AWS Immersion Day LAB 2

DISTRIBUTED FIREWALL FOR SECURE EGRESS

Brad Hedlund
Principal Solutions Architect, Aviatrix Systems

aviatrix

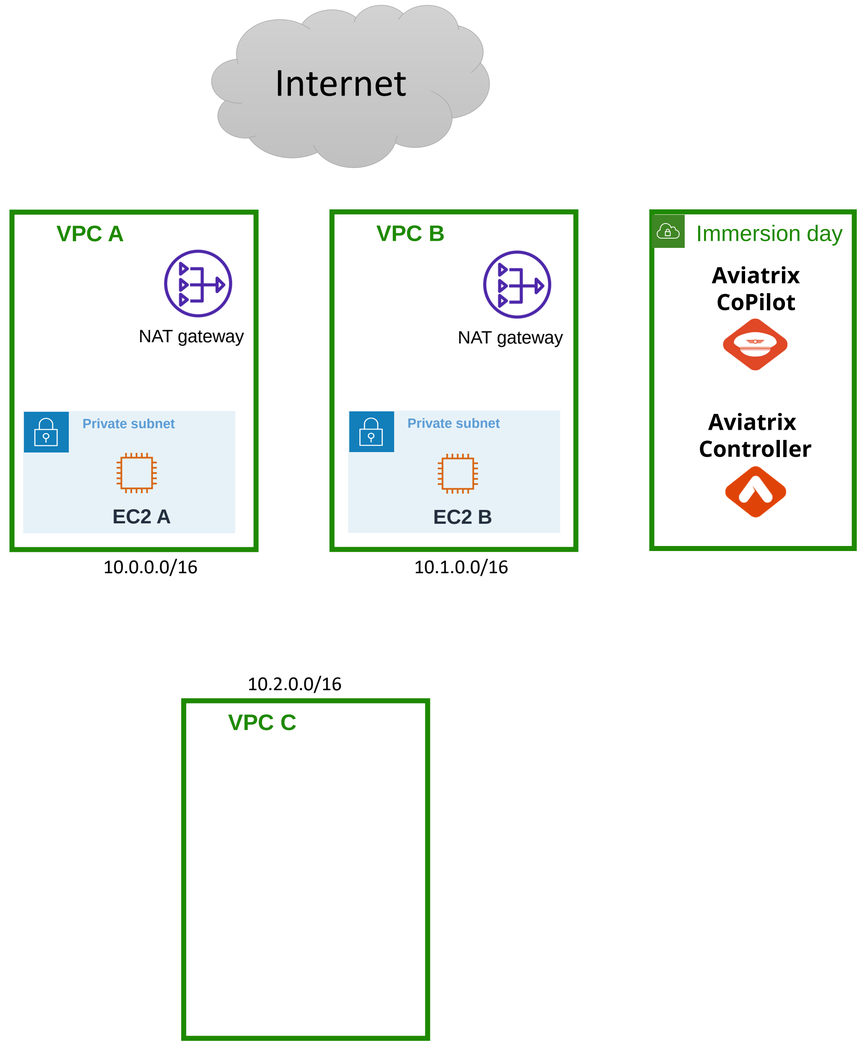
Lab 1 Recap

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 4th 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

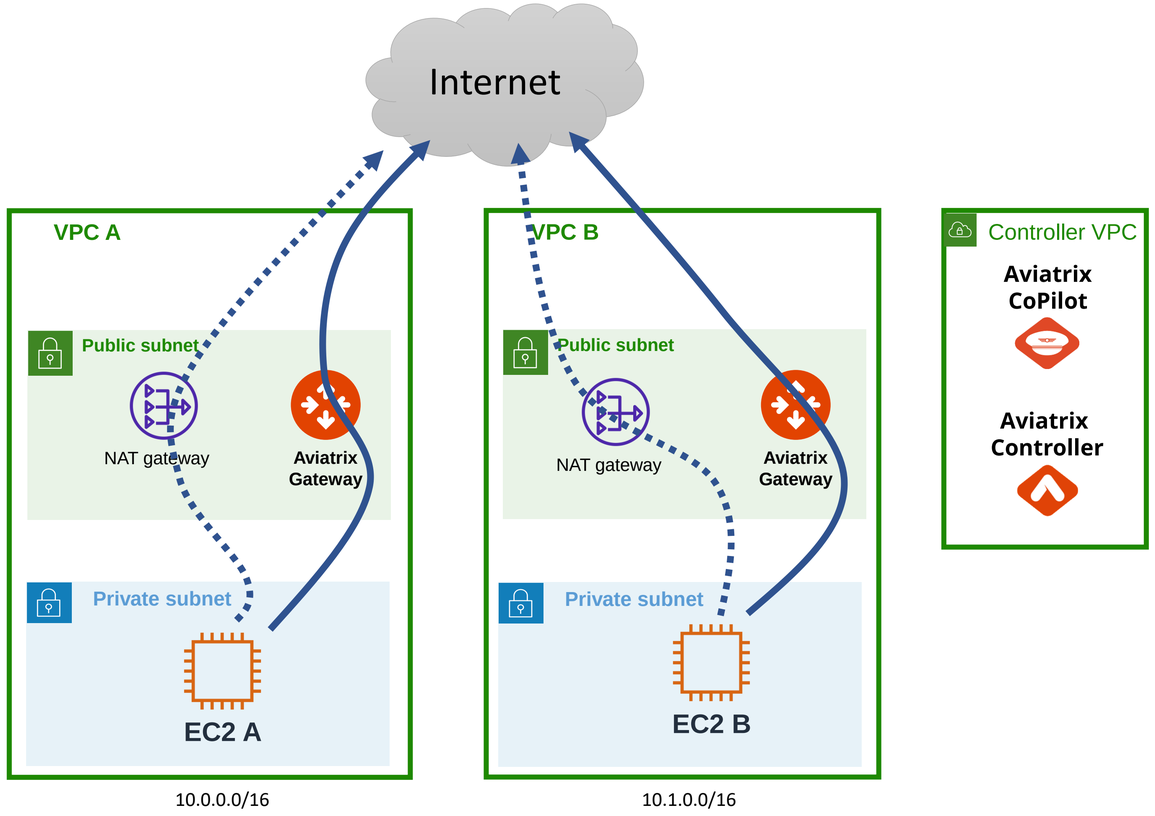


Lab 2 Intro

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

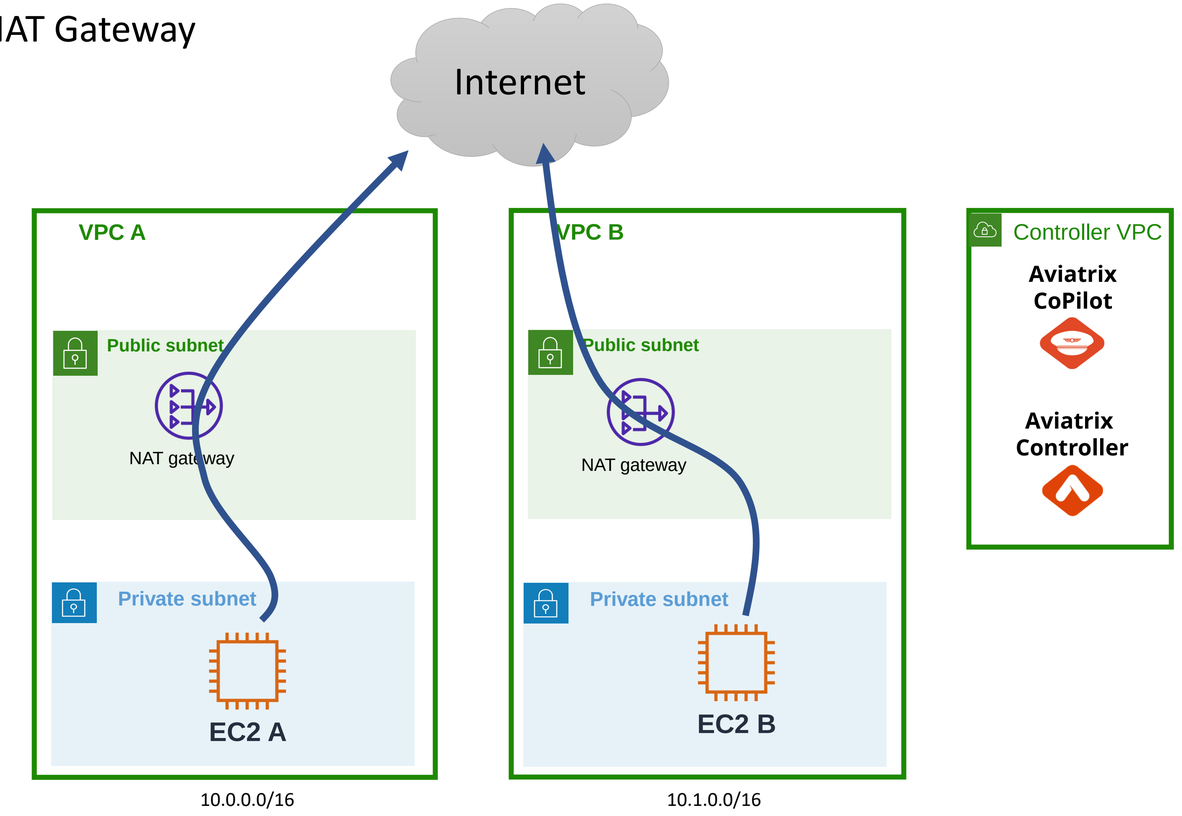
Lab 2: Step 2

Test egress with AWS NAT Gateway

Before we do anything with Aviatrix, 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

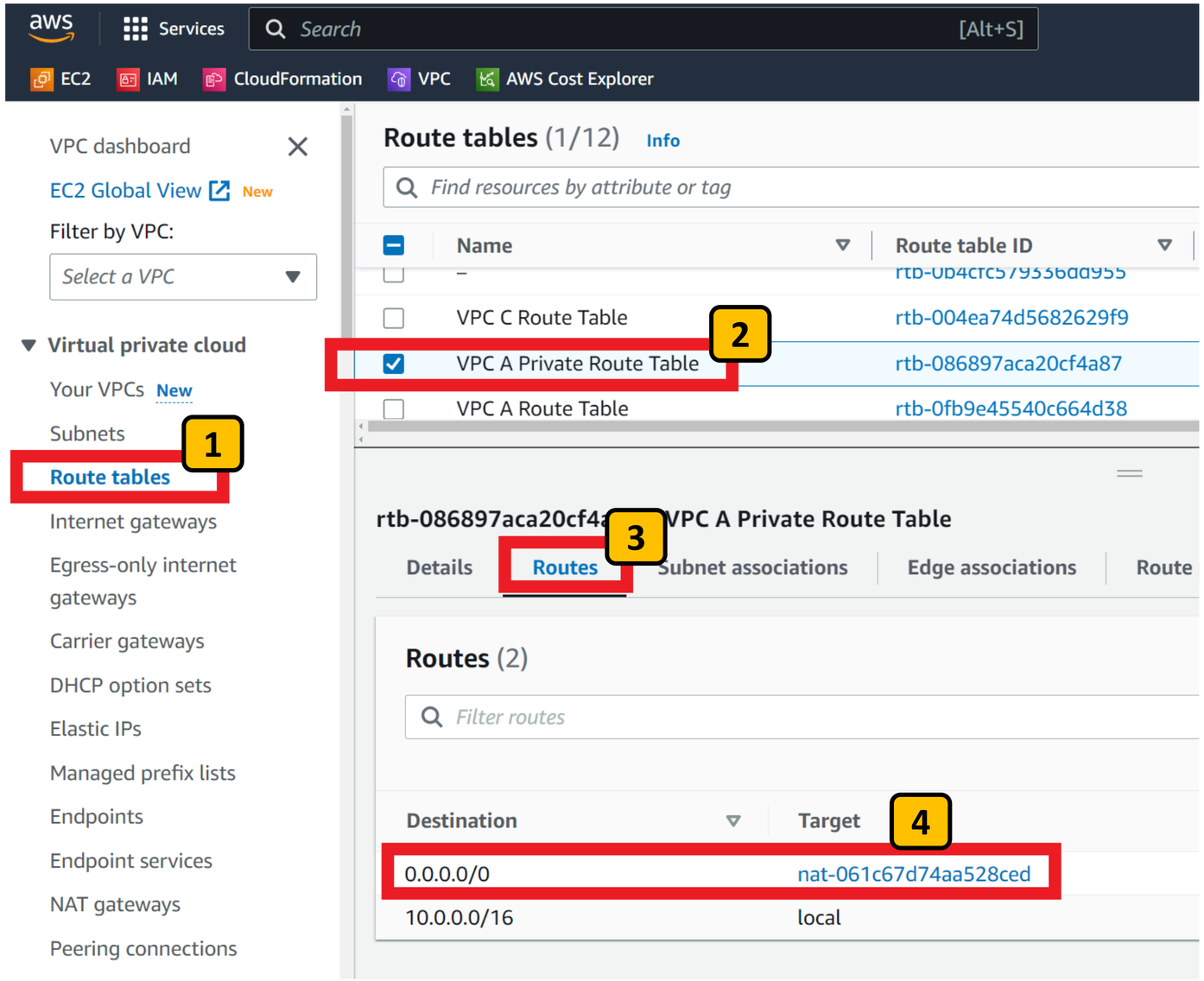
Make sure your AWS Console is in the us-east-1 **N. Virginia** region.

