note:** Aviatrix does not charge data processing charges for NAT. The Aviatrix cost for simple egress NAT is \$0.14 /hr per gateway (plus the EC2 gateway instance charges)

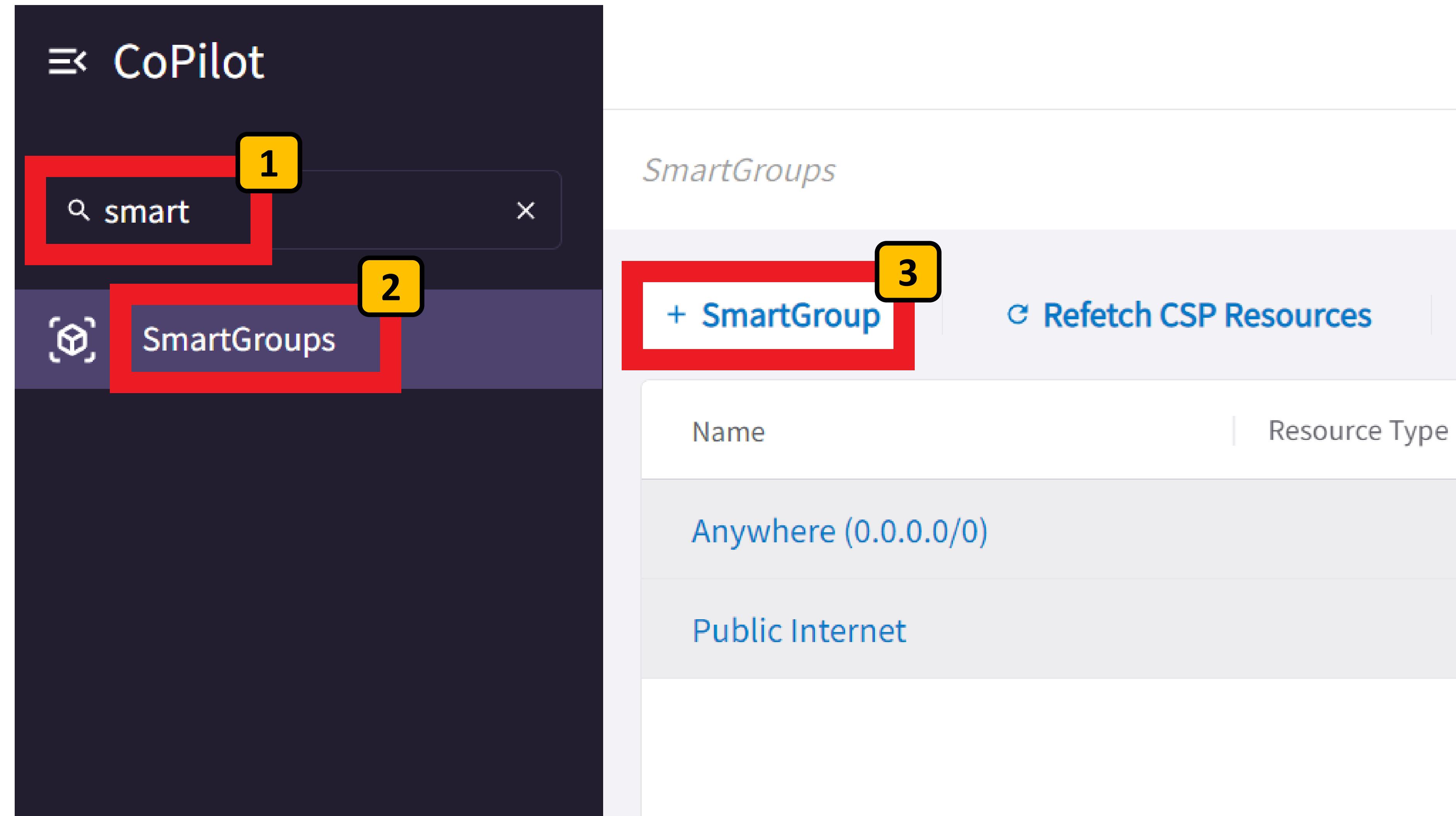
## Lab 2: Step 2.18

Create a SmartGroup to use for firewall rules

From the CoPilot search bar type **smart** 1

Click the search result **SmartGroup** 2

Click the **+SmartGroup** button 3



## Lab 2: Step 2.19

Create a SmartGroup to use for firewall rules

Name the SmartGroup **DEV** 1

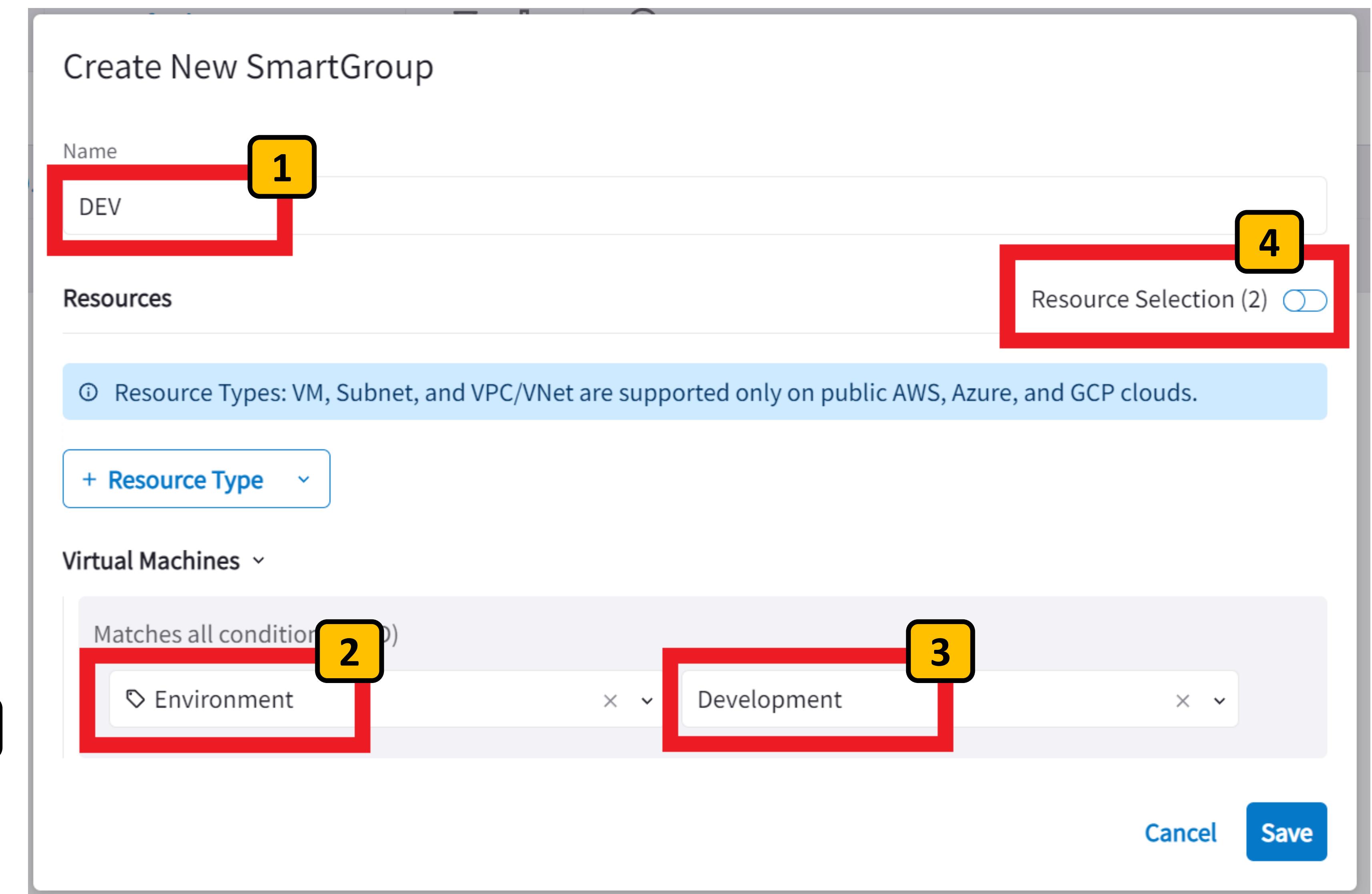
We want all EC2 instances tagged  
**Environment = Development** to be in  
this SmartGroup.

The instances EC2 A and EC2 B have  
already been tagged this way.

Select the CSP Tag key **Environment** 2

Select the value **Development** 3

Click **Resource Selection** to confirm 4



## Lab 2: Step 2.20

Create a SmartGroup to use for firewall rules

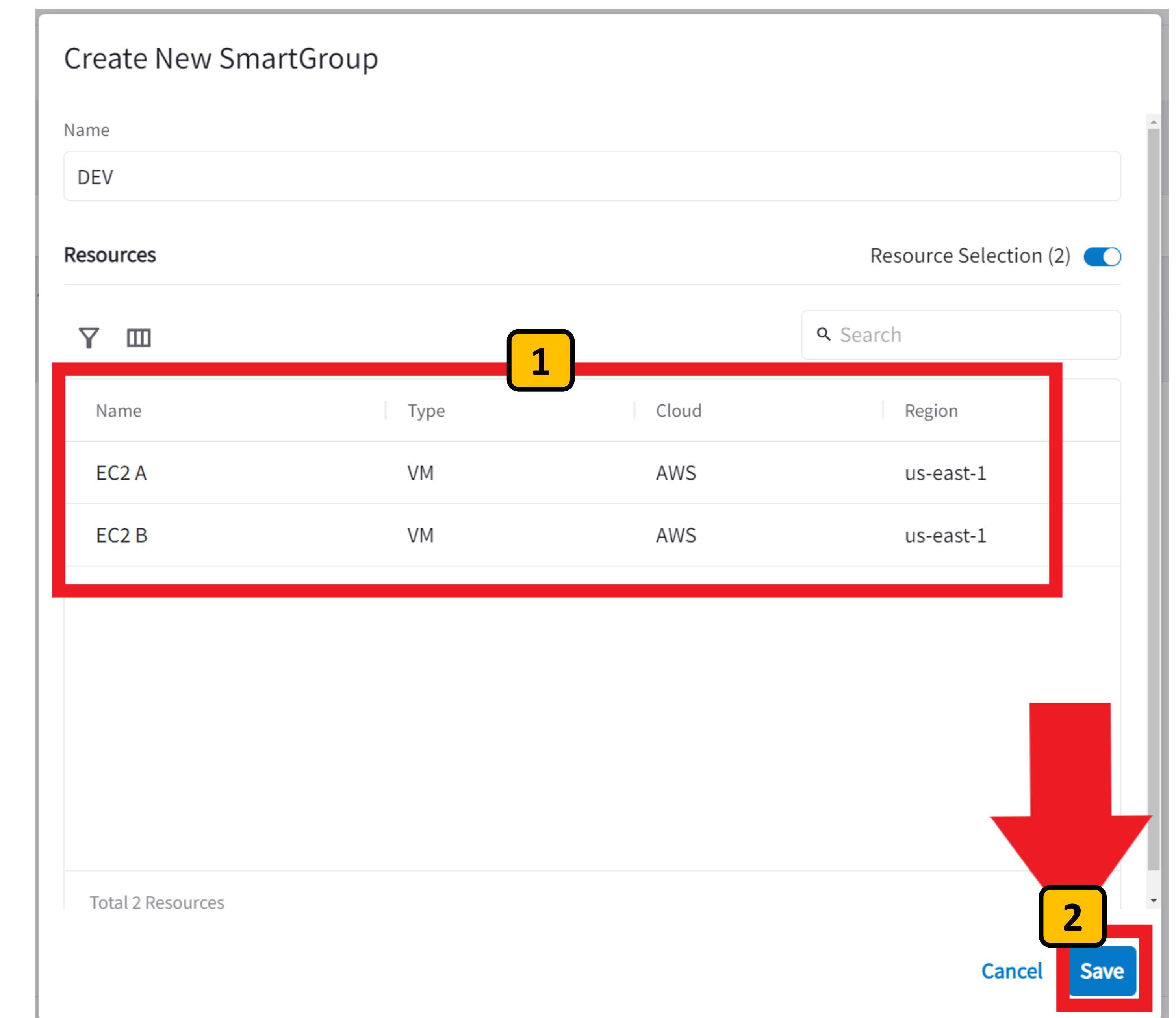
Aviatrix CoPilot already knows that your instances EC2 A and EC2 B were tagged with the EC2 tags

**Environment = Development**

Confirm that both EC2 A and EC2 B instances match the SmartGroup **1**

Click **Save** to create the SmartGroup **2**

We'll use this SmartGroup to create firewall rules



## Lab 2: Step 2.21

Create a SmartGroup to use for firewall rules

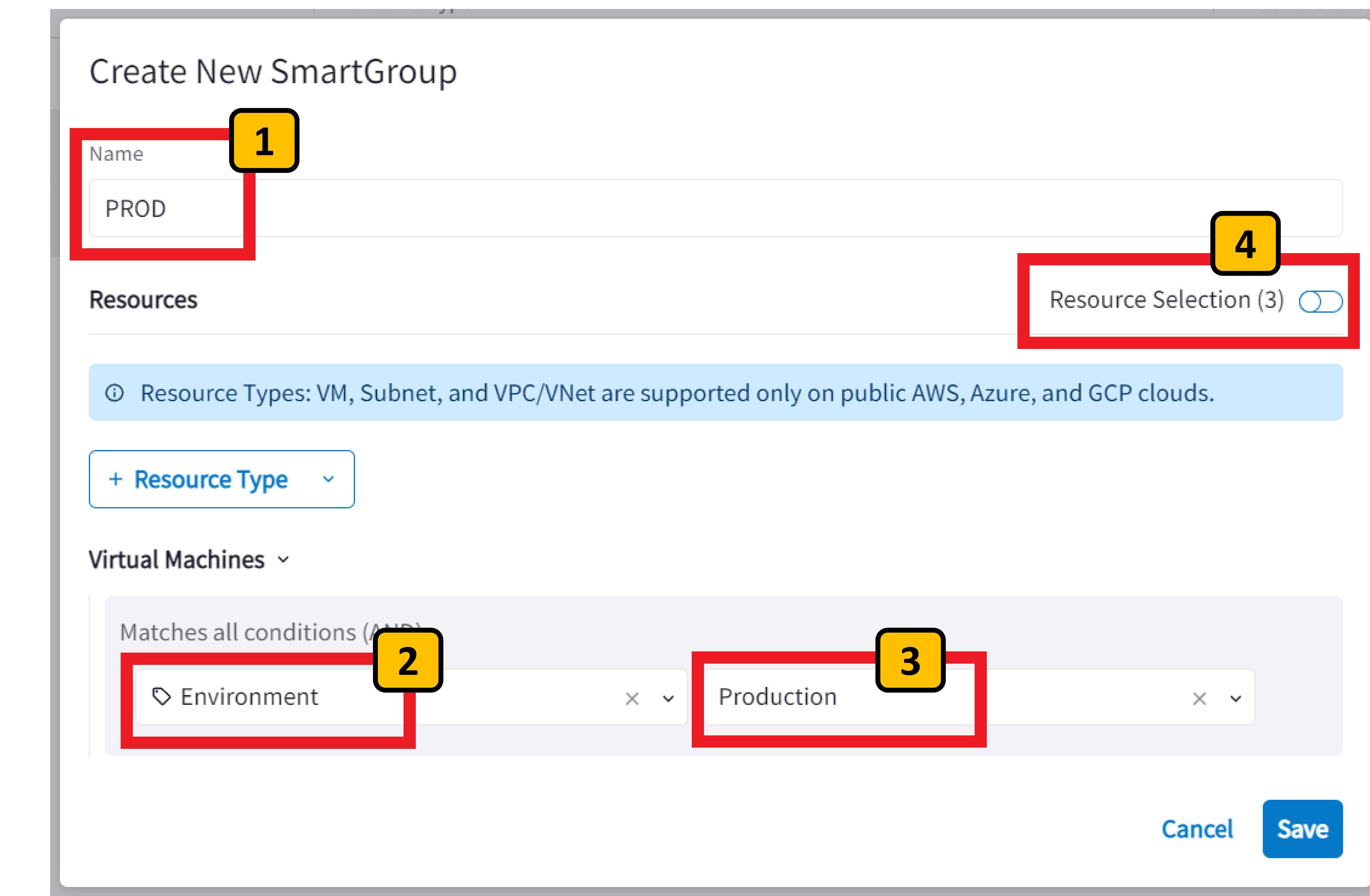
Create another SmartGroup and name it PROD 1

We want all EC2 instances tagged **Environment = Production** to be in this SmartGroup.