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

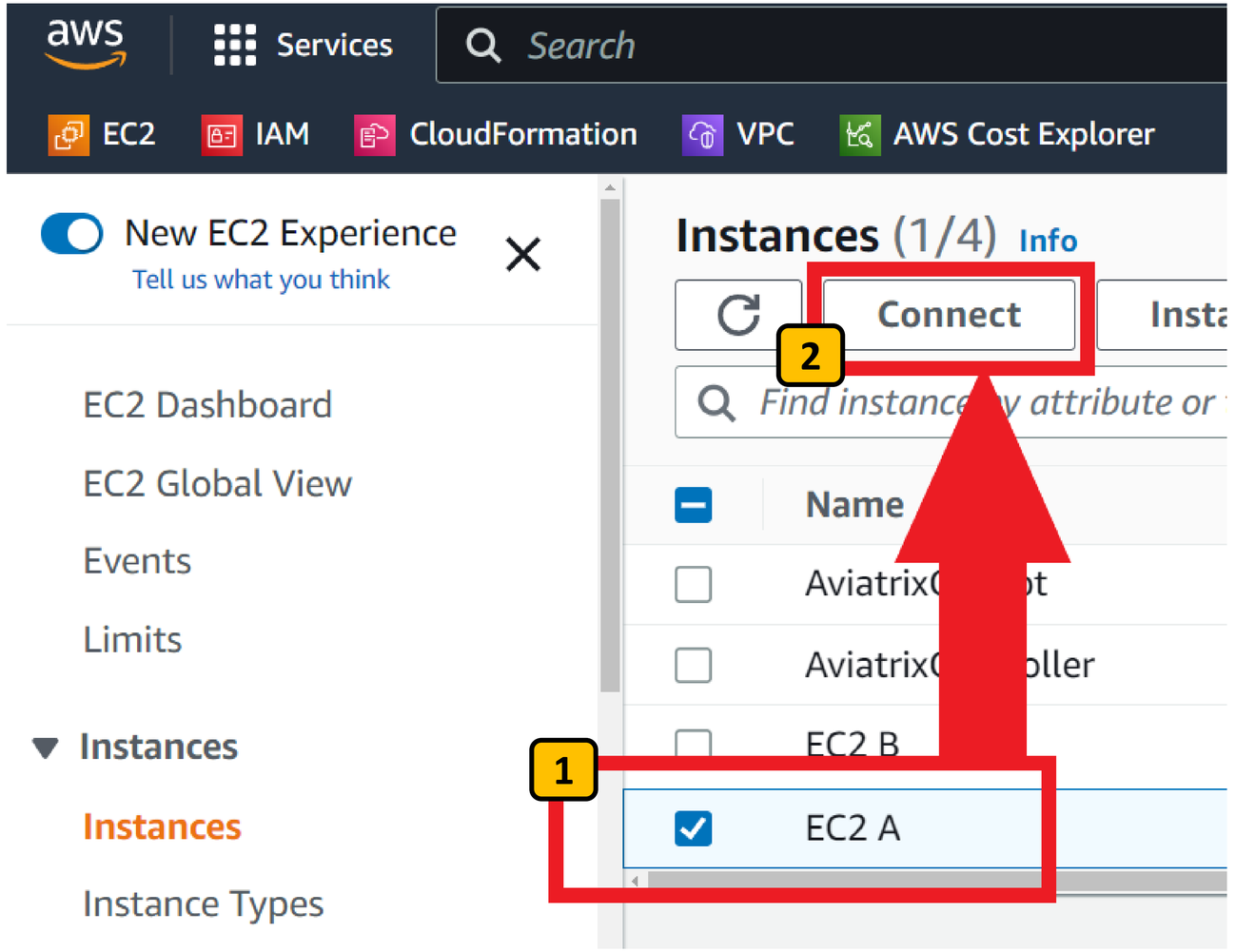
Connect to EC2 A instance console

Go the EC2 section of the AWS Console.

Select **Instances**

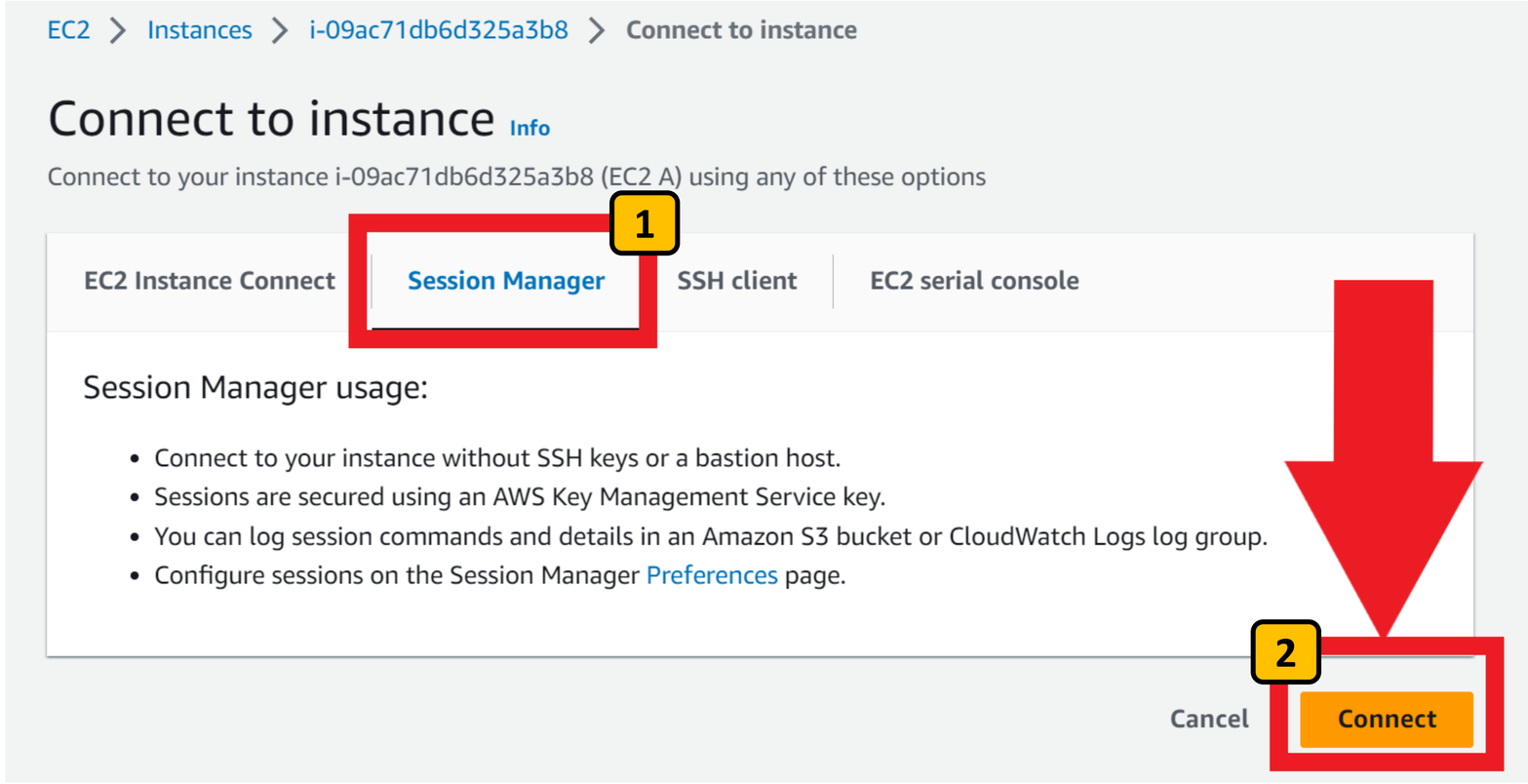
Find the **EC2 A** instance and select it. 1

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.

Next, let’s test internet egress.

Lab 2: Step 2.5

Test connection to google.com

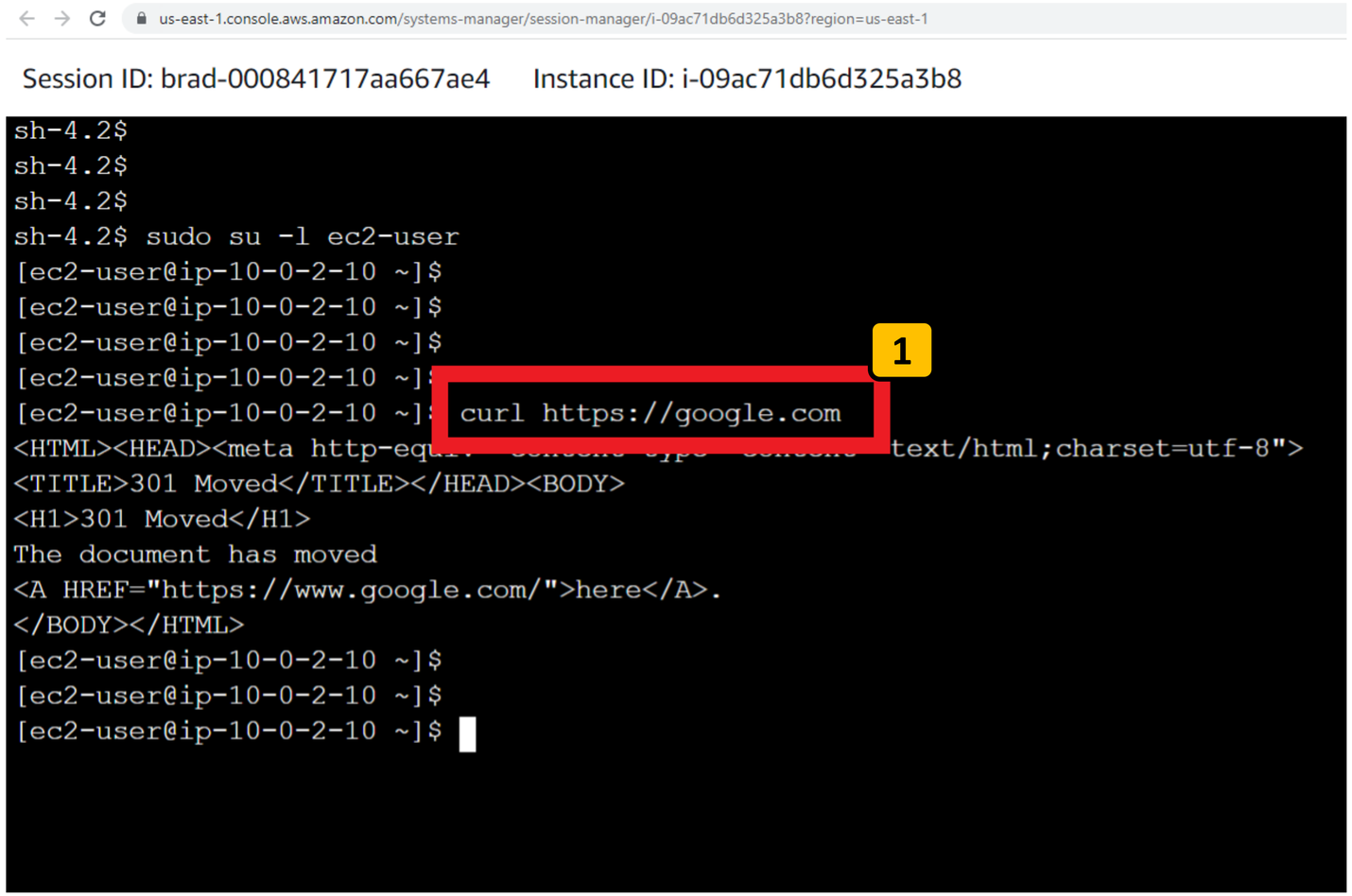
Let’s test a connection to google.com

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…



Lab 2: Step 2.6

Run a software update

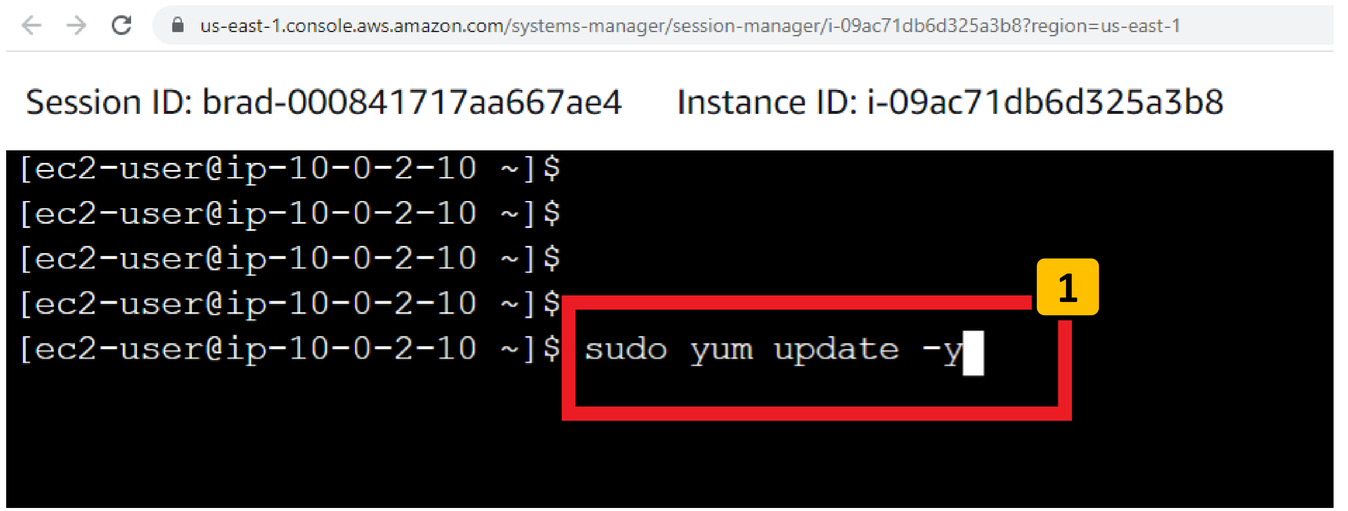
Let’s run a software update from the AWS repositories

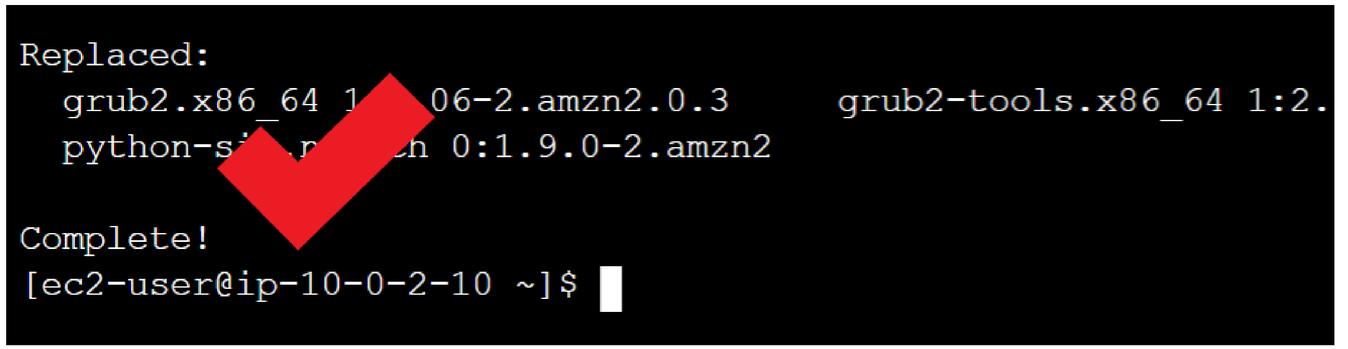
Run the command:
sudo yum update -y 1

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…





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 2

curl https://botnet.com 3

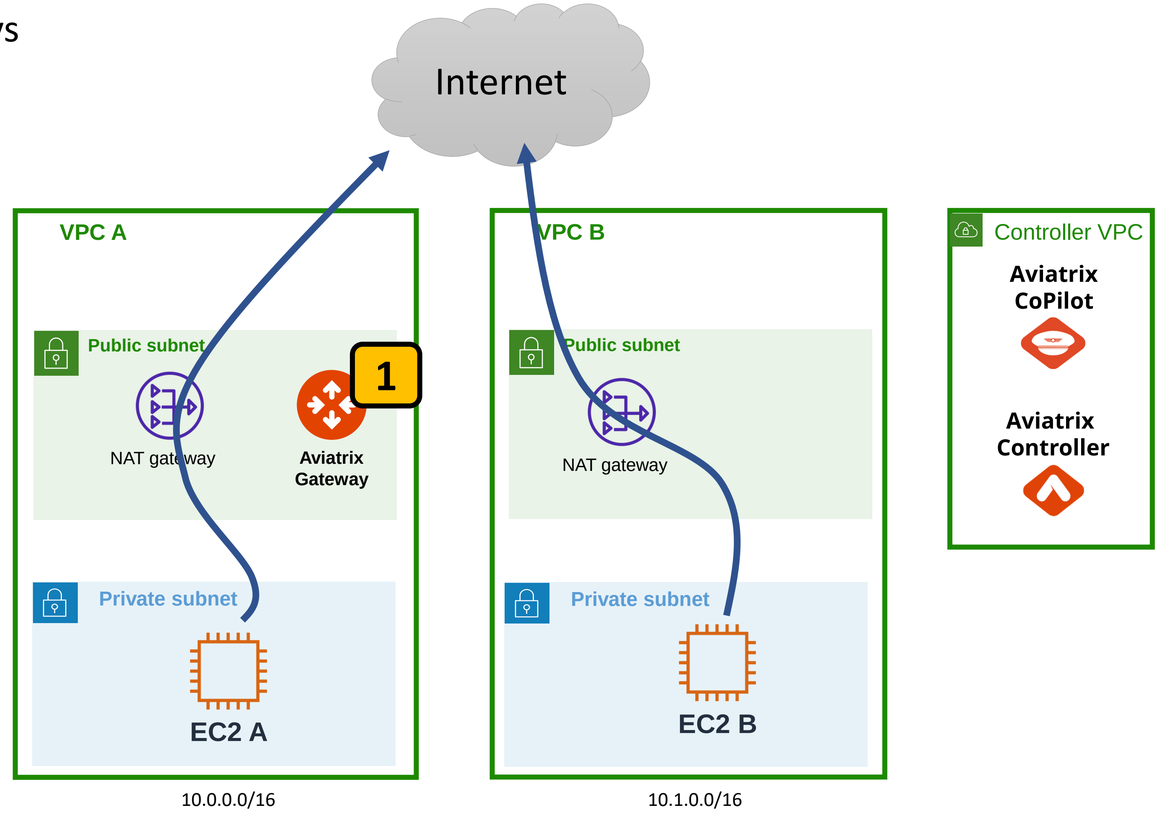
For each command you should see the CLI return HTML code from those domains … a successful connection 4