Select the CSP Tag key **Environment** 2

Select the value **Production** 3

Click **Resource Selection** to confirm 4



## Lab 2: Step 2.22

Create a SmartGroup to use for firewall rules

Aviatrix CoPilot already knows your instances tagged with the EC2 tags

**Environment = Production**

Confirm that both there are three instances named SAP in the us-west-2 region in the SmartGroup **1**

Click **Save** to create the SmartGroup **2**

Create New SmartGroup

Name  
PROD

Resources

Resource Selection (3)

Name	Type	Cloud	Region
SAP1	VM	AWS	us-west-2
SAP2	VM	AWS	us-west-2
SAP3	VM	AWS	us-west-2

Total 3 Resources

Cancel **2** Save

## Lab 2: Step 2.23

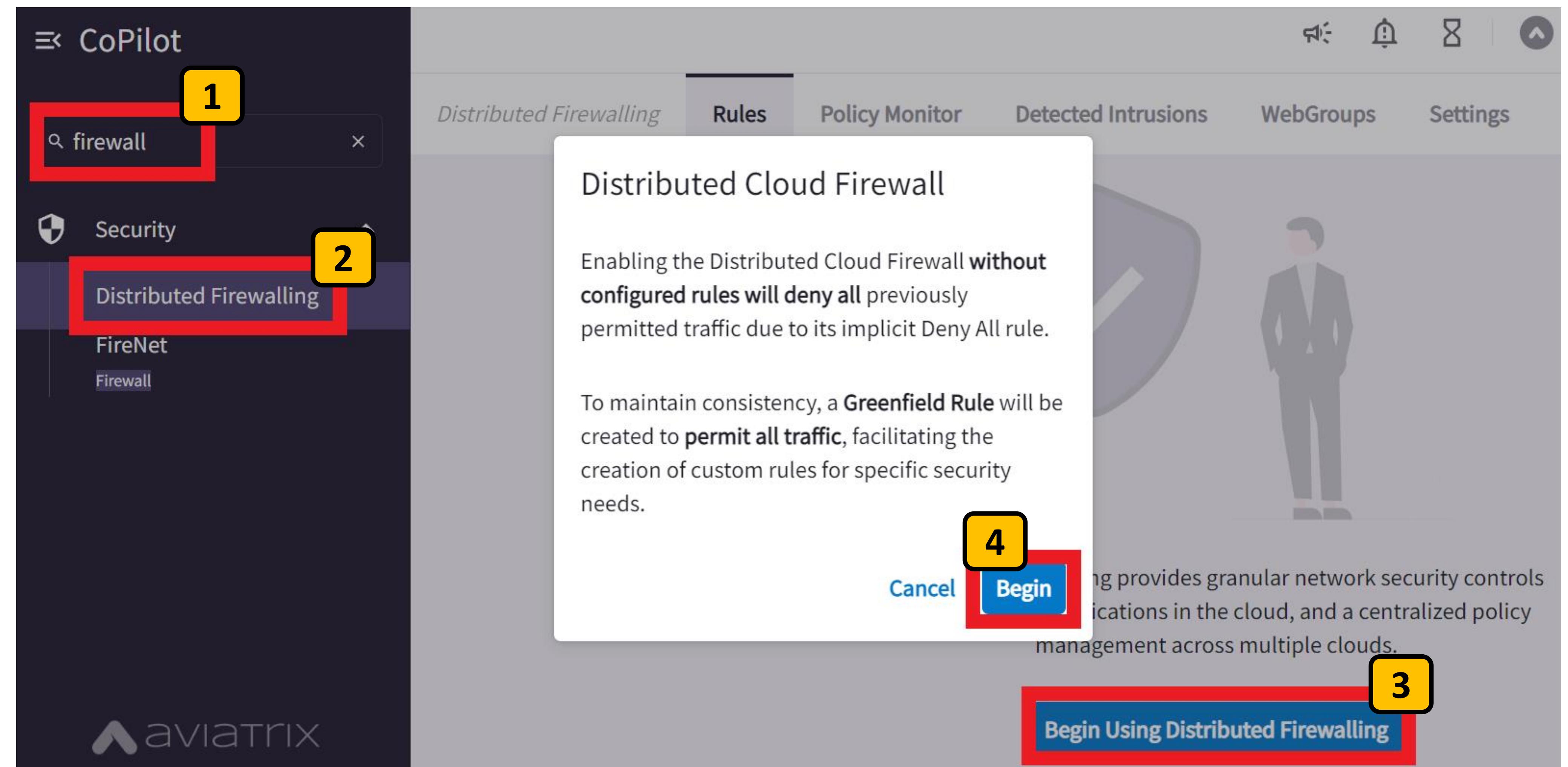
Enable Aviatrix Distributed Firewalling

From the CoPilot search bar type **firewall** 1

Click the search result **Distributed Firewalling** 2

Click **Begin Using Distributed Firewalling** 3

In the pop-up dialog click **Begin** 4



## Lab 2: Step 2.24

Create a WebGroup to use for firewall rules

The screenshot shows the Aviatrix CoPilot interface. On the left, there's a sidebar with a search bar containing 'firewall'. Below it, under 'Security', are 'Distributed Firewalling' and 'FireNet Firewall'. The main area is titled 'Distributed Firewalling' and contains tabs for 'Rules', 'Policy Monitor', 'Detected Intrusions', and 'WebGroups'. The 'WebGroups' tab is highlighted with a red box and a yellow box labeled '1'. Below it, a button labeled '+ WebGroup' is also highlighted with a red box and a yellow box labeled '2'. The table below shows one entry: 'Any-Web' (Predefined WebGroup). The table has columns for Name, Type, Domains/URLs, and Rules.

Name	Type	Domains/URLs	Rules
Any-Web	Predefined WebGroup	*	0

On the Distributed Firewall click the **WebGroup** **1**

Click the button **+ WebGroup** to create a new WebGroup **2**

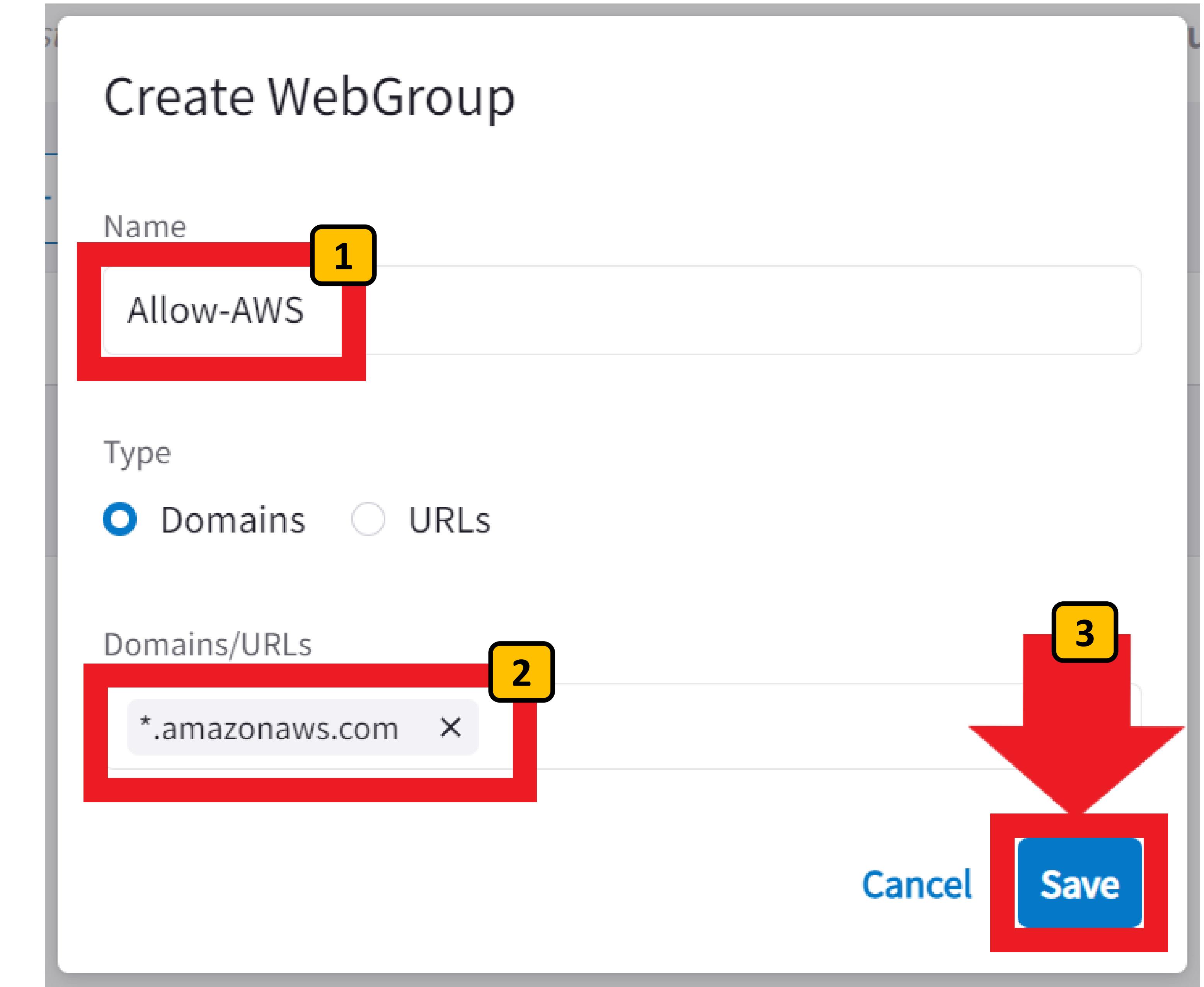
## Lab 2: Step 2.25

Create a WebGroup to use for firewall rules

Name the WebGroup **Allow-AWS** 1

Enter the domain name  
**\*.amazonaws.com** 2

Click **Save** 3



## Lab 2: Step 2.26

Create a Distributed Firewall Rule

The screenshot shows the Aviatr ix CoPilot interface. On the left, there's a sidebar with a search bar containing 'firewall'. Below it are sections for 'Security' (with a shield icon), 'Distributed Firewalling' (which is selected and highlighted in purple), and 'FireNet Firewall'. The main area is titled 'Distributed Firewalling' and contains three tabs: 'Rules' (highlighted with a red box and yellow number 1), 'Policy Monitor', and 'Detected Intrusions'. Below the tabs is a table with columns: Priority, Name, and Source. A single row is visible, labeled '2147...' with a checkmark, 'Greenfield-Rule', and 'Anywhere (0.0.0.0/0)'. At the bottom of the main area, there's a toolbar with a '+ Rule' button (highlighted with a red box and yellow number 2), 'Actions' dropdown, filter icon, and sorting icons.

Priority	Name	Source
<input type="checkbox"/>	2147... Greenfield-Rule	Anywhere (0.0.0.0/0)

On the Distributed Firewall click the **Rules** tab **1**

Click the button **+ Rule** to create a new distributed firewall rule **2**

## Lab 2: Step 2.27

Create a firewall rule

Name the rule **Allow-NTP** 1

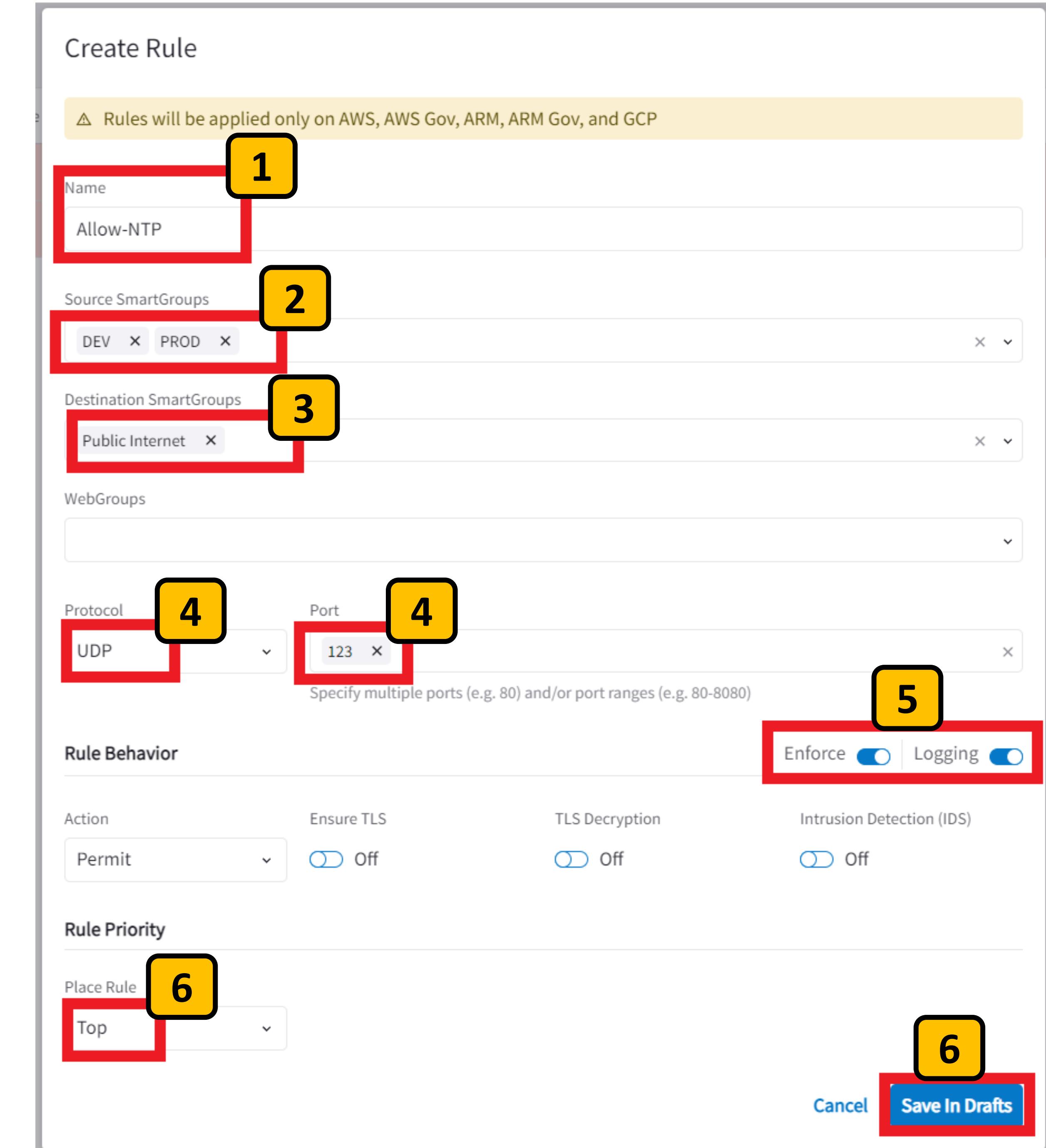
Chose the Source SmartGroups **DEV** and **PROD** 2