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



Lab 2: Step 2.8

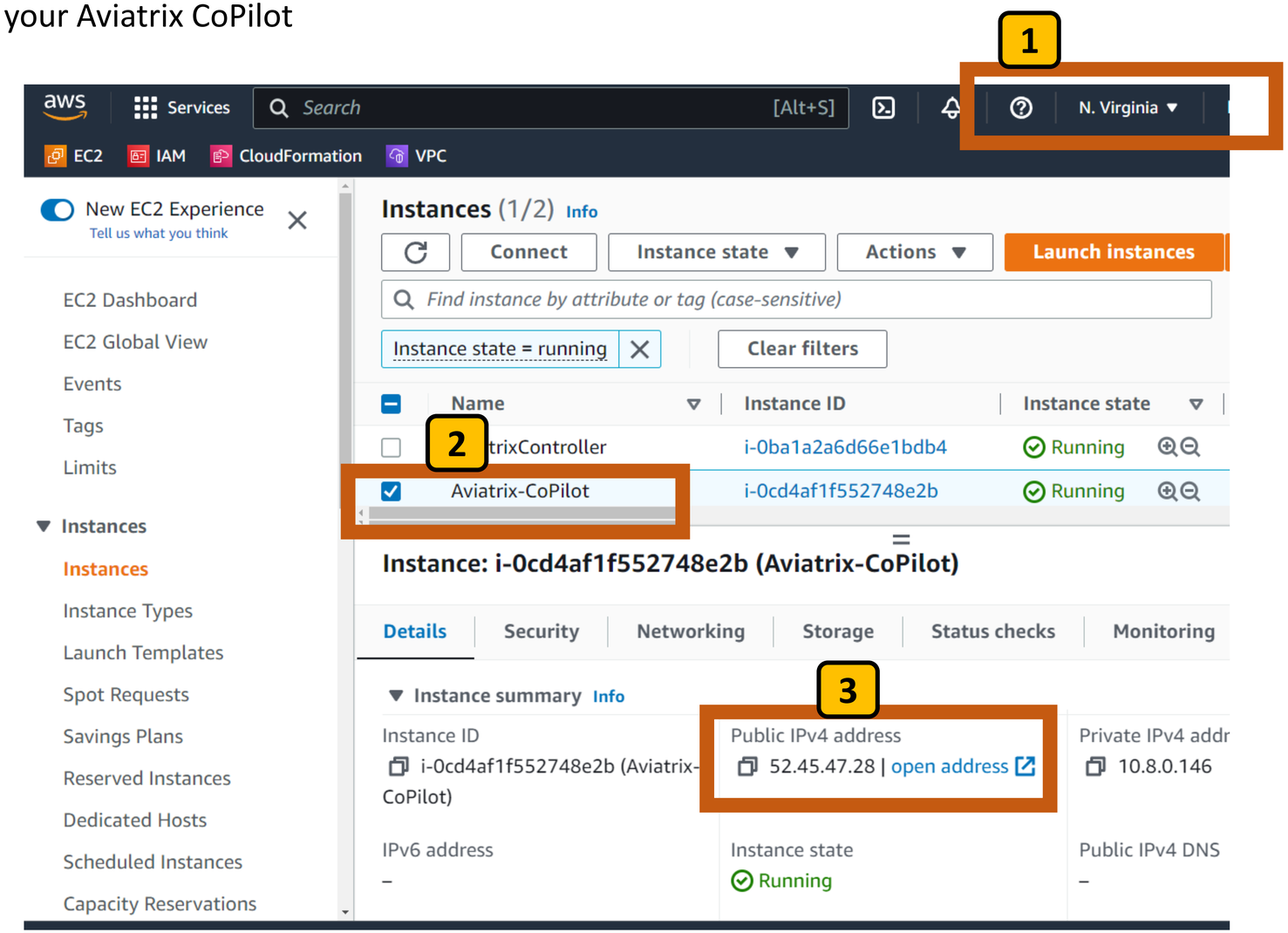
Log in to your Aviatrix CoPilot

Go 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



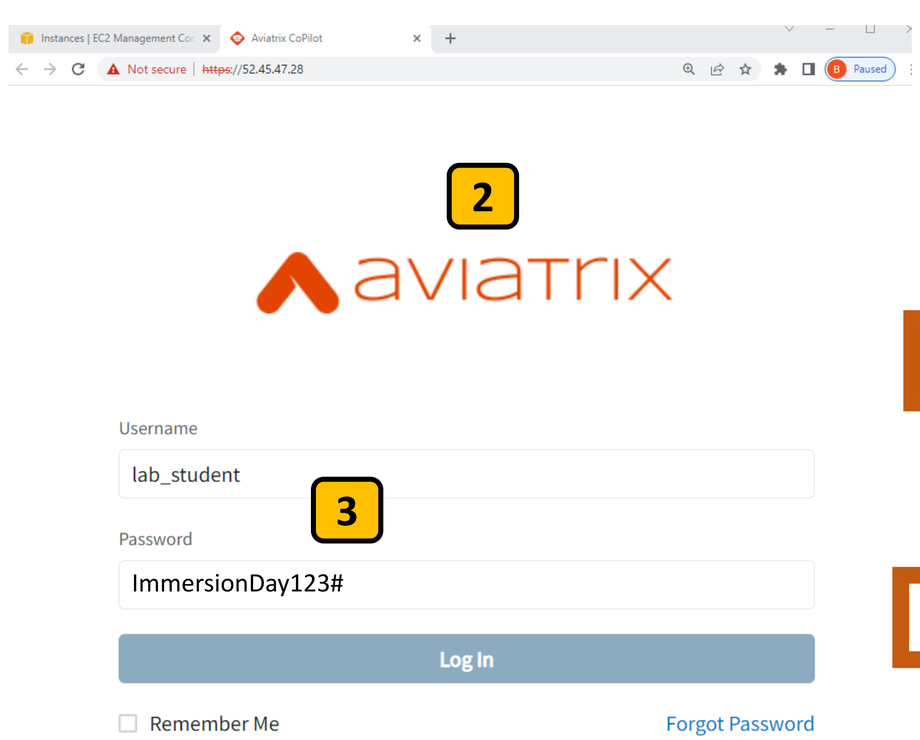
Lab 2: Step 2.9

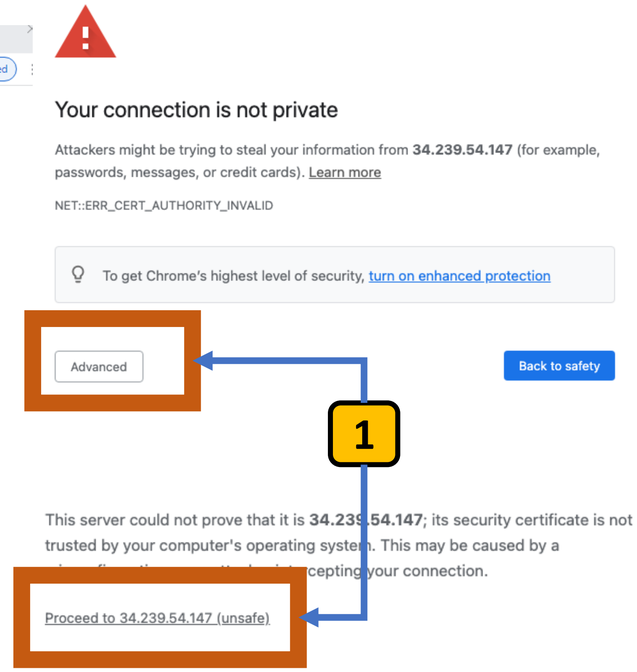
Log in to your Aviatrix CoPilot

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





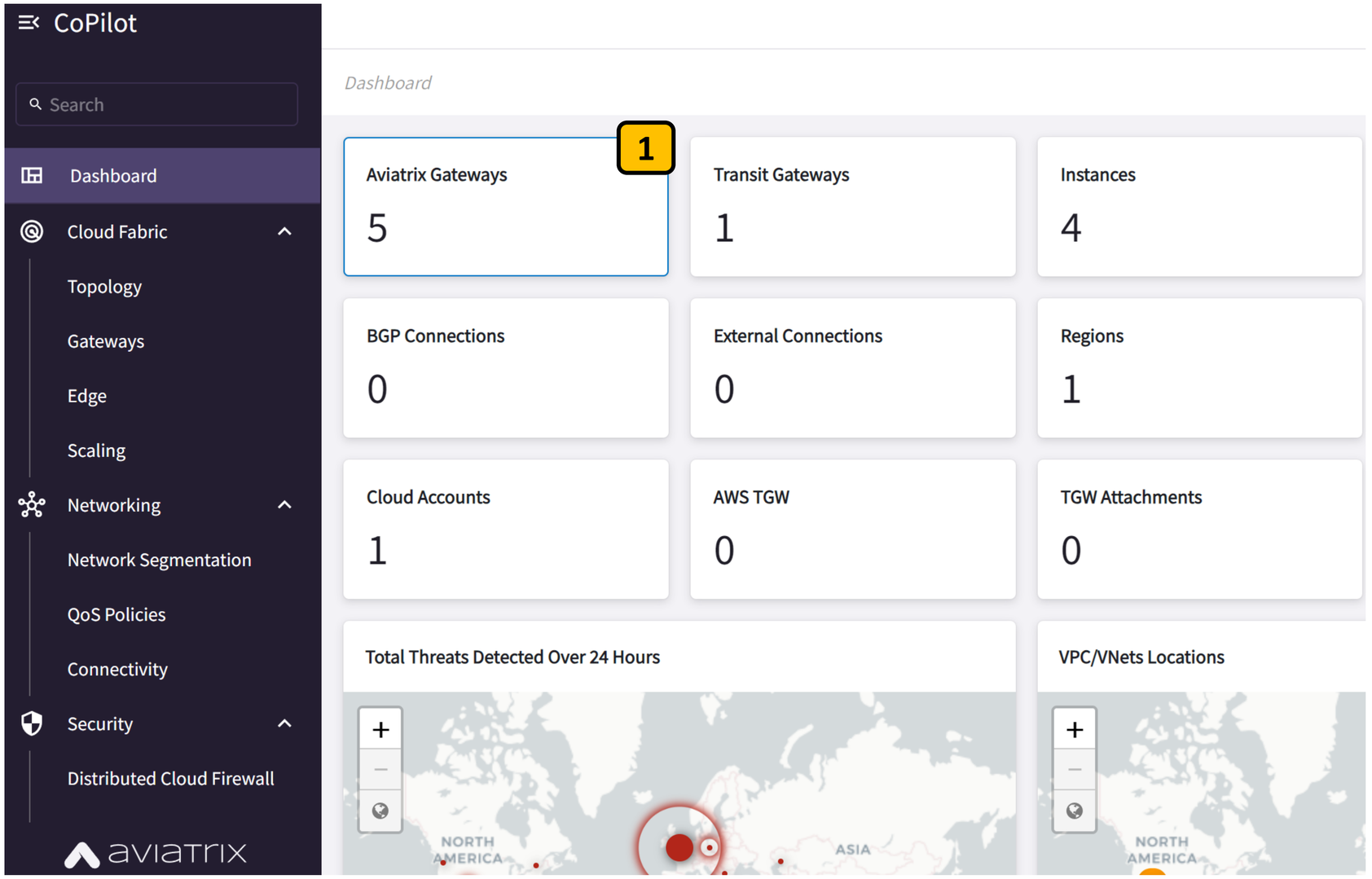
Lab 2: Step 2.10

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

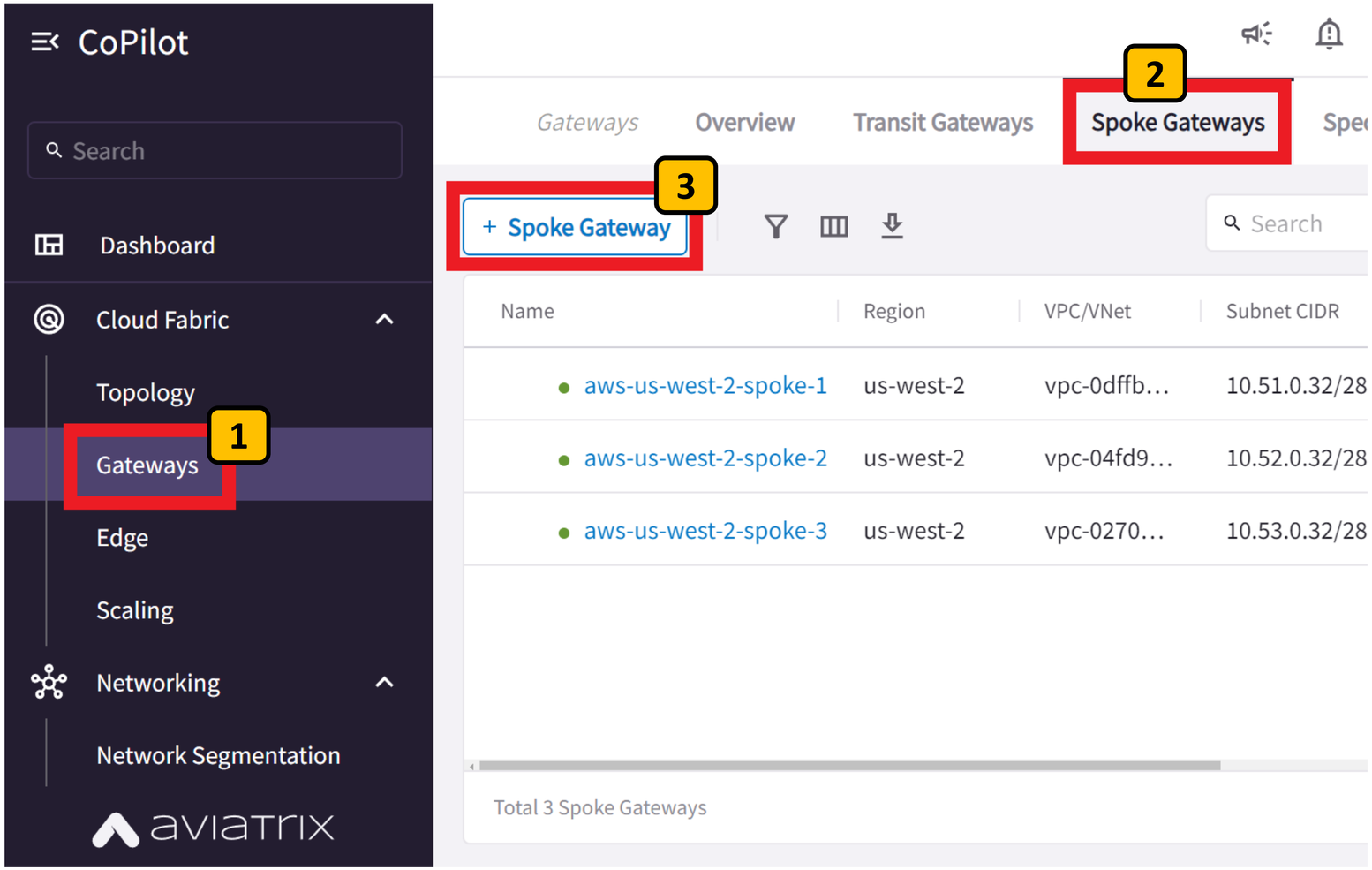
Deploy Aviatrix Spoke Gateway in VPC A Public Subnet

From the left-hand navigation in Aviatrix CoPilot…

Under Cloud Fabric select **Gateways** 1

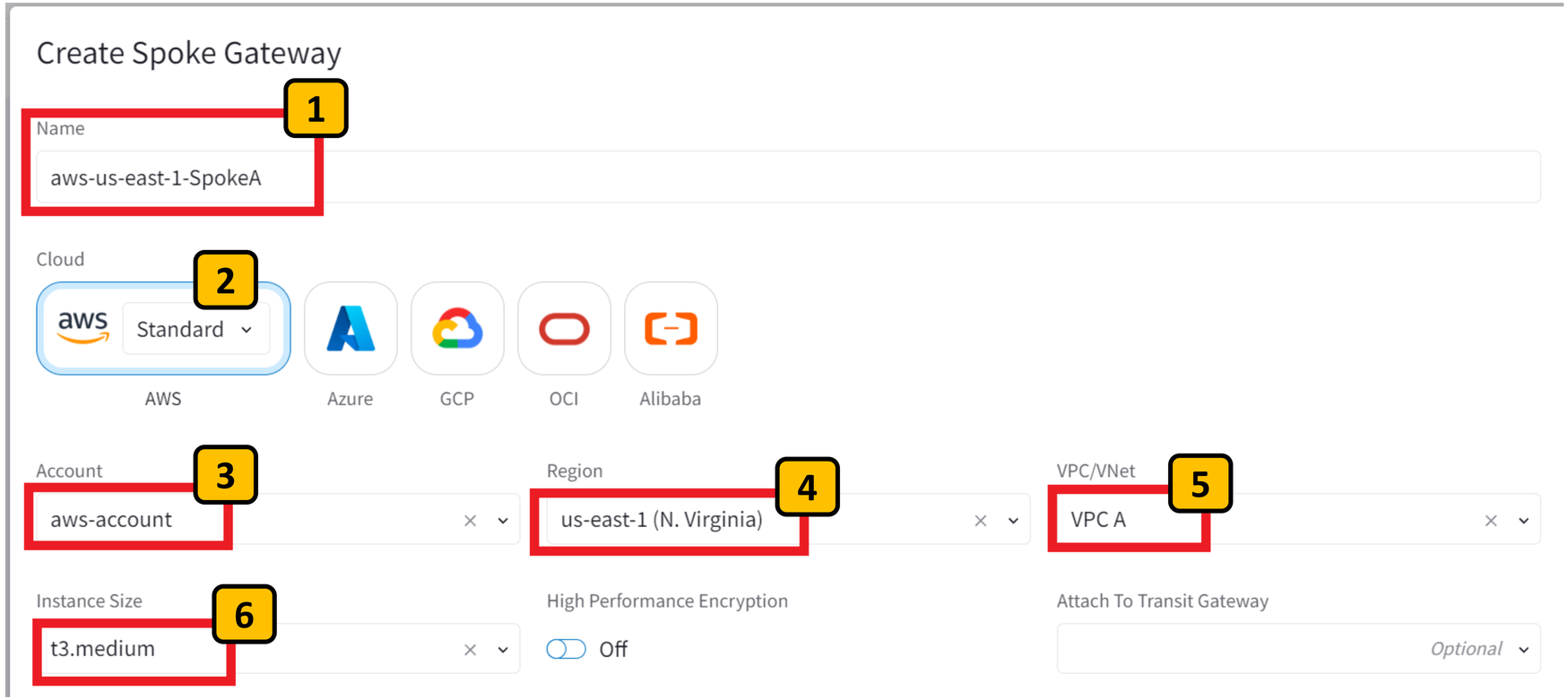
Select the **Spoke Gateways** tab 2

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



Lab 2: Step 2.12

Deploy Aviatrix Spoke Gateway in VPC A Public Subnet



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

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

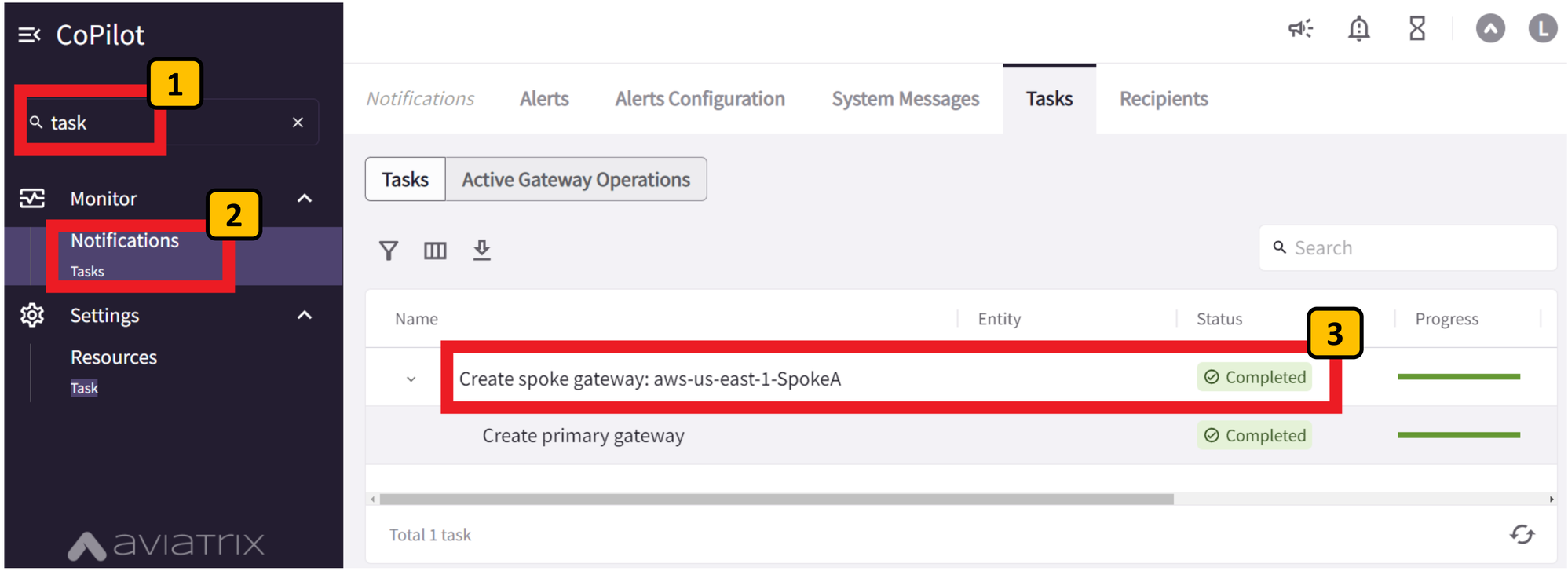
Leave the Public IP at the default selection of Allocate New