Choose the Destination SmartGroup **Public Internet** 3

Choose the Protocol **UDP** and enter Port number **123** 4

Enable Enforce and Logging 5

Place Rule at Top and click **Save In Drafts** 6



## Lab 2: Step 2.28

Create a firewall rule for AWS domains

Create another rule named **Allow-AWS** 1

Chose the Source SmartGroups **DEV** and **PROD** 2

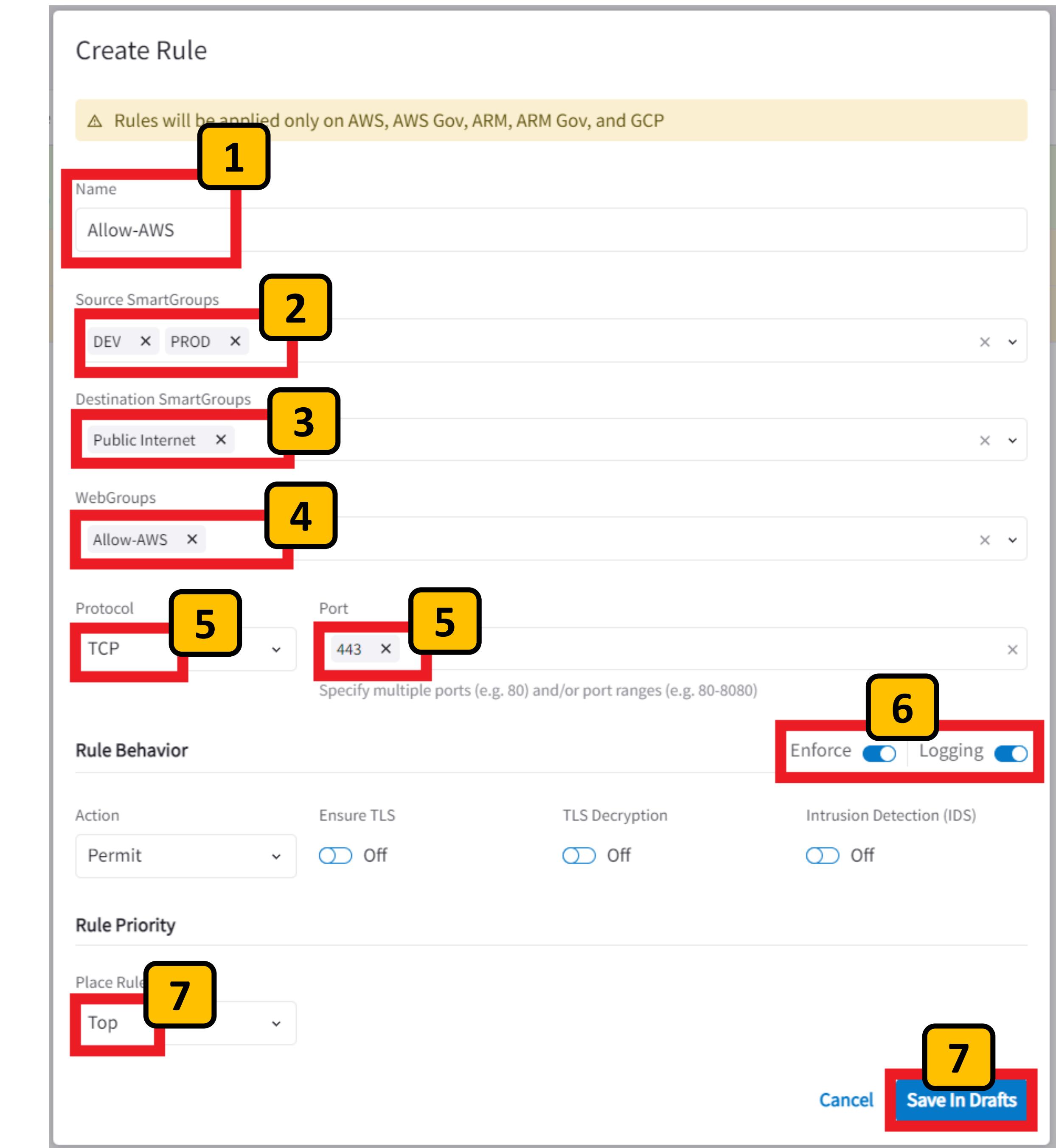
Choose the Destination SmartGroup **Public Internet** 3

Choose the WebGroup **Allow-AWS** 4

Choose the Protocol **TCP** and enter Port number **443** 5

Enable Enforce and Logging 6

Place Rule at Top and click **Save In Drafts** 7



## Lab 2: Step 2.29

Delete the default permit any rule

The screenshot shows the Aviatrix Distributed Firewalling Rules interface. The top navigation bar includes tabs for *Distributed Firewalling*, **Rules**, *Policy Monitor*, *Detected Intrusions*, *WebGroups*, and *Settings*. Below the tabs is a toolbar with buttons for **+ Rule**, **Actions**, sorting, and filtering, along with status indicators for **2 New** and **1 Modified**, and buttons for **Discard** and **Commit**. A search bar is also present.

The main table lists three rules:

Priority	Name	Source	Destination	WebGroup	Actions	
0	Allow-AWS	DEV	Public Internet	Allow-AWS		
1	Allow-NTP	DEV	Public Internet			
2147...	Greenfield-Rule	Anywhere (0.0.0.0/0)	Anywhere (0.0.0.0/0)			

A context menu is open over the **Greenfield-Rule** row, listing options: **Reset Traffic Count**, **Turn Off Enforcement**, **Turn On Logging**, **Delete Rule**, and **Discard Change**. The **Delete Rule** option is highlighted with a red box and a yellow box labeled **2**.

Find the **Greenfield-Rule** that permits all traffic and delete it

Select the **three vertical dots** on the rule **1**

In the drop-down select **Delete Rule** **2**

## Lab 2: Step 2.30

Commit your firewall rules

The screenshot shows the Aviatrix CoPilot interface for managing distributed firewall rules. On the left, there's a sidebar with a search bar containing 'firewall', a 'Security' section, and a 'Distributed Firewalling' section where 'FireNet Firewall' is selected. The main area has tabs for 'Distributed Firewalling', 'Rules' (which is active), 'Policy Monitor', 'Detected Intrusions', 'WebGroups', and 'Settings'. Below the tabs is a toolbar with '+ Rule', 'Actions', and status indicators for '2 New' and '1 Deleted'. The main table lists three rules:

Priority	Name	Source	Destination	WebGroup
0	Allow-AWS	DEV	Public Internet	Allow-AWS
1	Allow-NTP	DEV	Public Internet	
2147...	Greenfield-Rule	Anywhere (0.0.0.0/0)	Anywhere (0.0.0.0/0)	

A large red arrow with the number '1' above it points to the blue 'Commit' button in the toolbar.

Commit your distributed firewall rule configurations by clicking **Commit** 1

Once committed, Aviatrix will write the rules to the Aviatrix Gateway(s)

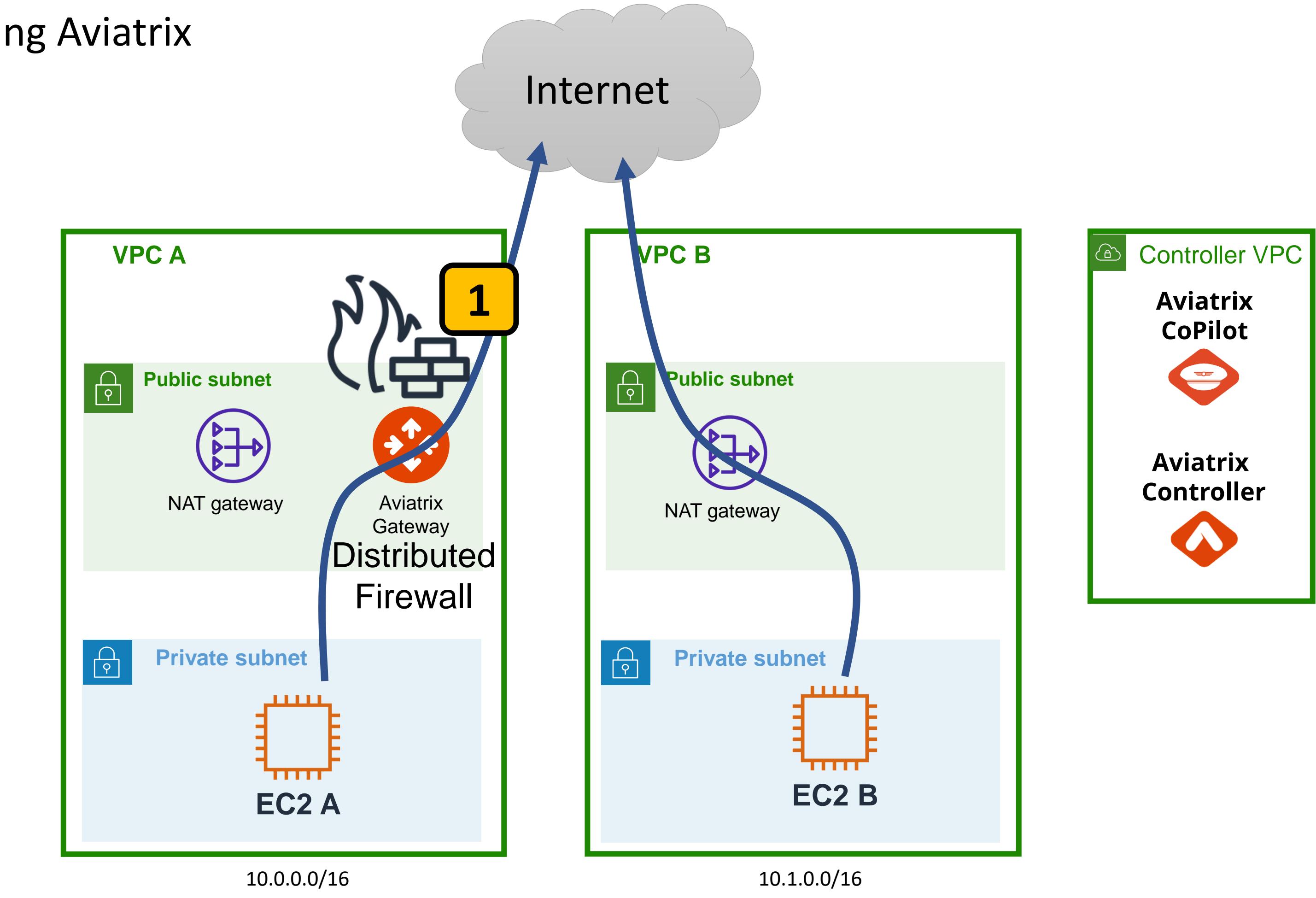
## Lab 2: Checkpoint 3

Simple egress NAT using Aviatrix

At this point you just deployed the Aviatrix Distributed Firewall on your Aviatrix Gateway in VPC A. **1**

Your EC2 A instance will only be able access the internet according to the firewall rules you created.

**Next: Let's test the internet again from the instance EC2 A and see if we have secured our egress traffic using the Aviatrix Distributed Firewall.**



**AWS us-east-1**

Note: Aviatrix does not charge data processing charges for Firewall and NAT. The Aviatrix cost for Distributed Firewall and NAT is **\$0.23 /hr per gateway** (plus the EC2 gateway instance charges)

## Lab 2: Step 2.31

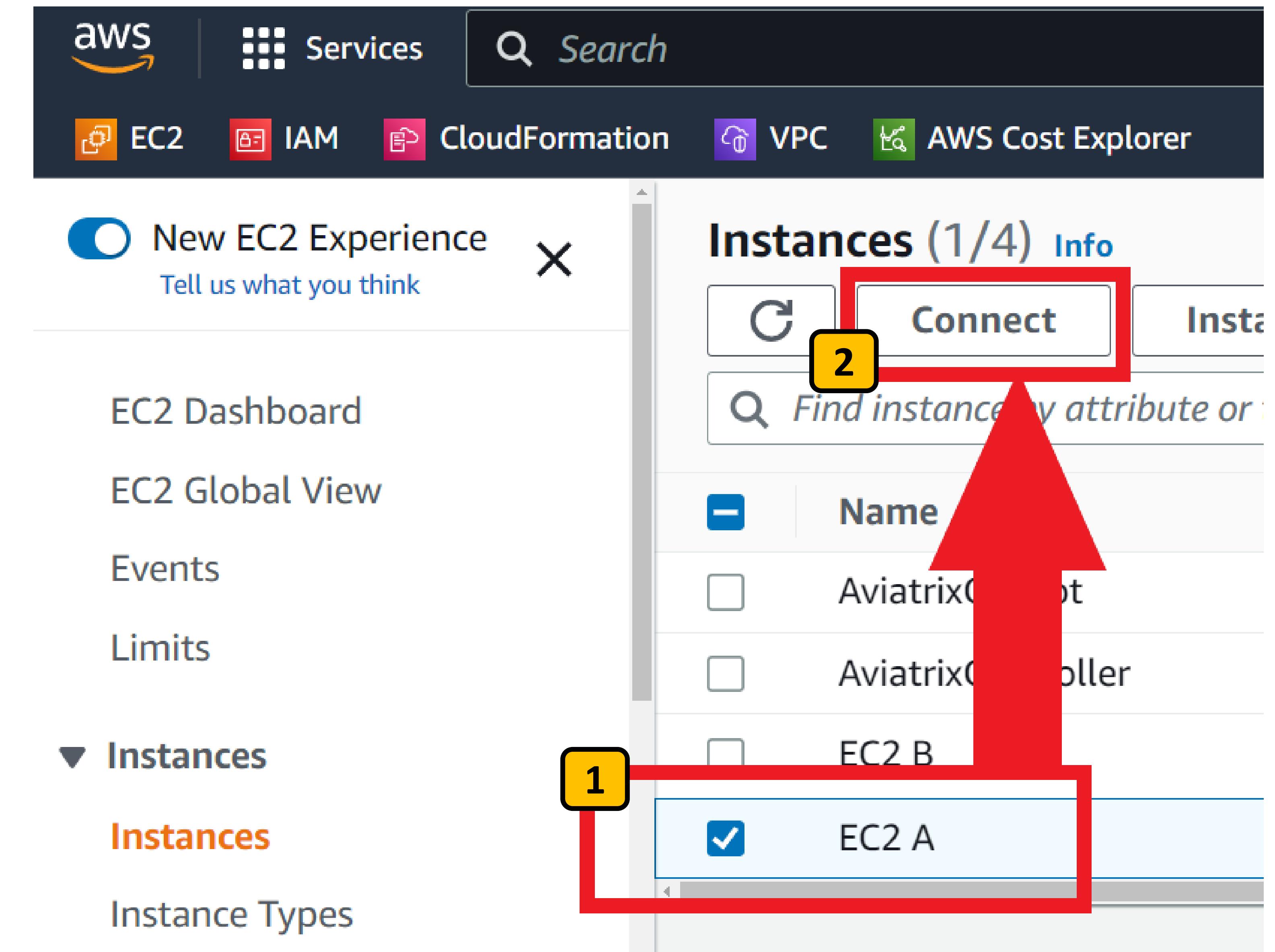
Connect to EC2 A instance console

Go the EC2 section of the AWS Console.

Select Instances

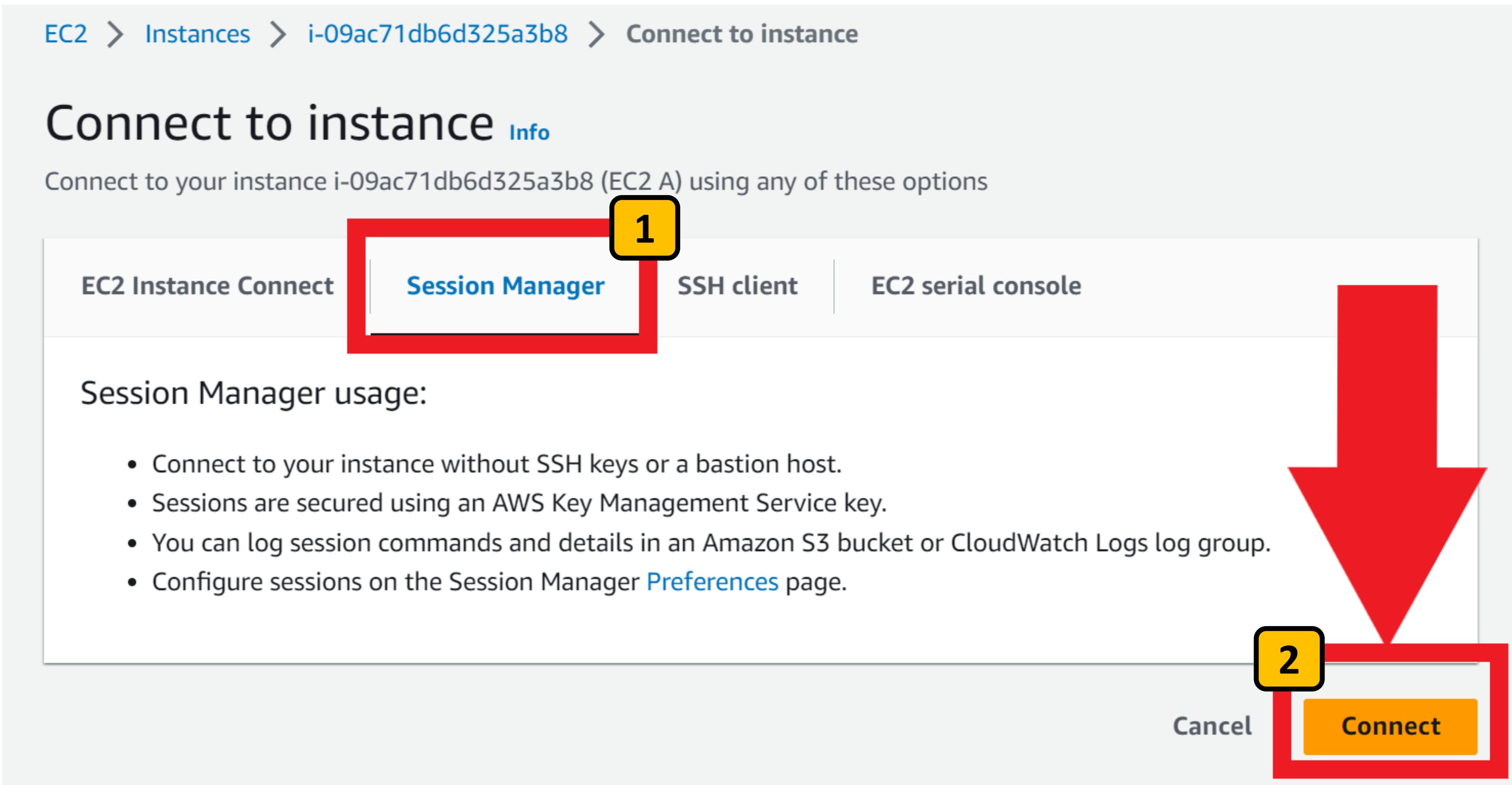
Find the EC2 A instance and select it. **1**

Click the Connect button **2**



## Lab 2: Step 2.32

Connect to EC2 A instance console



Select the Session Manager tab. **1**

Click the Connect button **2**

**Note:** If you get an error here, try rebooting EC2 A to accelerate the process of Session Manager reconnecting through the Aviatrix Gateway

## Lab 2: Step 2.34

Connect to EC2 A instance console

Your browser should open a new tab giving you a CLI session.

Type the command:

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

(dash lower case L)

You are now logged on as the ec2-user and should see your private IP address in the hostname.

The screenshot shows a terminal window with a black background and white text. At the top, the URL is visible: us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-09ac71db6d325a3b8?region=us-east-1. Below the URL, the session information is displayed: Session ID: brad-000841717aa667ae4 and Instance ID: i-09ac71db6d325a3b8. The terminal prompt is sh-4.2\$. The user then types the command sudo su -l ec2-user. A red rectangular box highlights the command, and a yellow box labeled '1' is positioned over the number 1 in the sequence. The terminal then displays the output [ec2-user@ip-10-0-2-10 ~]\$ five times, indicating a loop or a successful login. A vertical scroll bar is visible on the right side of the terminal window.

Next, let's test internet egress.

Test that your EC2 A instance has secured internet access that will prevent it from connecting to potentially harmful domains.

Run the commands:

**curl https://ransomware.org** 1

**curl https://malware.net** 2

**curl https://botnet.com** 3

For each command you should see the CLI return an SSL Error, because the Aviatrix Distributed Firewall is blocking the connection 4

## Lab 2: Step 2.35

Test connections to potentially harmful domains

The image contains four screenshots of the AWS Systems Manager Session Manager interface, showing terminal sessions on an EC2 instance. Each screenshot shows a command being run followed by an error message indicating an SSL/TLS handshake failure due to certificate verification errors.

- Screenshot 1:** Shows the command `curl https://ransomware.org` with the output:

```
[ec2-user@ip-10-0-2-10 ~]$ curl https://ransomware.org
```

A red box highlights the URL, and a yellow box labeled "1" highlights the error message.

```
curl: (35) OpenSSL/1.0.2k-fips: error:14077410:SSL routines:SSL23_GET_SERVER_HELLO
```
- Screenshot 2:** Shows the command `curl https://malware.net` with the output:

```
[ec2-user@ip-10-0-2-10 ~]$ curl https://malware.net
```

A red box highlights the URL, and a yellow box labeled "2" highlights the error message.

```
curl: (35) OpenSSL/1.0.2k-fips: error:14077410:SSL routines:SSL23_GET_SERVER_HELLO
```
- Screenshot 3:** Shows the command `curl https://botnet.com` with the output:

```
[ec2-user@ip-10-0-2-10 ~]$ curl https://botnet.com
```

A red box highlights the URL, and a yellow box labeled "3" highlights the error message.

```
curl: (35) OpenSSL/1.0.2k-fips: error:14077410:SSL routines:SSL23_GET_SERVER_HELLO
```
- Screenshot 4:** Shows the command `curl https://ransomware.org` with the output:

```
[ec2-user@ip-10-0-2-10 ~]$ curl https://ransomware.org
```

A large red X is overlaid on the error message, and a yellow box labeled "4" highlights the error message.

```
curl: (35) OpenSSL/1.0.2k-fips: error:14077410:SSL routines:SSL23_GET_SERVER_HELLO
```

## Lab 2: Step 2.36

Run a software update

Let's make sure we can still run a software update from the AWS repositories

Run the command:

**sudo yum update -y** 1

This will connect to **amazonaws.com** domains to download software updates.

Great! Software updates are working because of the Allow-AWS rule in the Aviatrix Distributed Firewall.

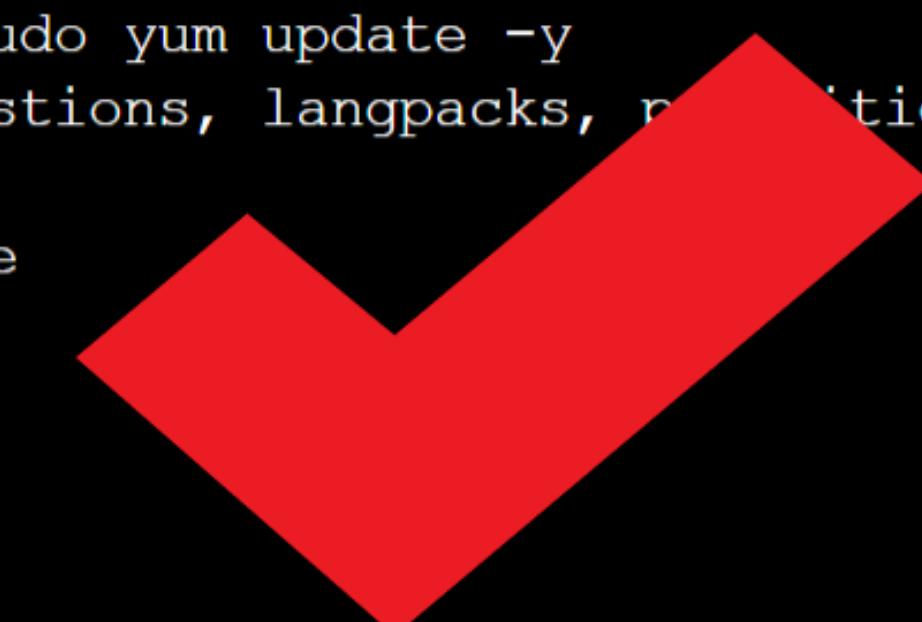
← → C [us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-09ac71db6d325a3b8?region=us-east-1](https://us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-09ac71db6d325a3b8?region=us-east-1)

Session ID: brad-000841717aa667ae4 Instance ID: i-09ac71db6d325a3b8

```
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$ sudo yum update -y1
```

Session ID: brad-06257fdad7c018c5c Instance ID: i-09ac71db6d325a3b8

```
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$ sudo yum update -y  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core  
No packages marked for update  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$  
[ec2-user@ip-10-0-2-10 ~]$
```



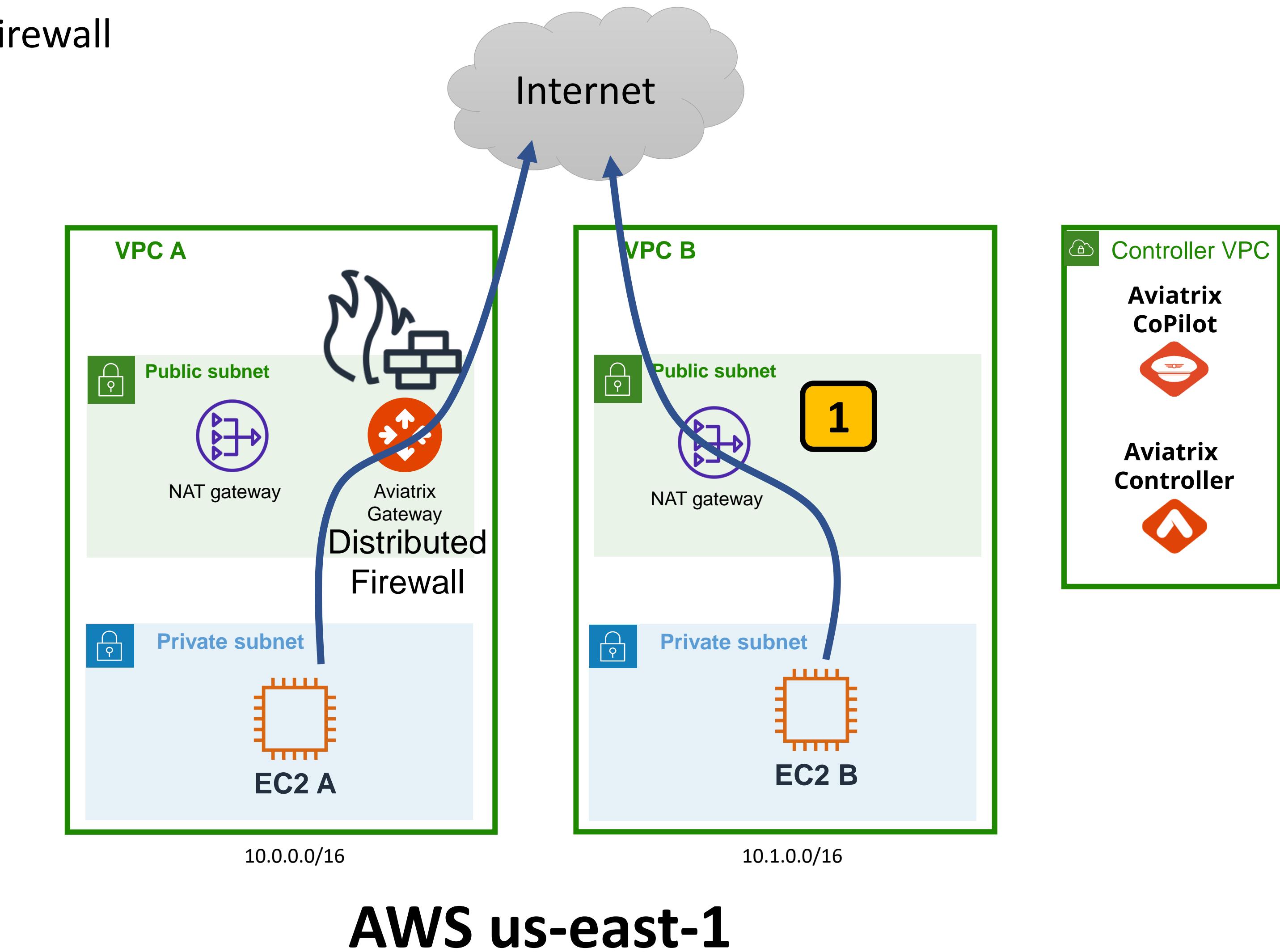
## Lab 2: Checkpoint 4

Aviatrix Distributed Firewall

At this point you've successfully deployed and tested the Aviatrix Distributed Firewall for VPC A.

Your EC2 A instance will only be able access the internet according to the firewall rules you created.

**Next: Let's extend the Distribute Firewall to VPC B, only this time we will use the same EIP that the AWS NAT Gateway is 1 using. Because maybe you have that EIP whitelisted with a partner or customer you work with, and you want to keep it.**



## Lab 2: Step 2.37

Delete your AWS NAT Gateways

The screenshot shows the AWS VPC NAT gateways page. The left sidebar has a red box around the 'NAT gateways' link, which is highlighted with a yellow box and the number 1. The main table lists two NAT gateways:

Name	NAT gateway ID	Connectivit...
NGW VPC A	nat-061c67d74aa528ced	Public
NGW VPC B	nat-009cc8d3d6a4a0611	Public

To the right of the table, the public IP address for NGW VPC B, 3.220.133.116, is highlighted with a red box and the number 2. An 'Actions' dropdown menu is open over NGW VPC B, with the 'Delete NAT gateway' option highlighted with a red box and the number 3.

Go to the VPC section of the AWS Console and select **NAT gateways** 1

Find the NAT Gateway for VPC B and take note of the public IP address 2

Select **NGW VPC B** and delete it by choosing **Actions** then **Delete NAT gateway** 3

Once the NAT Gateway is deleted, we can use its EIP for the Aviatrix Gateway

## Lab 2: Step 2.38

Delete your AWS NAT Gateways

✓ You successfully deleted nat-009cc8d3d6a4a0611 (NGW VPC B). X

NAT gateways (2) <span style="color: blue;">Info</span>		<span style="border: 1px solid #ccc; padding: 2px;">C</span>	Actions ▾	<span style="background-color: orange; color: white; padding: 2px 10px;">Create NAT gateway</span>
<span style="border: 1px solid #ccc; padding: 2px;">Filter NAT gateways</span>		<span style="float: right;">&lt; 1 &gt; </span>		
Name	NAT gateway ID	Connectivit...	State	State message
NGW VPC A	<a href="#">nat-061c67d74aa528ced</a>	Public	<span style="color: green;">Available</span>	-
NGW VPC B	<a href="#">nat-009cc8d3d6a4a0611</a>	Public	<span style="color: red;">Deleted</span>	<span style="border: 1px solid yellow; border-radius: 50%; padding: 2px 5px; text-align: center;">1</span>

Wait for the NAT Gateway for VPC B to show as Deleted 1

**Optional:** While you're waiting, you can also delete the NAT Gateway for VPC A, as it's not longer being used.

## Lab 2: Step 2.39

Deploy Aviatrix Spoke Gateway in VPC B Public Subnet

From the left-hand navigation in Aviatrix CoPilot...