Click **Save** 8

Upon clicking Save, Aviatrix CoPilot will begin the gateway deployment

Lab 2: Step 2.14

Monitor the gateway deployment Task



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

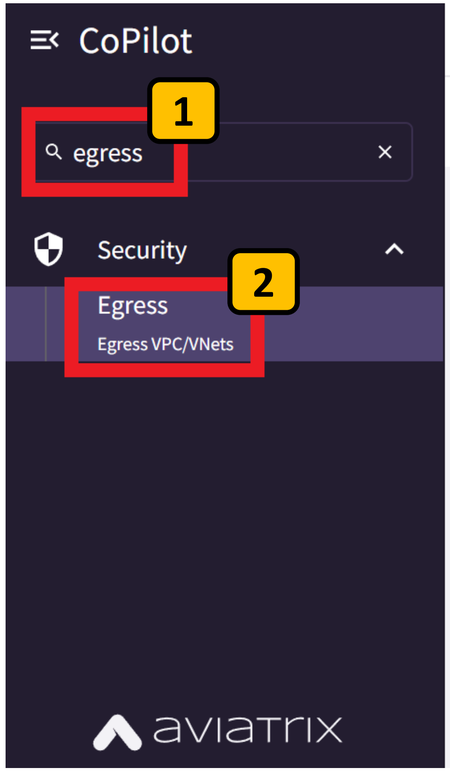
Click the search result **Notifications / Tasks** 2

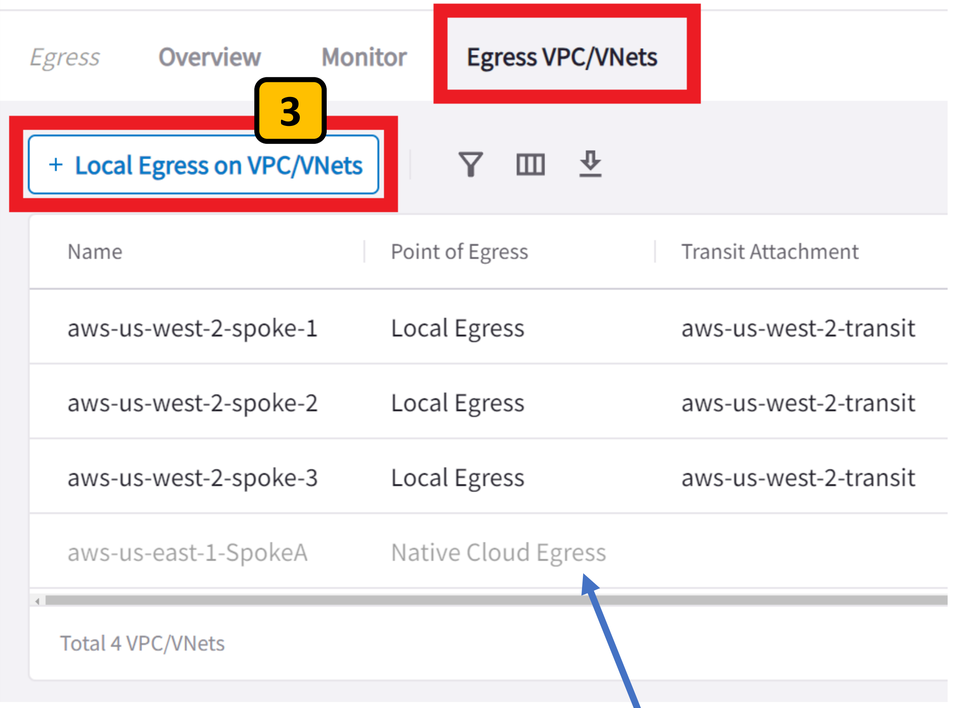
Observe the spoke gateway creation Task and wait for it to complete 3

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.





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

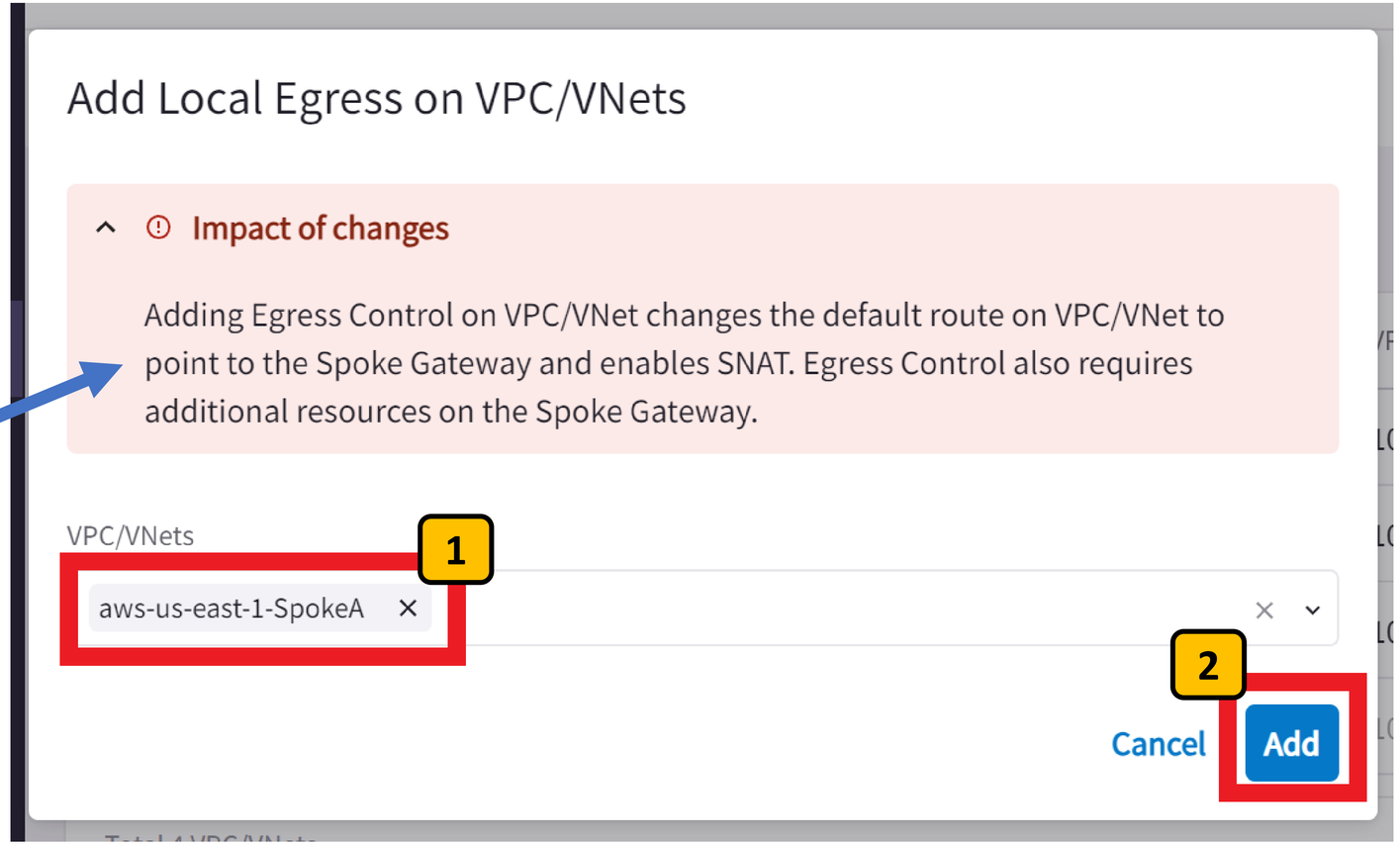
Lab 2: Step 2.16

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

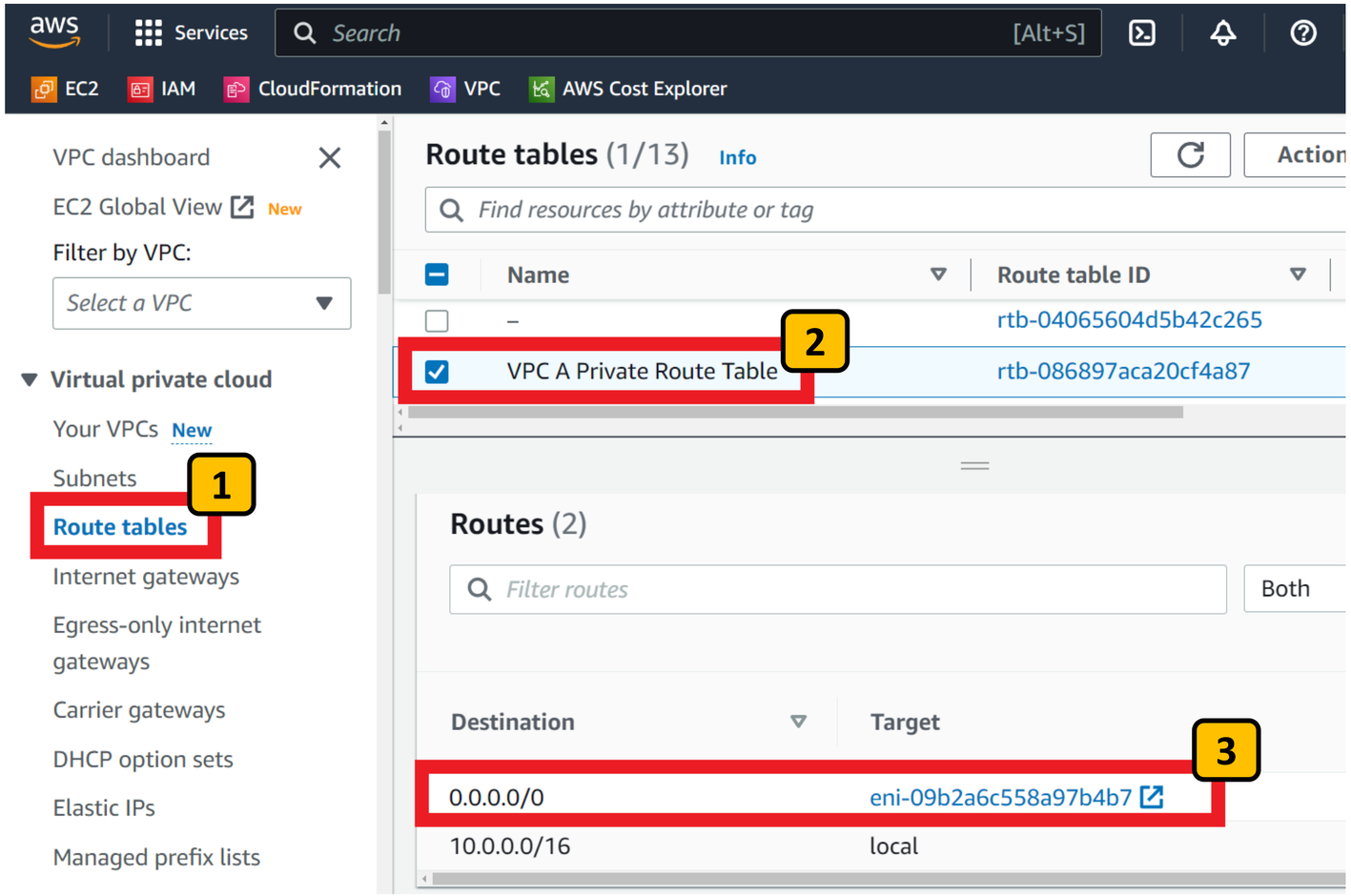
After enabling Egress on the Aviatrix Gateway, the Aviatrix Controller changed the default route in your VPC private route table.