Type the word **spoke** in the CoPilot search bar **1**

Select the **Spoke Gateways** search result **2**

Click the **+Spoke Gateway** button to deploy a new Spoke Gateway for VPC B. **3**

The screenshot shows the Aviatrix CoPilot interface. On the left, a sidebar has a search bar with 'spoke' typed in, a 'Cloud Fabric' section, and a 'Gateways' section with a 'Spoke Gateways' item. The main area has tabs for 'Gateways', 'Overview', 'Transit Gateways', and 'Spoke Gateways' (which is highlighted with a red box and numbered 3). Below the tabs is a search bar and a button labeled '+ Spoke Gateway'. The main table lists four existing spoke gateways with columns for Name, Region, VPC/VNet, and Subnet CIDR.

Name	Region	VPC/VNet	Subnet CIDR
aws-us-east-1-SpokeA	us-east-1	vpc-08f1e...	10.0.0.0/24
aws-us-west-2-spoke-1	us-west-2	vpc-0dff...	10.51.0.32/28
aws-us-west-2-spoke-2	us-west-2	vpc-04fd9...	10.52.0.32/28
aws-us-west-2-spoke-3	us-west-2	vpc-0270...	10.53.0.32/28

## Lab 2: Step 2.40

Deploy Aviatrix Spoke Gateway in VPC B Public Subnet

Create Spoke Gateway

Name **1**: aws-us-east-1-SpokeB

Cloud **2**: AWS Standard

Account **3**: aws-account

Region **4**: us-east-1 (N. Virginia)

VPC/VNet **5**: VPC B

Instance Size **6**: t3.medium

High Performance Encryption: Off

Attach To Transit Gateway: Optional

Name the gateway **aws-us-east-1-SpokeB** **1**

Select the cloud AWS standard **2**

Select the account **aws-account** **3**

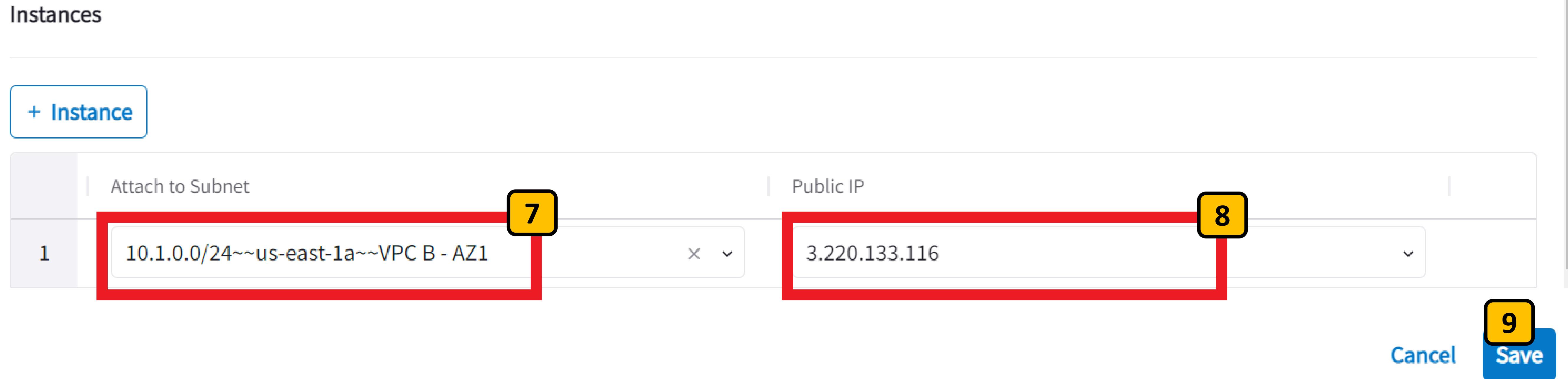
Select the region **us-east-1** **4**

Select **VPC B** **5**

Select the **t3.medium** instance size **6**

## Lab 2: Step 2.41

Deploy Aviatrix Spoke Gateway in VPC B Public Subnet



Select the 10.1.0.0/24 subnet in AZ1 **7**

Select the Public IP that the NAT Gateway in VPC B was using before it was deleted. **8**

Click **Save** **9**

## Lab 2: Step 2.42

Monitor the gateway deployment Task

The screenshot shows the Aviatr ix CoPilot interface. On the left is a dark sidebar with the following items:

- CoPilot (with a yellow box around the number 1)
- task (with a red box around the search bar)
- Monitor (with a yellow box around the number 2)
- Notifications / Tasks (with a red box around the result)
- Settings
- Resources
- Task

The main area has tabs at the top: Notifications, Alerts, Alerts Configuration, System Messages, Tasks (selected), and Recipients. Below is a sub-tab bar with Tasks (selected) and Active Gateway Operations. A search bar is present. The main table lists tasks:

Name	Entity	Status	Progress
Create spoke gateway: aws-us-east-1-SpokeB		In Progress	[Progress bar]
Create spoke gateway: aws-us-east-1-SpokeA		Completed	[Green progress bar]

From the CoPilot search bar type **task** 1

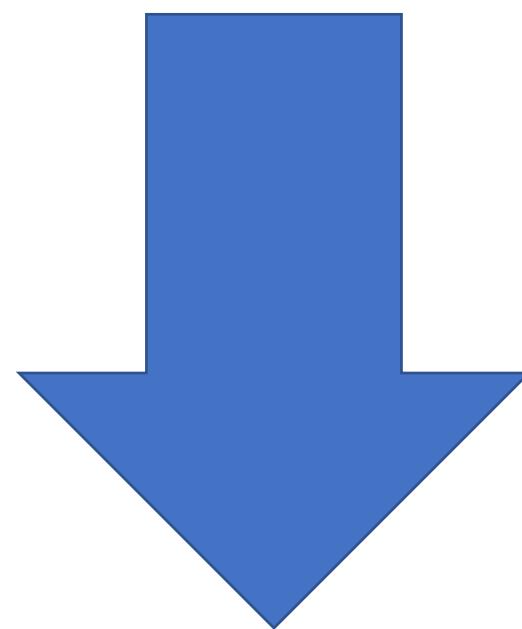
Click the search result **Notifications / Tasks** 2

Observe the spoke gateway creation Task and wait for it to complete 3

## Lab 2: Step 2.43

Enable Egress on the AviatrIx Spoke Gateway

Now let's configure your new AviatrIx Spoke Gateway to do Local Egress for VPC B, so we can use it for egress NAT from our private subnets.



From the CoPilot search bar type **egress** **1**

Click the search result **Egress VPC/VNets** **2**

Click the **+ Local Egress on VPC/VNets** button **3**

The screenshot shows the AviatrIx CoPilot interface. At the top, there is a search bar with the word "egress" (marked with a yellow box and number 1). Below the search bar, the sidebar has a "Security" icon, followed by "Egress" (marked with a yellow box and number 2), and "Egress VPC/VNets". The main content area has a tab bar with "Egress" (marked with a yellow box and number 3), "Overview", and "Monitor". The "Egress" tab is active. In the main content area, there is a button labeled "+ Local Egress on VPC/VNets". Below this, there is a table with two columns: "Name" and "Point of Egress". The table lists five entries: "aws-us-east-1-SpokeA" (Local Egress), "aws-us-west-2-spoke-1" (Local Egress), "aws-us-west-2-spoke-2" (Local Egress), "aws-us-west-2-spoke-3" (Local Egress), and "aws-us-east-1-SpokeB" (Native Cloud Egress). At the bottom of the table, it says "Total 5 VPC/VNets".

Name	Point of Egress
aws-us-east-1-SpokeA	Local Egress
aws-us-west-2-spoke-1	Local Egress
aws-us-west-2-spoke-2	Local Egress
aws-us-west-2-spoke-3	Local Egress
aws-us-east-1-SpokeB	Native Cloud Egress

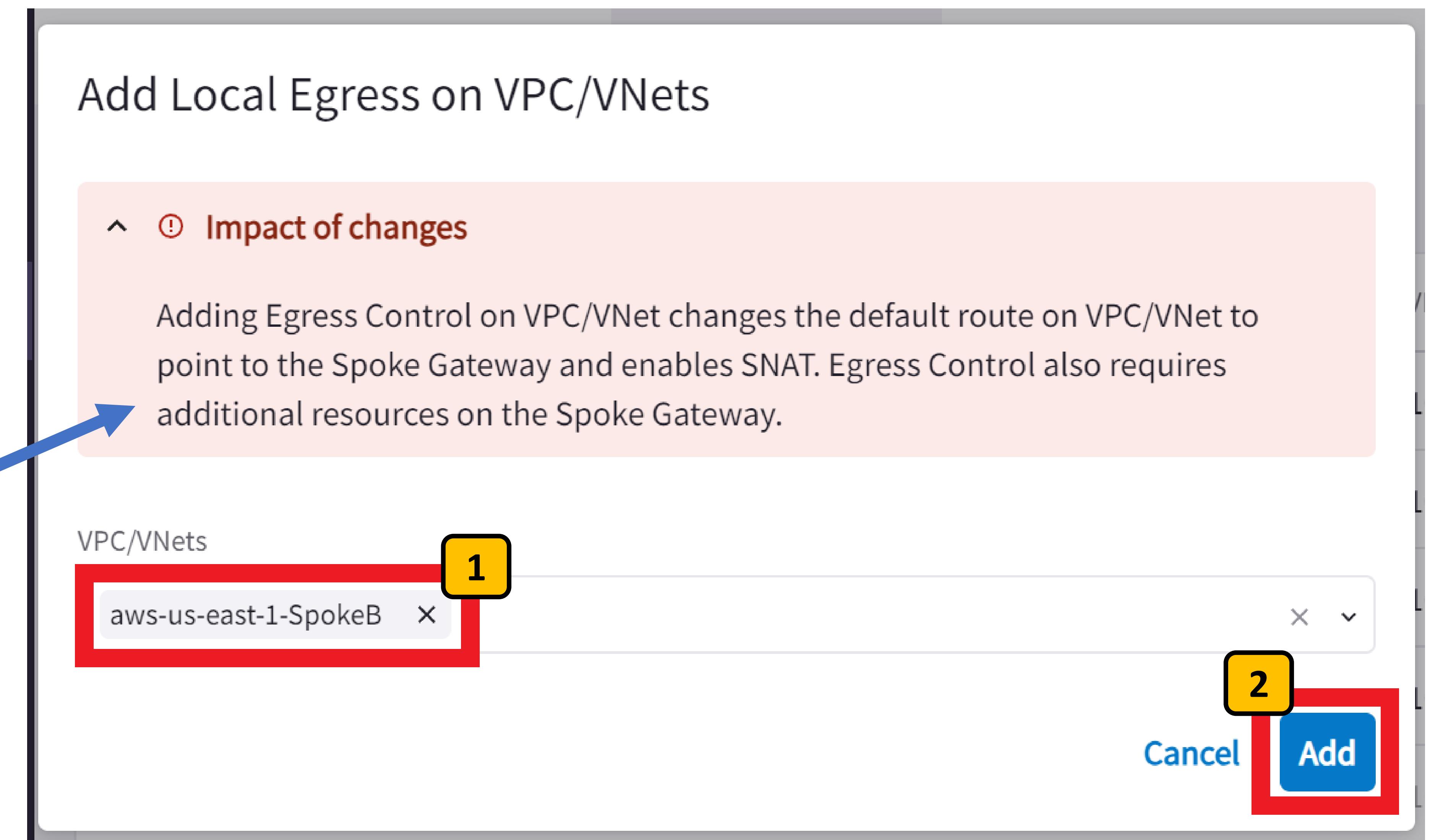
## Lab 2: Step 2.44

Enable Egress on the Aviatrix Spoke Gateway

In the Add Local Egress on VPC/VNets pop-up, select the new **aws-us-east-1-SpokeB** gateway from the VPC/VNets drop down **1**

Click **Add** **2**

After you click Add, Copilot will change the VPC default route associated to all private subnets in VPC B to point to your new Aviatrix Gateway



## Lab 2: Checkpoint 5

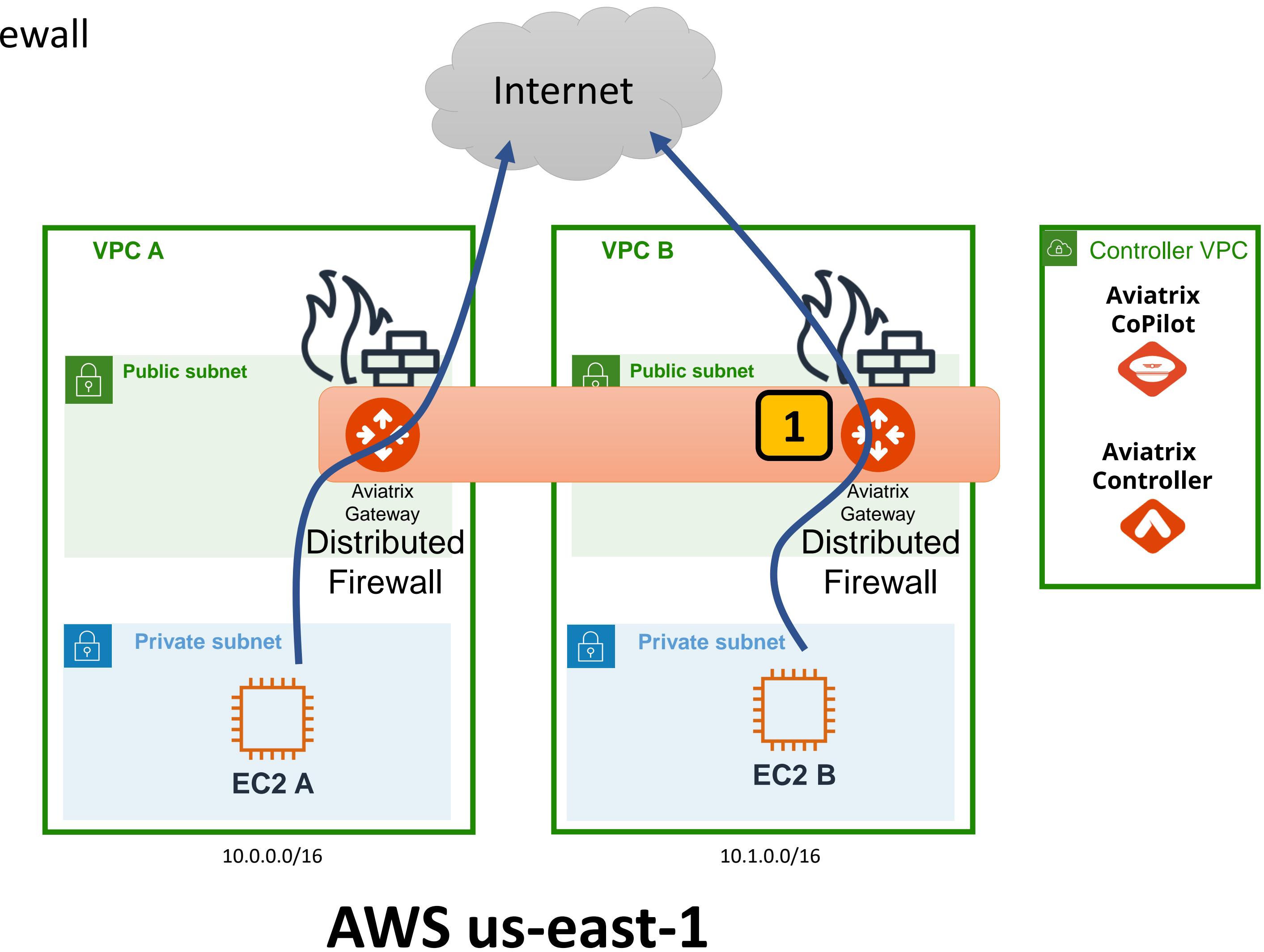
Aviatrix Distributed Firewall

At this point you've just extended your Aviatrix Distributed Firewall to VPC B. **1**

The policies that you've already centrally configured in Aviatrix CoPilot have been extended and applied to VPC B.

Egress traffic for the instance EC2 B is now flowing through the Aviatrix Gateway in VPC B where firewall rules are enforced.