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

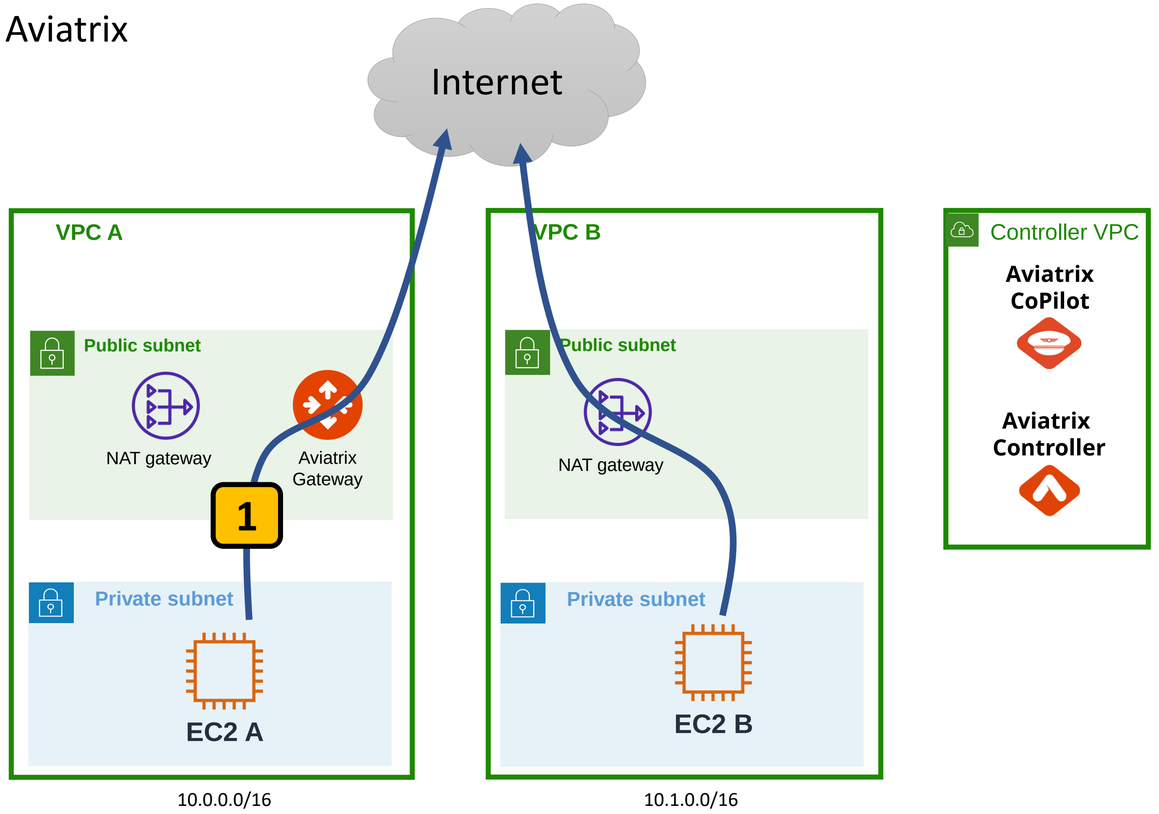
Simple egress NAT using Aviatrix

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



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)

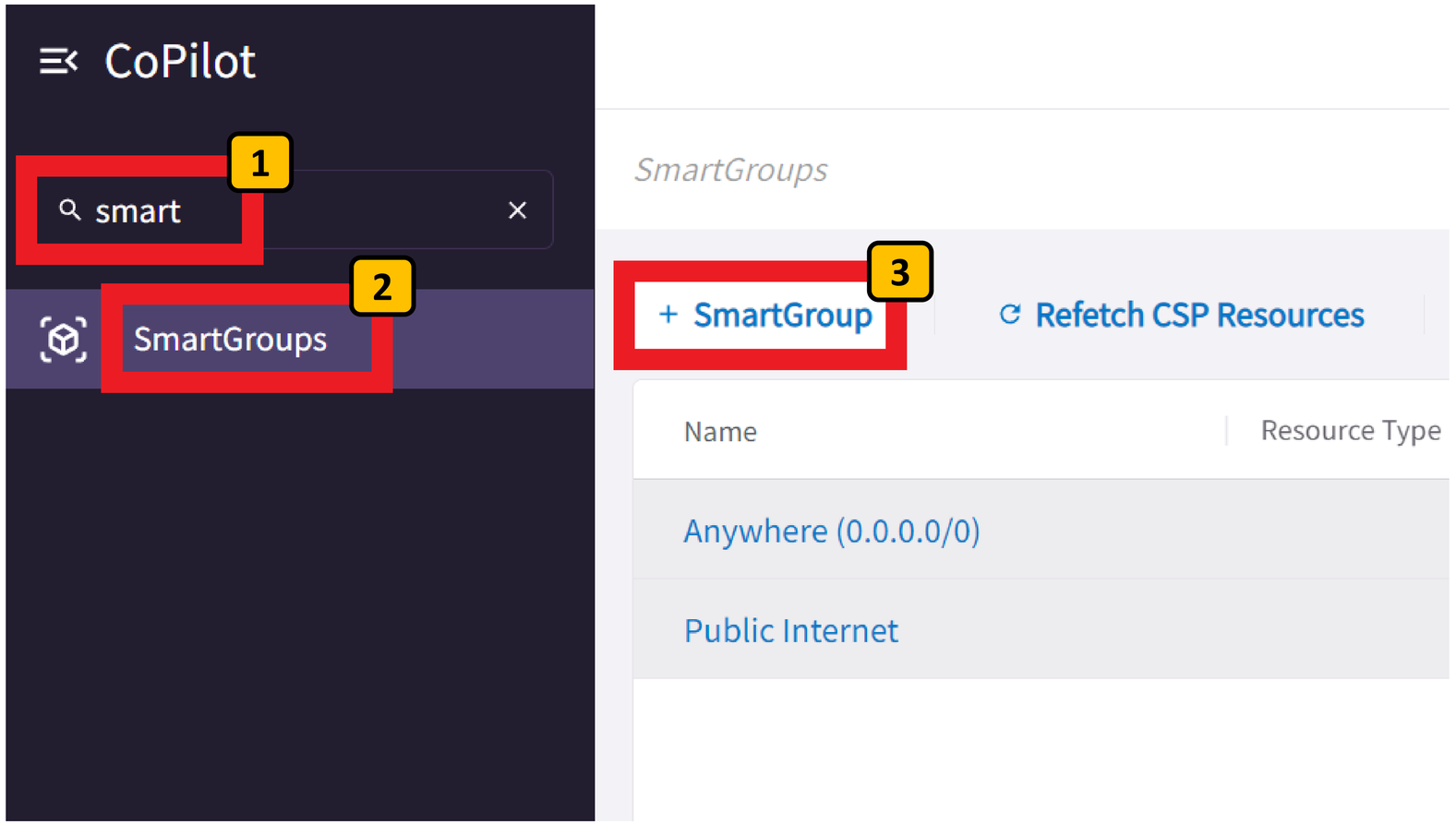
Lab 2: Step 2.18

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

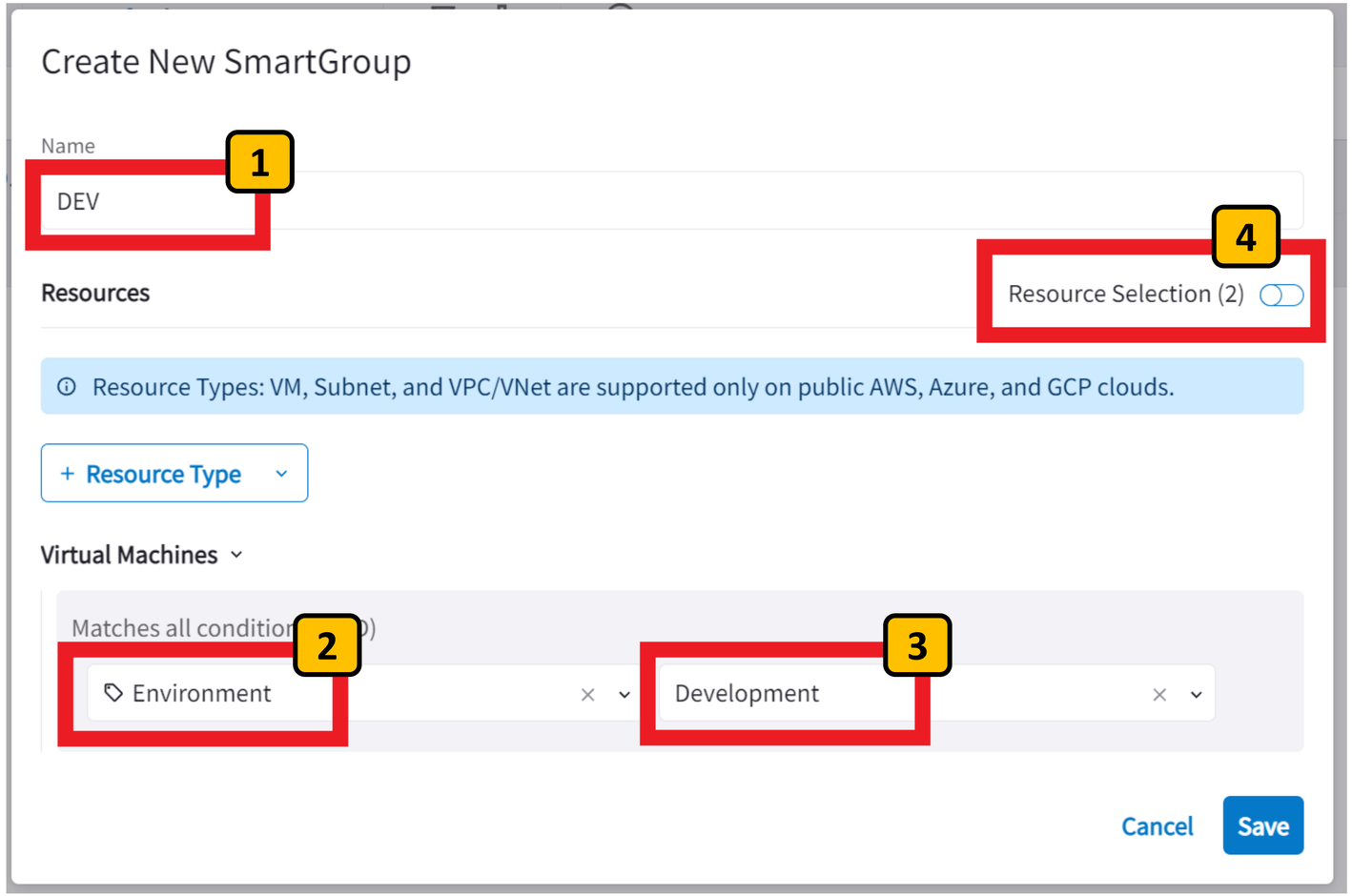
We want all EC2 instances tagged **Environment = Development** to be in this SmartGroup.

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

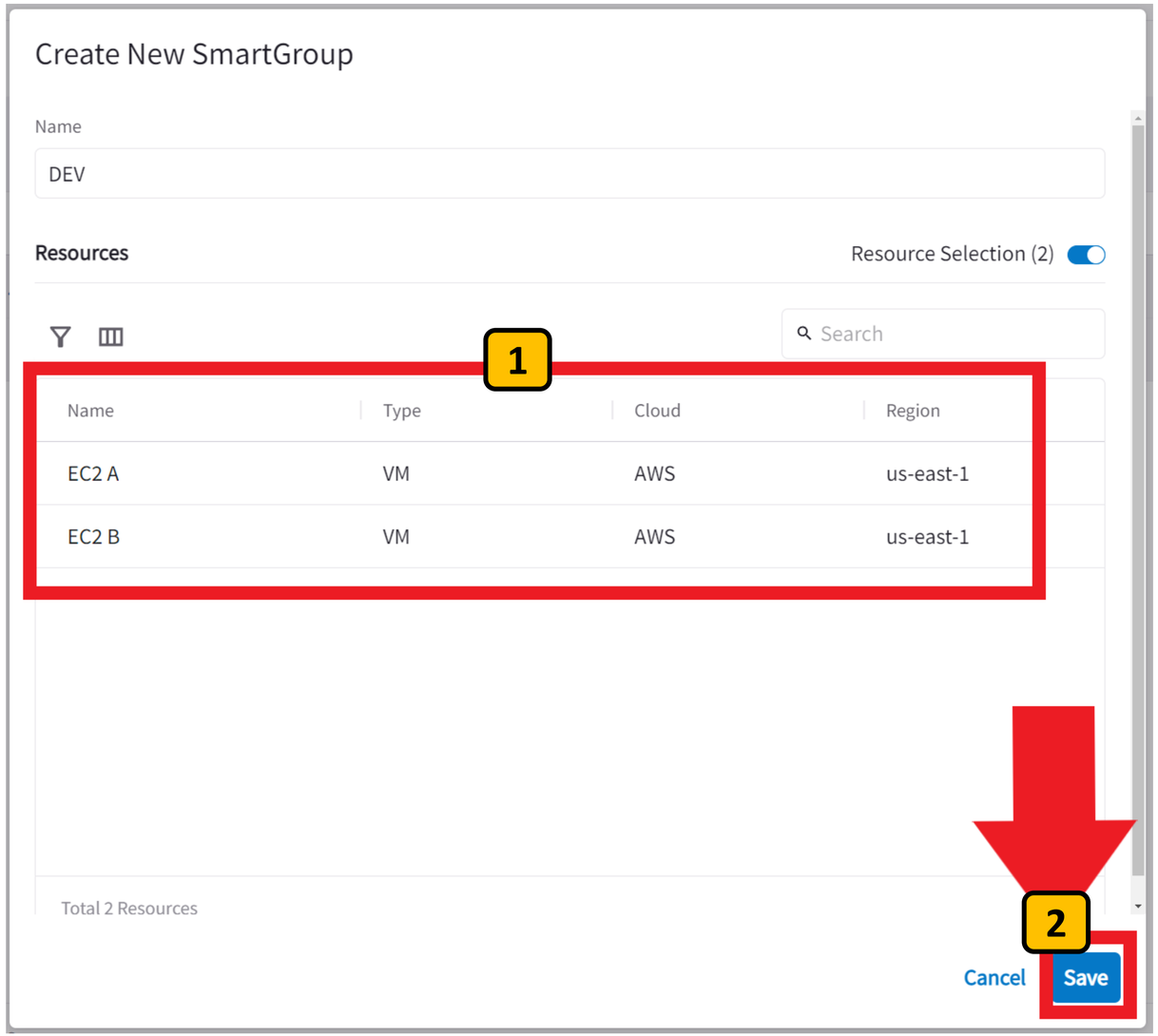
Create a SmartGroup to use for firewall rules