**Next: Let's test the internet again from the instance EC2 B and see if it's already secured.**



**AWS us-east-1**

**Note:** Aviatrix does not charge data processing charges for Firewall and NAT. The Aviatrix cost for Distributed Firewall and NAT is **\$0.23 /hr per gateway** (plus the EC2 gateway instance charges)

## Lab 2: Step 2.45

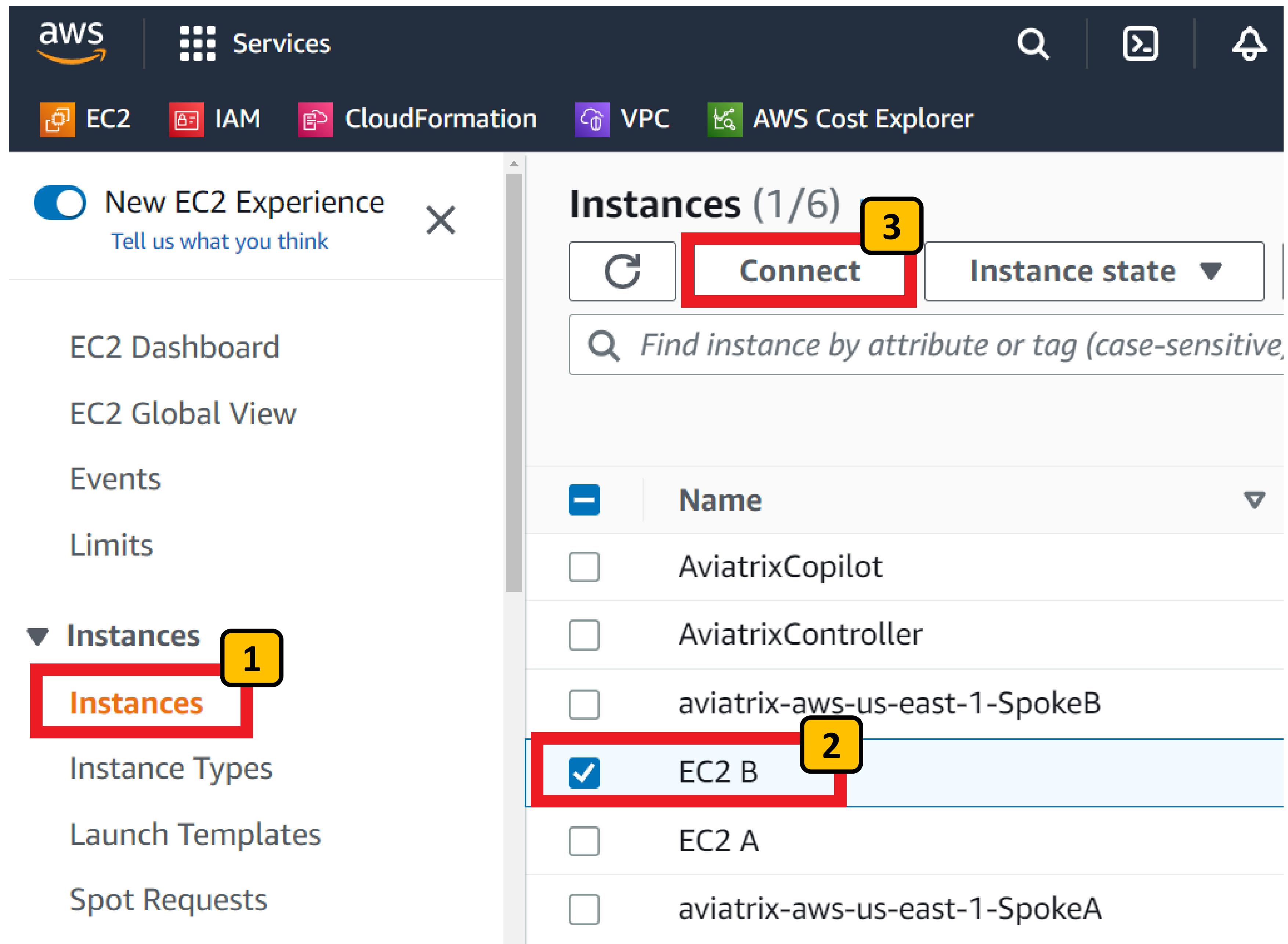
Connect to EC2 B instance console

Go the EC2 section of the AWS Console.

Select Instances 1

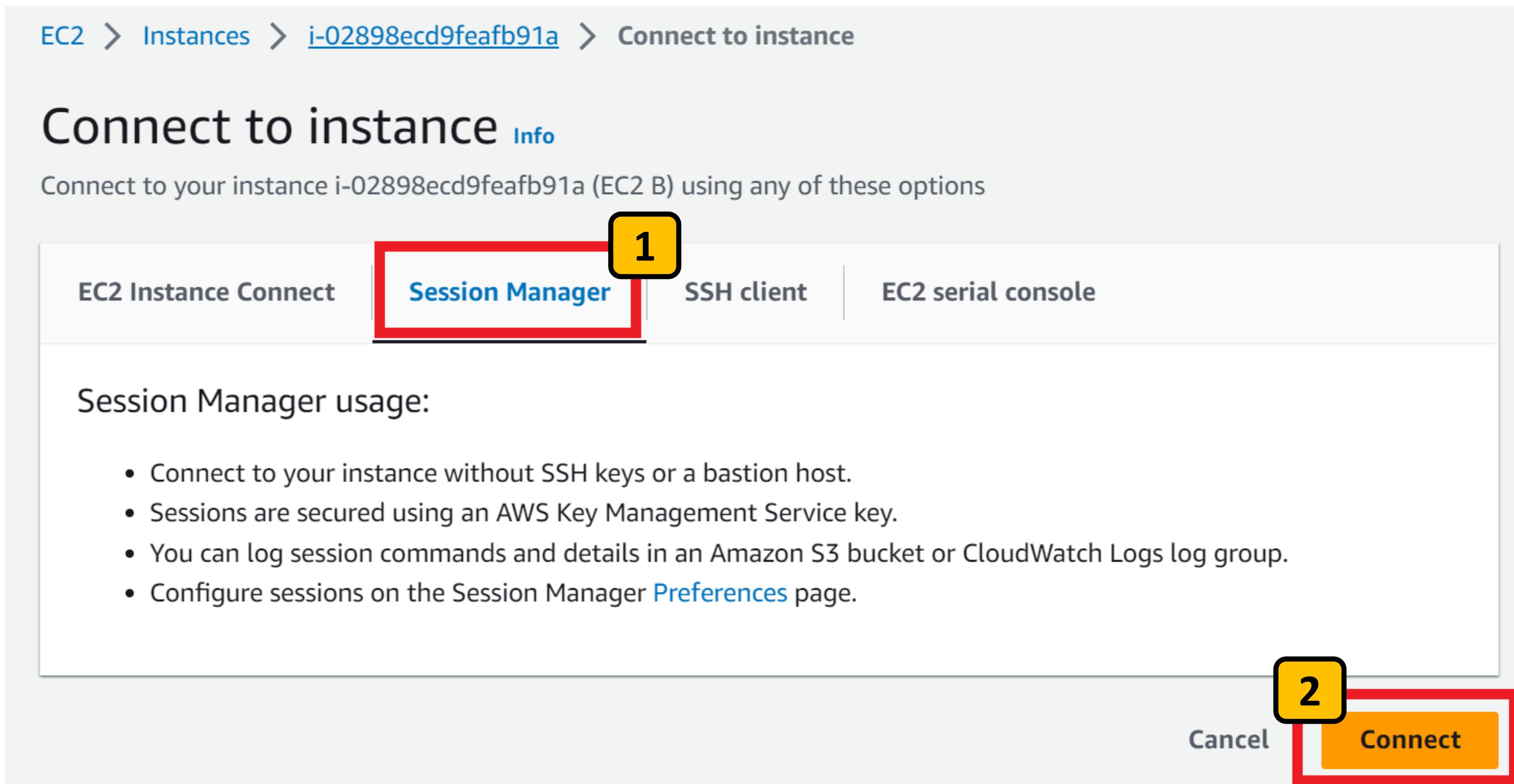
Find the EC2 B instance and select it. 2

Click the Connect button 3



## Lab 2: Step 2.46

Connect to EC2 B instance console



Select the Session Manager tab. **1**

Click the Connect button **2**

*Note:* If you get an error here, try rebooting EC2 B to accelerate the process of Session Manager reconnecting through the Aviatrix Gateway

## Lab 2: Step 2.47

Connect to EC2 B instance console

Your browser should open a new tab giving you a CLI session.

Type the command:

**sudo su –l ec2-user** 1

*(dash lower case L)*

You are now logged on as the ec2-user and should see your private IP address in the hostname.

Next, let's test internet egress.

```
Session ID: brad-  
0104b77d04-7dfb04  
sh-4.2$  
sh-4.2$  
sh-4.2$ sudo su -l ec2-user1  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$
```

## Lab 2: Step 2.48

Test connections to potentially harmful domains

Test that your EC2 B instance has secured internet access that will prevent it from connecting to potentially harmful domains.

Run the commands:

**curl https://ransomware.org** 1

**curl https://malware.net** 2

**curl https://botnet.com** 3

For each command you should see the CLI return an SSL Error, because the Aviatrix Distributed Firewall is blocking the connection 4

```
Session ID: brad-  
0f07141bd207757a6  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$ curl https://ransomware.org
```

1

```
Session ID: brad-  
0f07141bd207757a6  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$ curl https://malware.net
```

2

```
Session ID: brad-  
0f07141bd207757a6  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$ curl https://botnet.com
```

3

```
Session ID: brad-  
0f07141bd207757a6  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$  
[ec2-user@ip-10-1-2-10 ~]$ curl https://botnet.com  
curl: (35) error:14077410:SSL routines:SSL23_GET_SERVER_HELLO:sslv3 alert
```



4

## Lab 2: Step 2.49

Run a software update

Let's make sure we can still run a software update from the AWS repositories

Run the command:  
**sudo yum update -y** 1

This will connect to ***amazonaws.com*** domains to download software updates.

Great! Software updates are working because of the Allow-AWS rule in the Aviatrix Distributed Firewall.

Session ID: brad-

Instance ID: i-02898ecd9feaf91a

```
[ec2-user@ip-10-1-2-10 ~] $  
[ec2-user@ip-10-1-2-10 ~] $  
[ec2-user@ip-10-1-2-10 ~] $  
[ec2-user@ip-10-1-2-10 ~] $ sudo yum update -y1
```

```
xz.x86_64 0:5.2.2-1.amzn2.0.3  
xz-libs.x86_64 0:5.2.2-1.amzn2.0.3  
yum.noarch 0:3.4.3-158.amzn2.0.6  
zlib.x86_64 0:1.2.7-19.amzn2.0.2
```

Replaced:

```
grub2.x86_64 1:2.06-2.amzn2.0.3  
python-colorama.noarch 0:0.3.2-3.amzn2.0.2
```

Complete!

```
[ec2-user@ip-10-1-2-10 ~] $
```

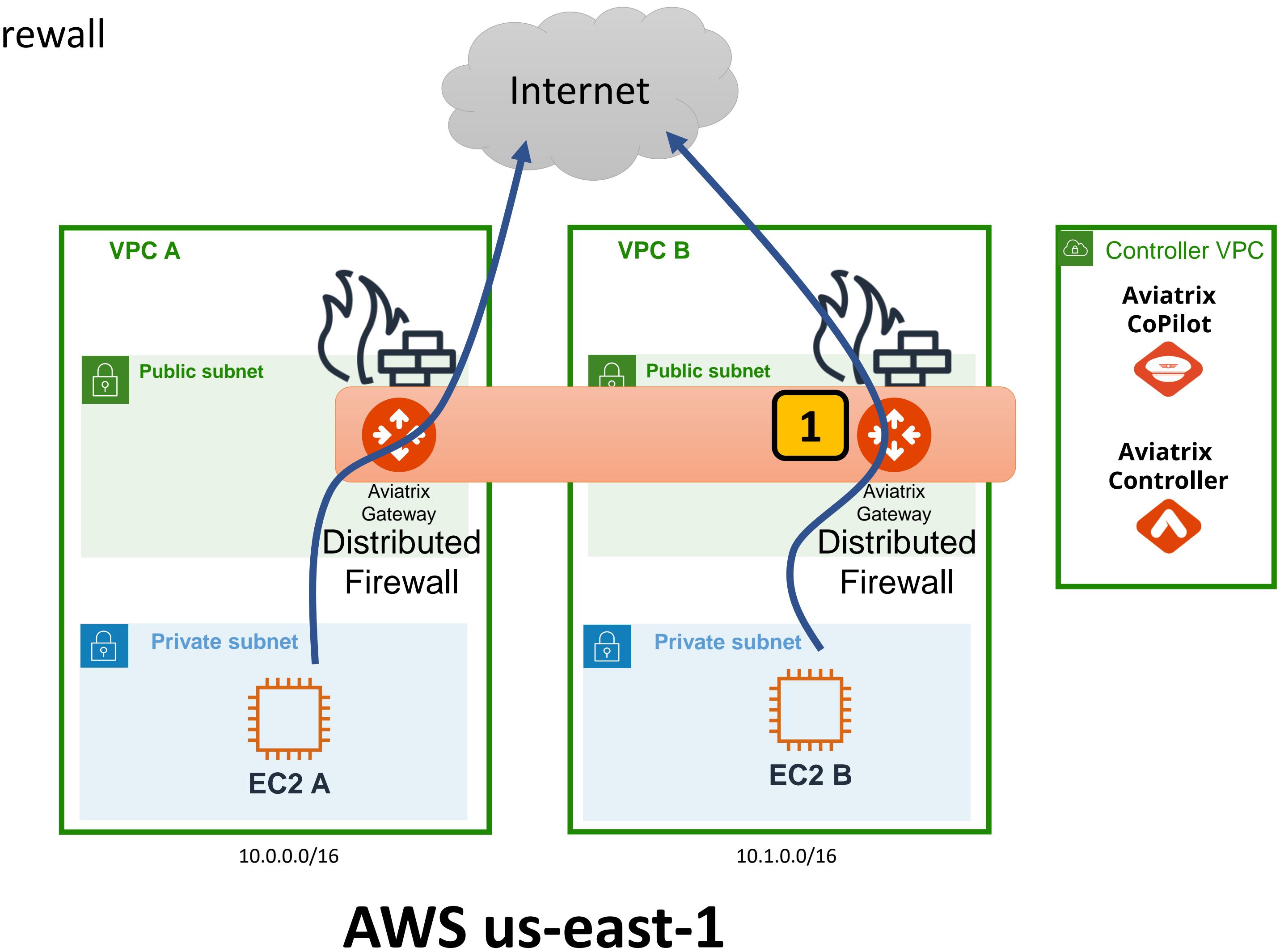
## Lab 2: Checkpoint 6

Aviatrix Distributed Firewall

At this point you've just validated that your Aviatrix Distributed Firewall extended **1** seamlessly to VPC B.

You also reused the EIP from the AWS NAT Gateway that was there before.

In addition to security, The Aviatrix Distributed Firewall also provides deep traffic visibility, monitoring, alerting, and scaling policies.



Next, let's look at traffic details, set monitoring alerts, and create a scaling policy.

## Lab 2: Step 2.50

See the Policy actions

The screenshot shows the CoPilot interface for AWS AviatrIX. On the left sidebar, under the Security section, the 'Distributed Cloud Firewall' tab is selected and highlighted with a red box. The main content area is titled 'Monitor' and also has a red box around it. A yellow box with the number '1' highlights the 'Monitor' tab. The table below displays log entries with a red box around the 'Action' column, which contains 'Permit' repeated six times. A yellow box with the number '2' highlights the 'Action' column. The table includes columns for Timestamp, Rule, L4/L7, Source SmartGroup, Action, and Enforced.