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

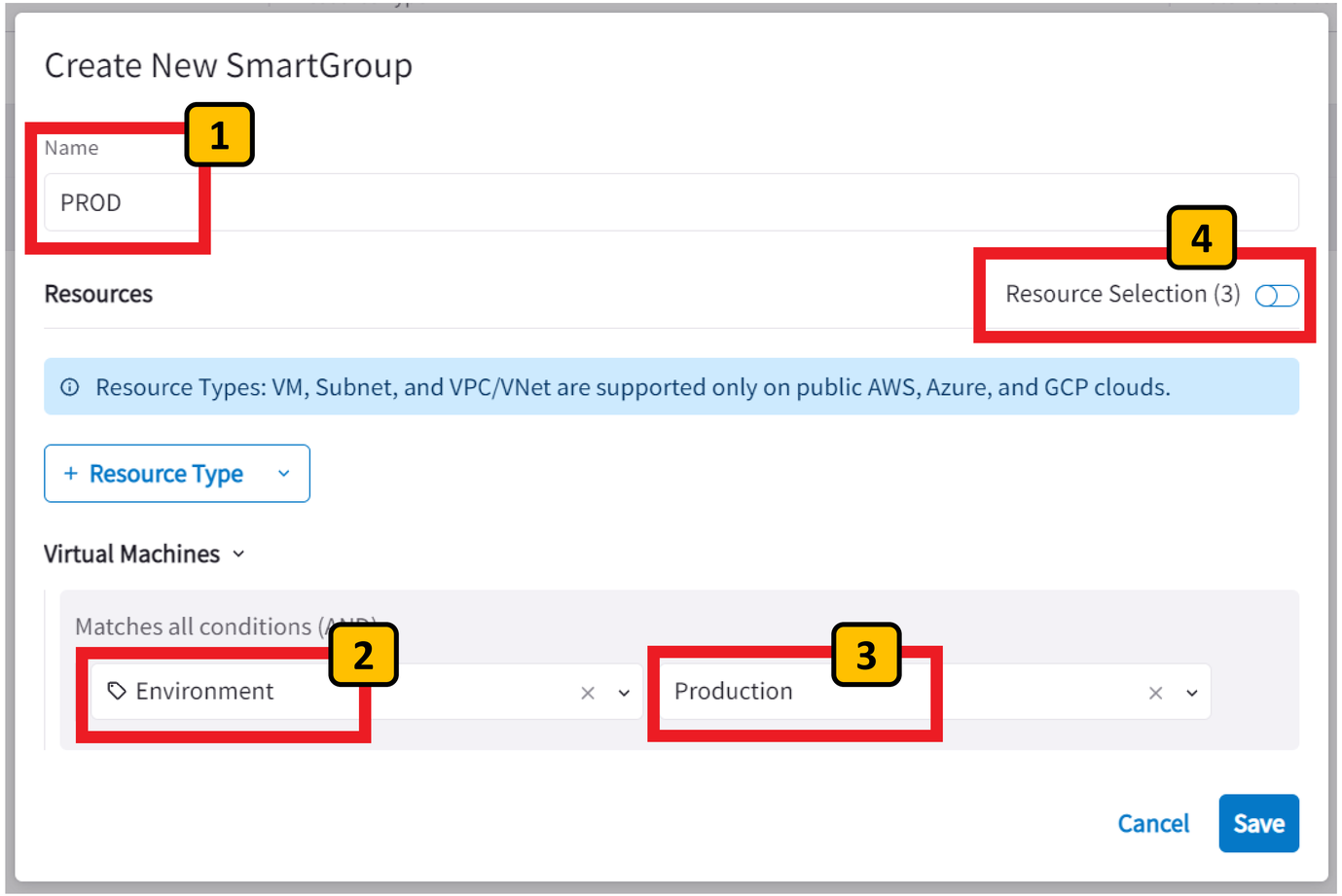
Create another SmartGroup and name it **PROD** 1

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



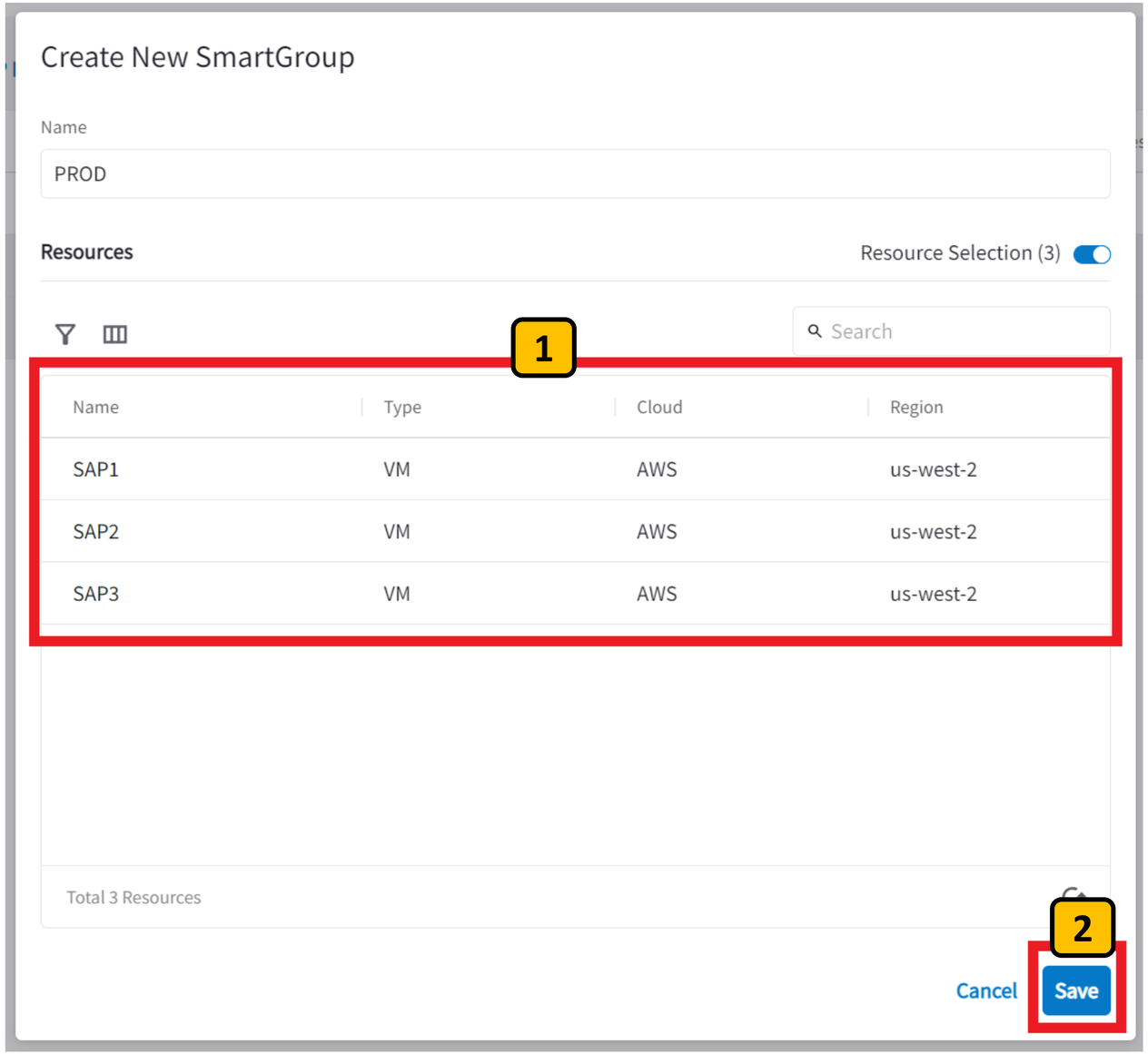
Lab 2: Step 2.22

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 in the us-west-2 region in the SmartGroup 1