Timestamp	Rule	L4/L7	Source SmartGroup	Action	Enforced
Aug 14, 2023 3:42:11 PM	Allow-AWS	L4	PROD	Permit	Yes
Aug 14, 2023 3:42:11 PM	Allow-AWS	L4	PROD	Permit	Yes
Aug 14, 2023 3:42:11 PM	Allow-AWS	L7	PROD	Permit	Yes
Aug 14, 2023 3:42:11 PM	Allow-AWS	L7	PROD	Permit	Yes
Aug 14, 2023 3:42:10 PM	Allow-AWS	L4	PROD	Permit	Yes

Total 83471 logs

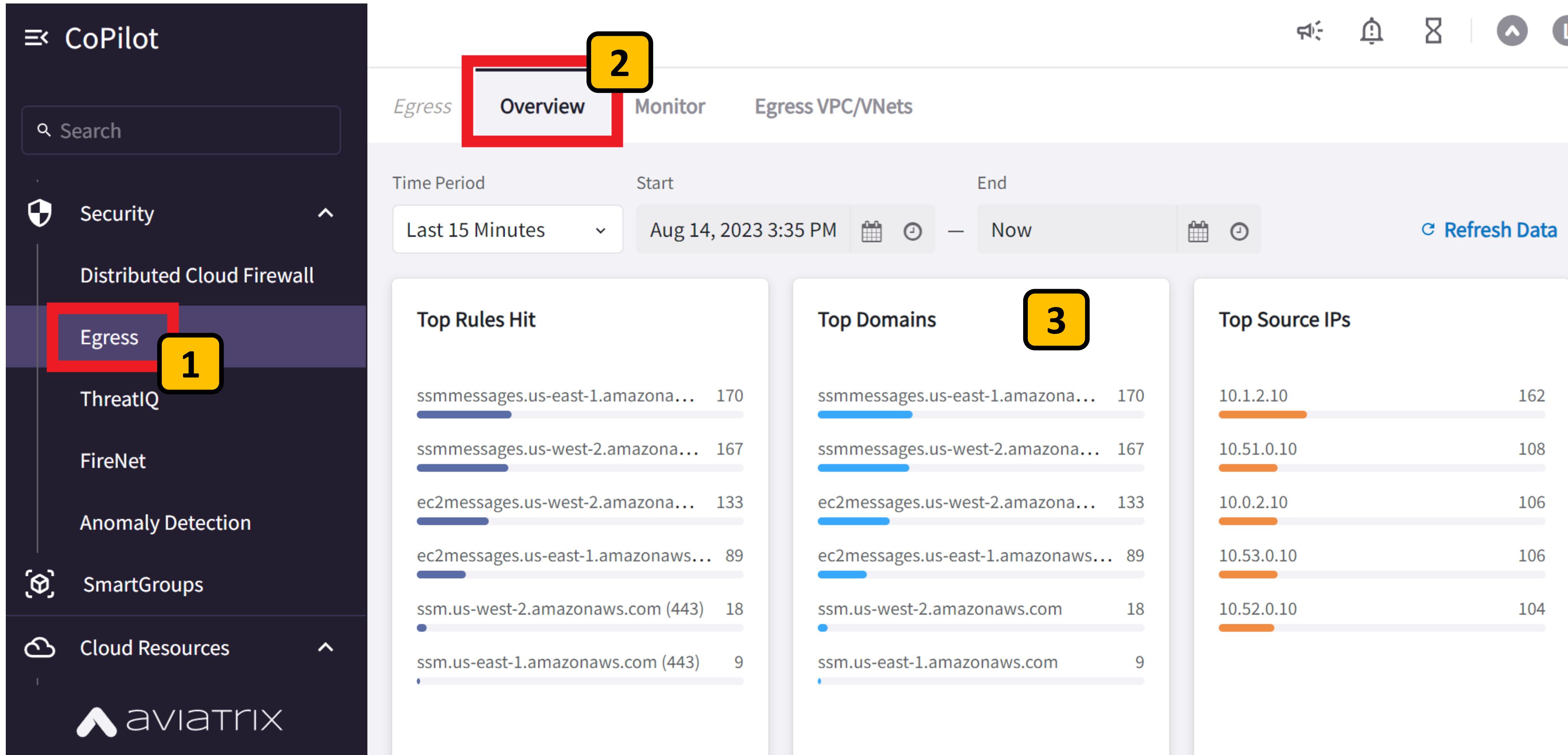
Monitor logs the last 1,000,000 packets that hit the Policy Rules in Distributed Cloud Firewall.

In the Distributed Cloud Firewall section of CoPilot select **Monitor** tab **1**

Observe the session details and policy decisions as traffic flows through the Distributed Cloud Firewall. **2**

## Lab 2: Step 2.50

Observe the Domain traffic activity



From the CoPilot navigation select Egress under Security **1**

Select the Overview tab **2**

Observe domain traffic details including Top Domains and Top Source IPs. **3**

Notice the **amazonaws.com** domains supporting Session Manager and software updates

# Lab 2: Step 2.50

Observe the Domain traffic activity

The screenshot shows the CoPilot interface with the following steps highlighted:

- 1**: Egress under Security in the navigation bar.
- 2**: Monitor tab in the top navigation bar.
- 3**: VPC/VNets dropdown menu showing 'aws-us-east-1-SpokeA' and 'aws-us-east-1-SpokeB' selected.

The main table displays domain traffic activity:

Timestamp	Source IP	VPC/VNet	Domain	Port	Rule Match	Action
Aug 14, 2023 3:55 PM	10.1.2.10	aws-us-east-1-S...	ec2messages.us...	443	Matched	Allowed
Aug 14, 2023 3:55 PM	10.0.2.10	aws-us-east-1-S...	ssmmessages.us...	443	Matched	Allowed
Aug 14, 2023 3:55 PM	10.0.2.10	aws-us-east-1-S...	ssmmessages.us...	443	Matched	Allowed
Aug 14, 2023 3:55 PM	10.0.2.10	aws-us-east-1-S...	ec2messages.us...	443	Matched	Allowed
Aug 14, 2023 3:55 PM	10.1.2.10	aws-us-east-1-S...	ssmmessages.us...	443	Matched	Allowed
Aug 14, 2023 3:55 PM	10.1.2.10	aws-us-east-1-S...	ssmmessages.us...	443	Matched	Allowed

From the CoPilot navigation select Egress under Security **1**

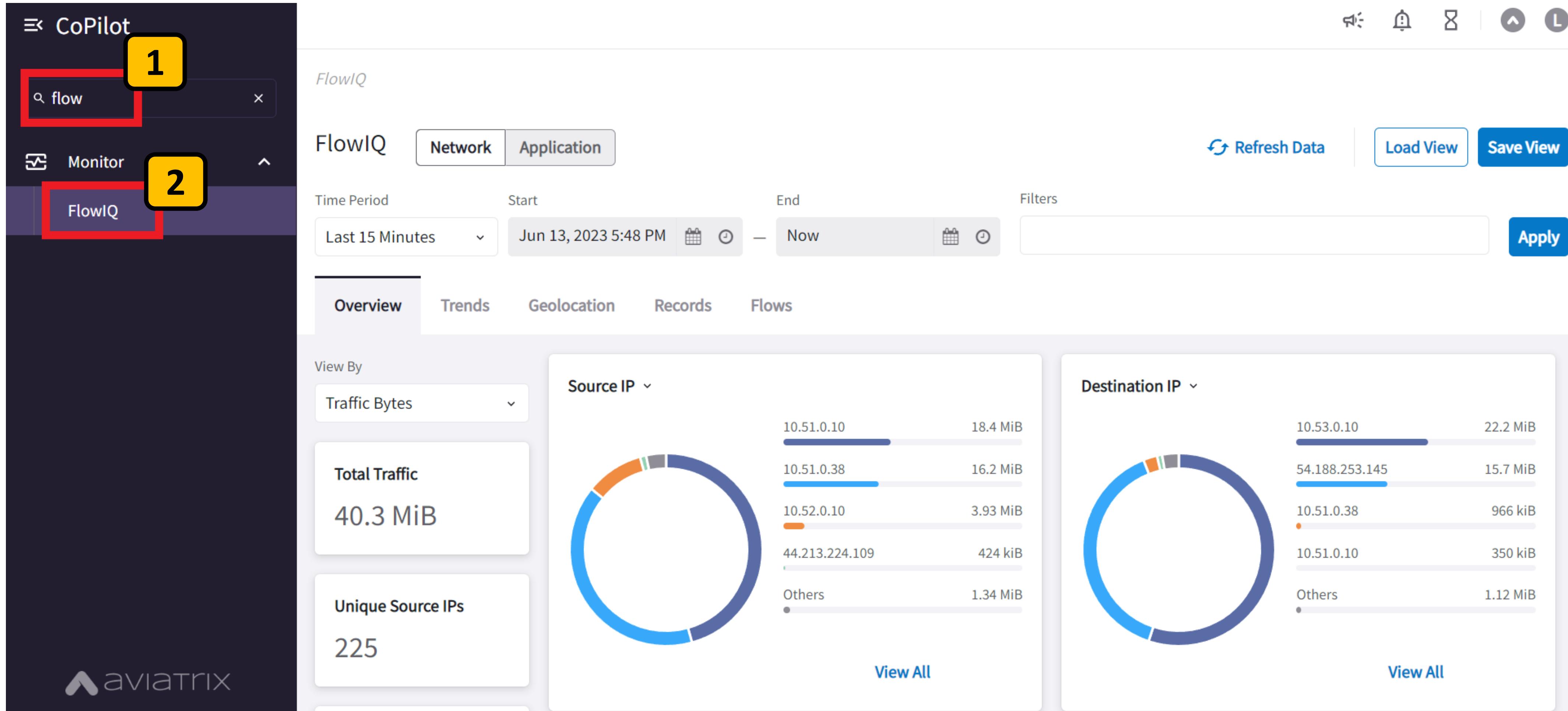
Select the Monitor tab **2**

Select SpokeA and SpokeB gateways in the VPC/VNets dropdown to monitor domain activity flowing through them. **3**

**Challenge:** Try to generate domain traffic on EC2 A using curl and come back to this screen to see if you can see the activity.

## Lab 2: Step 2.51

See traffic flow details in FlowIQ



In the CoPilot search bar type **flow** 1

Select the **FlowIQ** search result 2

**FlowIQ** provides an intuitive dashboard to visualize and query traffic details as it flows through your Aviatrix Gateways.

## Lab 2: Step 2.52

The screenshot shows the Aviatr ix CoPilot interface with the FlowIQ tab selected. The top navigation bar includes a search bar with 'flow', a bell icon, and a refresh button. Below the search bar, there are tabs for 'Monitor' and 'FlowIQ'. The 'FlowIQ' tab is active, showing a time period from 'Last 15 Minutes' to 'Now'. A red box labeled '1' highlights the 'Filters' input field containing 'Source CSP Tag = Environment: Development'. To the right of the filters is an 'Apply' button in a red box labeled '2'. The main area displays traffic summary cards for 'Source IP' and 'Destination IP'. The 'Source IP' card shows a total traffic of 3.79 kiB and two unique source IPs. The 'Destination IP' card shows a list of destination IPs with their respective traffic volumes. A red box labeled '3' highlights the top source IP, 10.1.2.10, which has 3.71 kiB of traffic.

Select the Filters box and filter on traffic sourced from only EC2 instances with the tag **Environment = Development** **1**

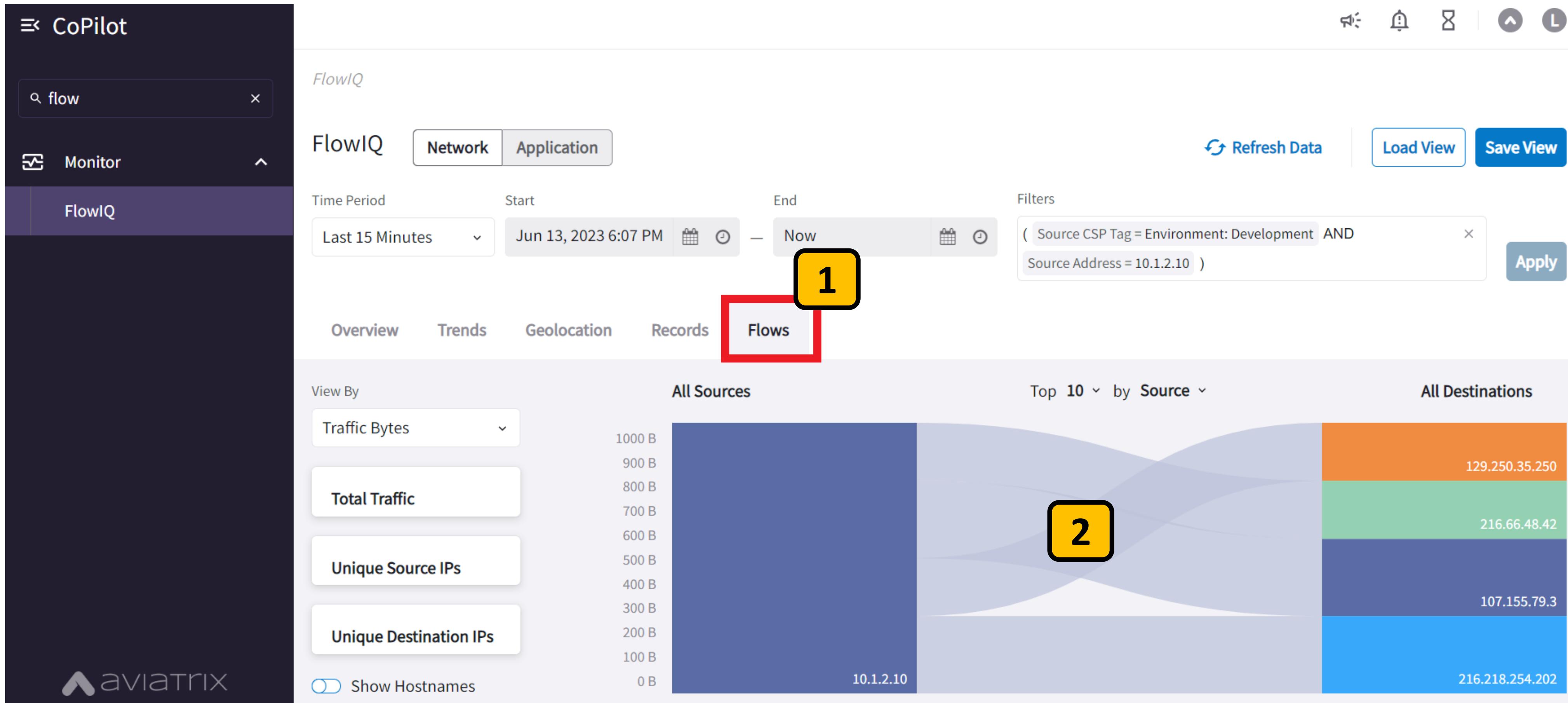
Click **Apply** to apply the filter **2**

You immediately found the top talker in the **Development** environment without even thinking about IP addresses!

Click the top Source IP address narrow in on this talker and click Apply again. **3**

## Lab 2: Step 2.53

Query traffic flow details in FlowIQ



Select the Flows tab to see a flow summary from this top talker in Development **1**

Hover your mouse over each flow to get more details. **2**

## Lab 2: Step 2.54

Query traffic flow details in FlowIQ

The screenshot shows the CoPilot interface with the FlowIQ tab selected. The 'Records' tab is highlighted with a red box and a yellow number 1. The table below displays three network flow records:

Timestamp	Host	Destination Address	Source Address	Bytes	Direction	Packets
Jun 13, 2023 6:07:22 PM	3.220.133.116	129.250.35.250	10.1.2.10	76	forward	1
Jun 13, 2023 6:07:22 PM	3.220.133.116	216.66.48.42	10.1.2.10	76	forward	1
Jun 13, 2023 6:07:22 PM	3.220.133.116	107.155.79.3	10.1.2.10	76	forward	1

Select the **Records** tab to see details of each traffic flow record from this top talker in Development **1**

Aviatrix Gateways export full Netflow information to CoPilot and you're seeing that raw data here.

## Lab 2: Step 2.55

The screenshot shows the Aviatr ix CoPilot interface. On the left, there's a sidebar with a search bar (1) containing 'tools'. Below it are sections for Troubleshoot (2), Diagnostic Tools (3), Ping (4), and Packet Capture (5). The main area is titled 'Take a packet capture' and contains tabs for Diagnostic Tools, Gateway Diagnostics, Connectivity Diagnostics, and BGP Diagnostics. It shows a gateway instance 'aws-us-east-1-Spo...' and an interface 'eth0'. A table lists network traffic with columns for Time, Source IP, Source Port, Destination, Dest Port, and Protocol. At the top right, there are 'Reset' and 'Run' buttons, with 'Run' highlighted (6). A red box highlights the 'Download PCAP File' button.

Time	Source IP	Source Port	Destination	Dest Port	Protocol
23:30:49.004481	44.213.224.109	443	10.0.0.226	42026	IP
23:30:49.004533	10.0.0.226	42026	44.213.224.109	443	IP
23:30:49.004698	10.0.0.226	42026	44.213.224.109	443	IP
23:30:49.004711	10.0.0.226	42026	44.213.224.109	443	IP
23:30:49.004896	44.213.224.109	443	10.0.0.226	42026	IP

In the CoPilot search bar type **tools** 1

Select **Packet Capture** 4

Select the **Diagnostic Tools** search result 2

Select the **eth0** interface 5

Select a gateway to capture from 3

Select **Run** and **Download PCAP file** when the capture completes. 6

## Lab 2: Step 2.56

Configure monitoring alerts

The screenshot shows the CoPilot interface with the following details:

- Left Sidebar:** Shows a search bar with "alerts" typed in, a "Monitor" section, and a "Notifications" section with "Alerts Configuration" selected.
- Top Navigation:** Includes tabs for "Notifications", "Alerts", "Alerts Configuration" (which is active and highlighted with a red border), "System Messages", "Tasks", and "Recipients".
- Main Content:** A table titled "Alerts Configuration" with the following data:

Name	Condition	Monitored Entities	Recipients
Global Control Plane Health	CPU Used (%) more than 90% Memory Used (%) more tha... Disk Free (%) less than 5%	Controller, + 1 more	<span>trash</span> <span>edit</span> <span>bell</span>
Global Network Health	Gateway Status changed Limit Exceeded Rate (PPS) ...	All Gateways: All Interfaces	<span>trash</span> <span>edit</span> <span>bell</span>
- Top Right:** Includes icons for volume, notifications, and other system controls.

In the CoPilot search bar type **alerts** 1

Select the **Alerts Configuration** search result 2

Click the **+ Alert Configuration** box 3

## Lab 2: Step 2.57

Configure monitoring alerts

Name the Alert **High CPU** 1

Select to monitor **Gateways and All Gateways Selected** 2

Match on the condition of **CPU Idle% less than 10%** 3

Set the Evaluation Period to **5 min** 4

Click **Save** 5

Create Alert Configuration

**Name**  
High CPU

**Monitor**

Controller  
 CoPilot  
 Gateways  
All Gateways Selected

**Condition**

Matches all conditions (AND) ▾

CPU Idle (%) less than 10 %

Evaluation Period 5 min

**Send Alerts To**

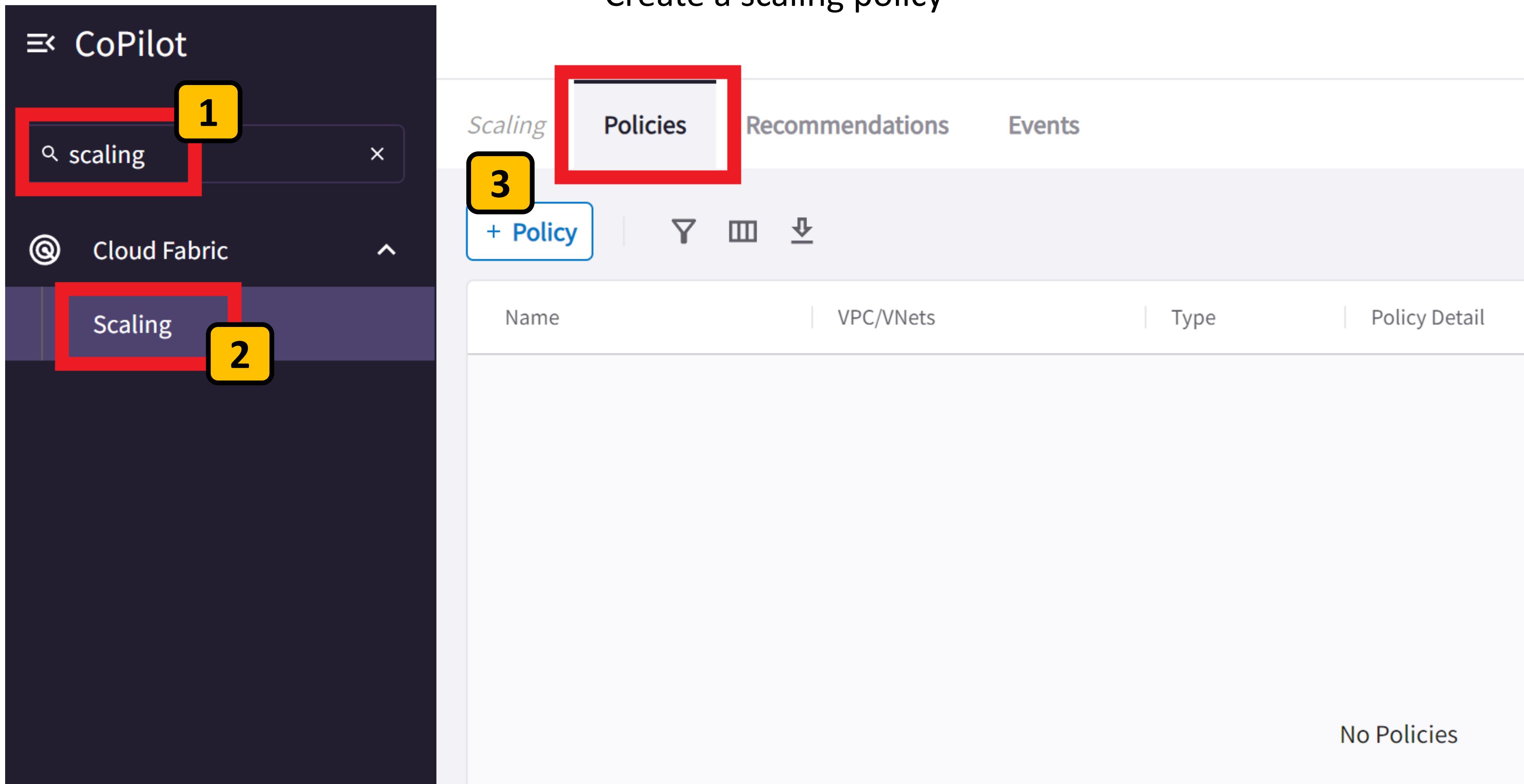
Recipients

Send separate alerts for each gateway  On

Cancel Save

## Lab 2: Step 2.58

Create a scaling policy



In the CoPilot search bar type **scaling** **1**

Select the **Scaling** search result **2**

Click the **+Policy** box to create a new scaling policy **3**

## Lab 2: Step 2.59

Create a scaling policy

Name scaling policy **Manual-Scaling** 1

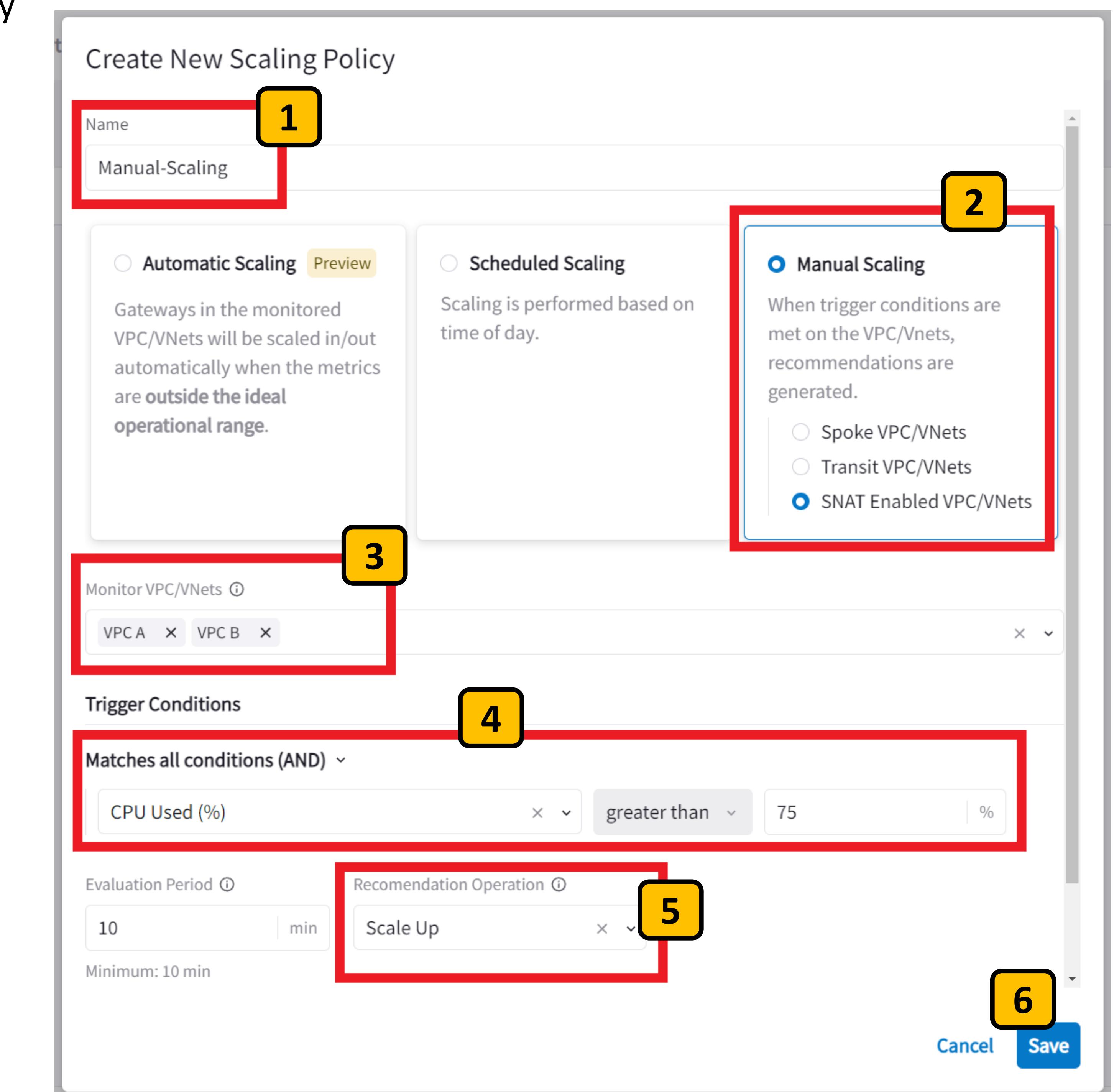
Select the **Manual Scaling** type and 2  
select **SNAT Enabled VPC**

Monitor **VPC A** and **VPC B** 3

Set the Trigger to **CPU Used%** greater  
than 75% 4

Set the recommendation operation to  
**Scale Up** 5

Click Save 6

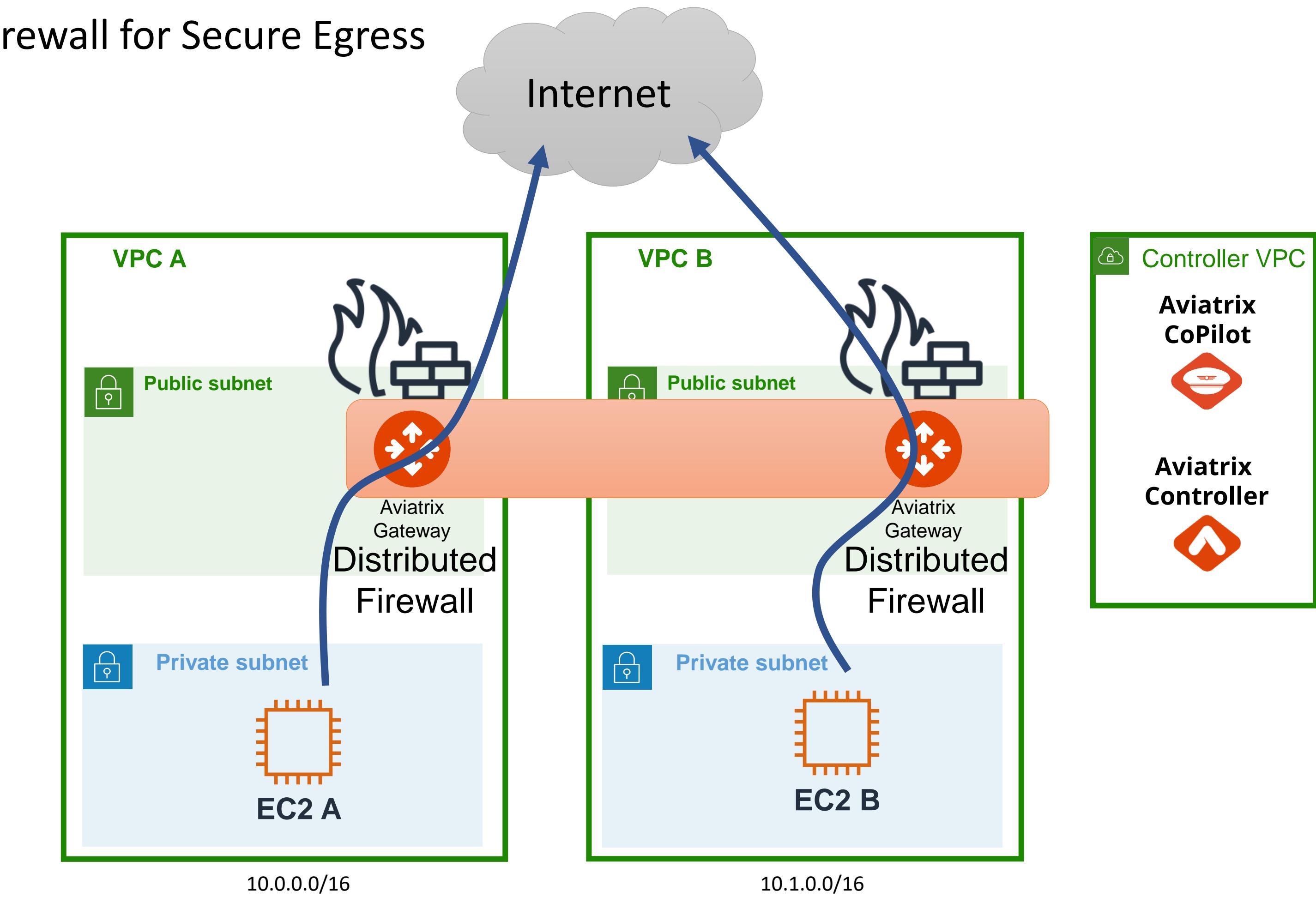


## Lab 2: Complete

Aviatrix Distributed Firewall for Secure Egress

**Congratulations!** You've just deployed the Aviatrix Distributed Cloud Firewall for Secure Egress.

You seamlessly switched your traffic from AWS NAT Gateways, created firewall rules that filtered on domain names, and easily extended the firewall to a second VPC.



You also observed traffic details, set monitoring alerts and created a scaling policy.

**AWS us-east-1**