Click **Save** to create the SmartGroup 2



Lab 2: Step 2.23

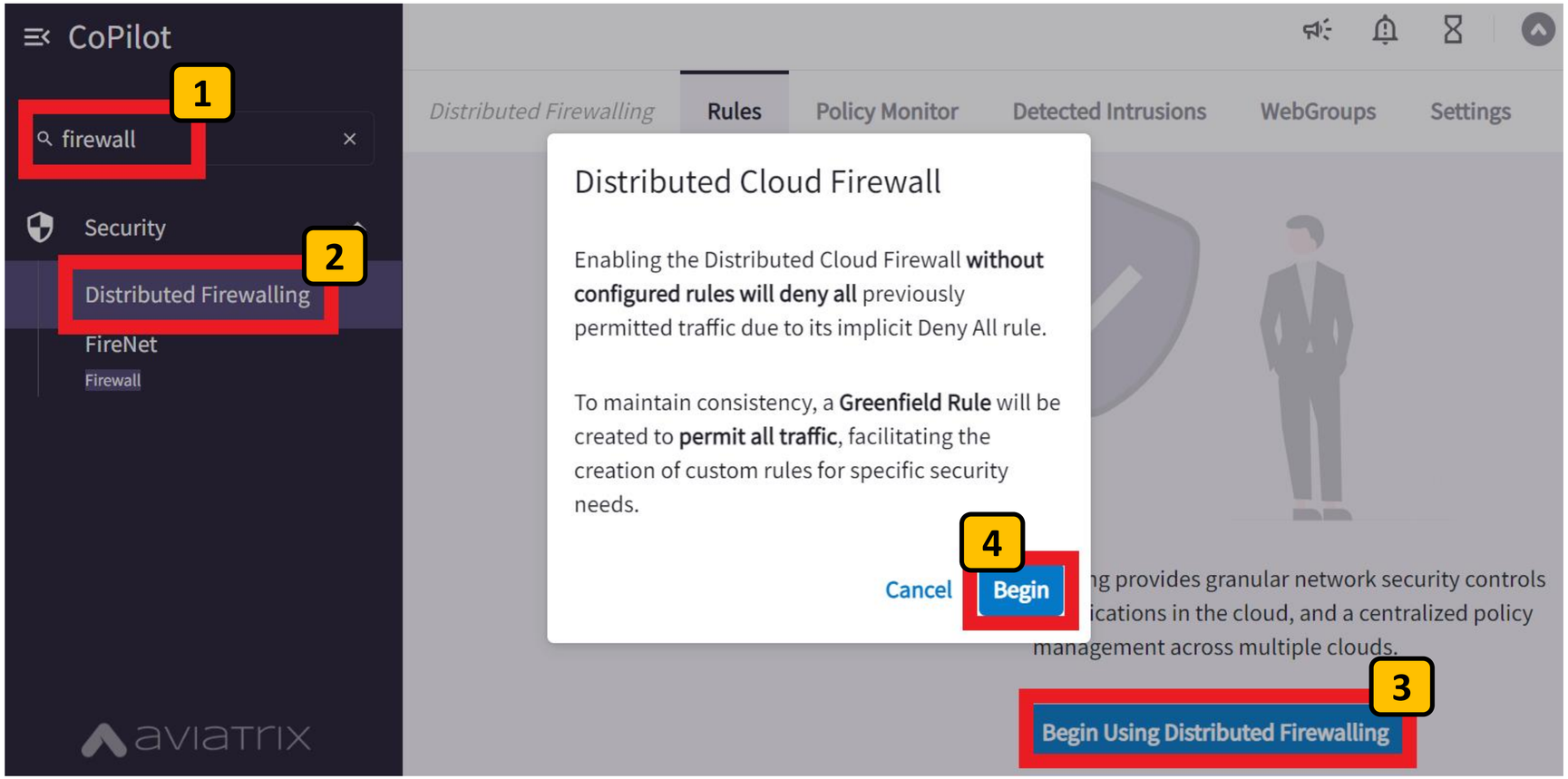
Enable Aviatrix Distributed Firewalling

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

Click the search result **Distributed Firewalling** 2

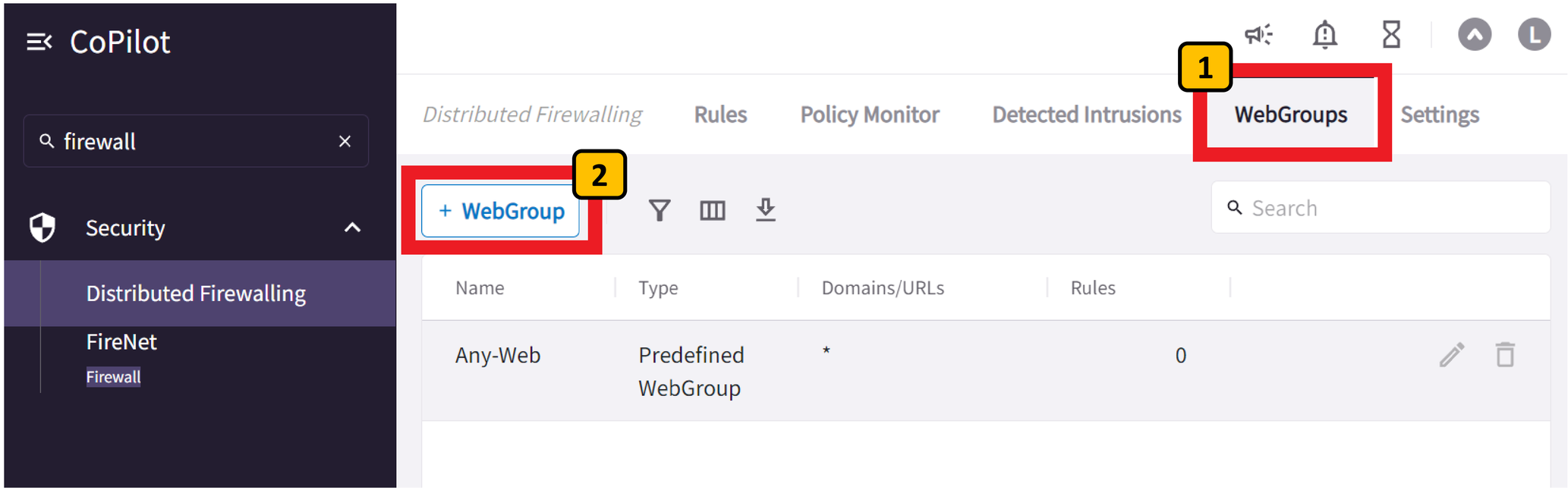
Click **Begin Using Distributed Firewalling** 3

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



Lab 2: Step 2.24

Create a WebGroup to use for firewall rules



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

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

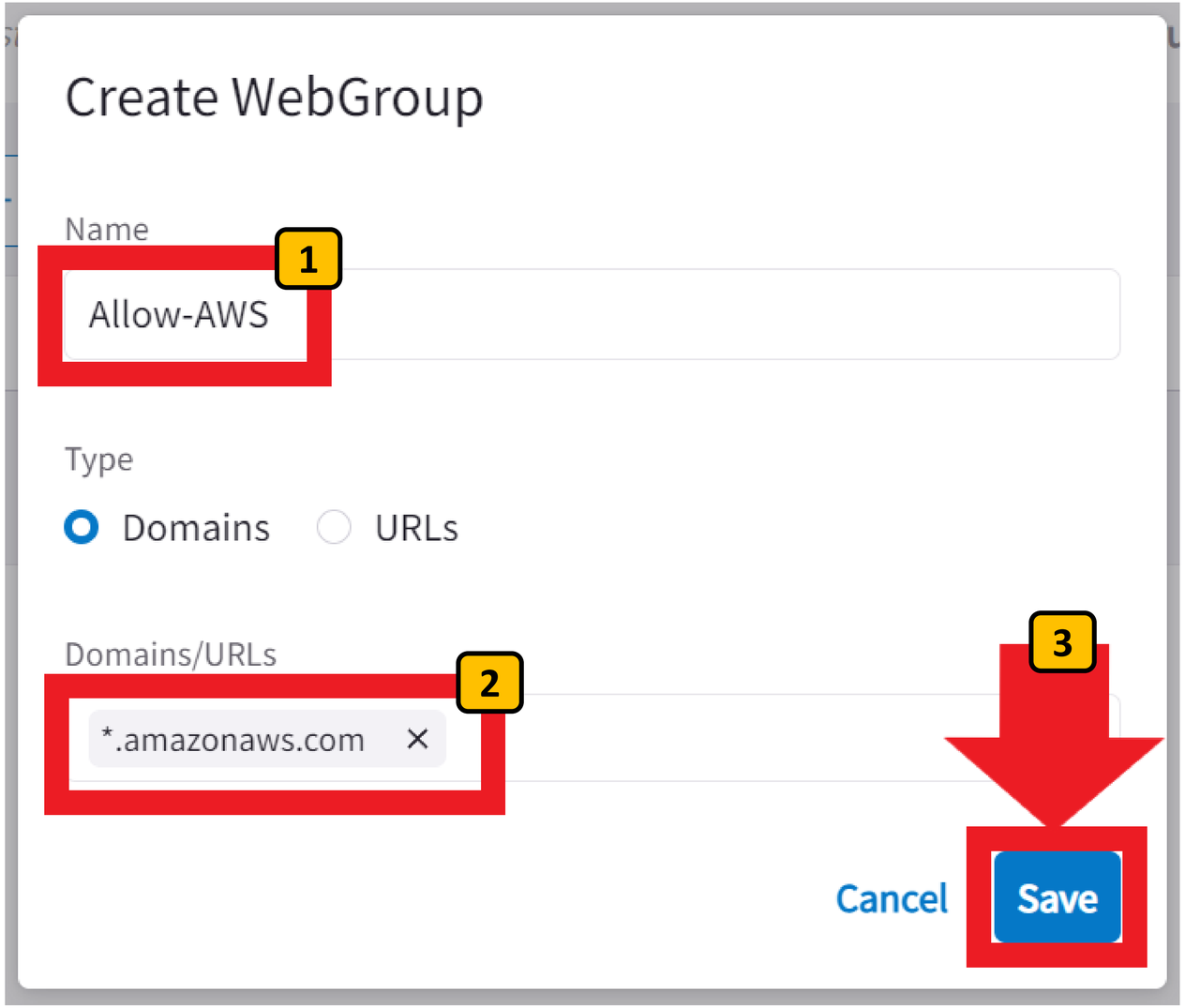
Lab 2: Step 2.25

Create a WebGroup to use for firewall rules

Name the WebGroup **Allow-AWS** 1

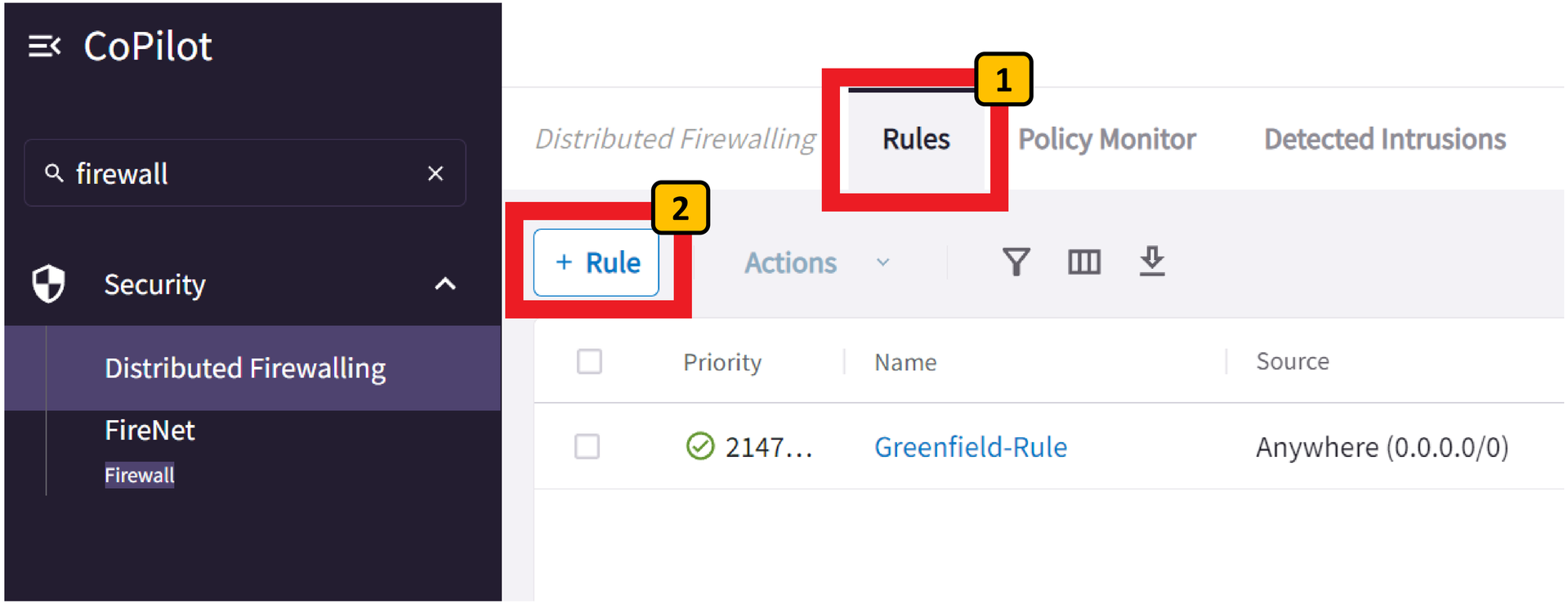
Enter the domain name
***.amazonaws.com** 2

Click **Save** 3



Lab 2: Step 2.26

Create a Distributed Firewall Rule



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

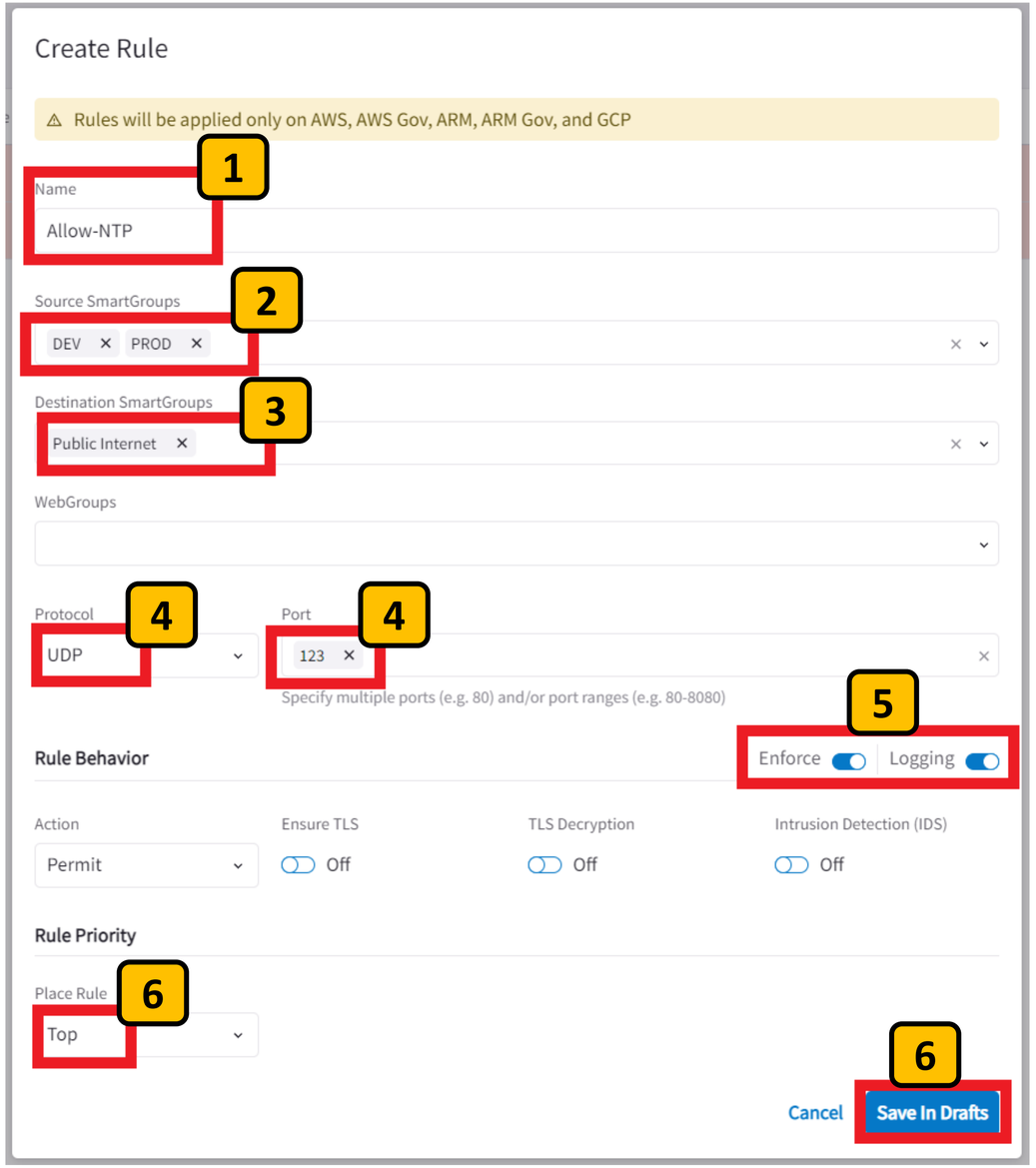
Chose the Source SmartGroups **DEV and PROD** 2

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

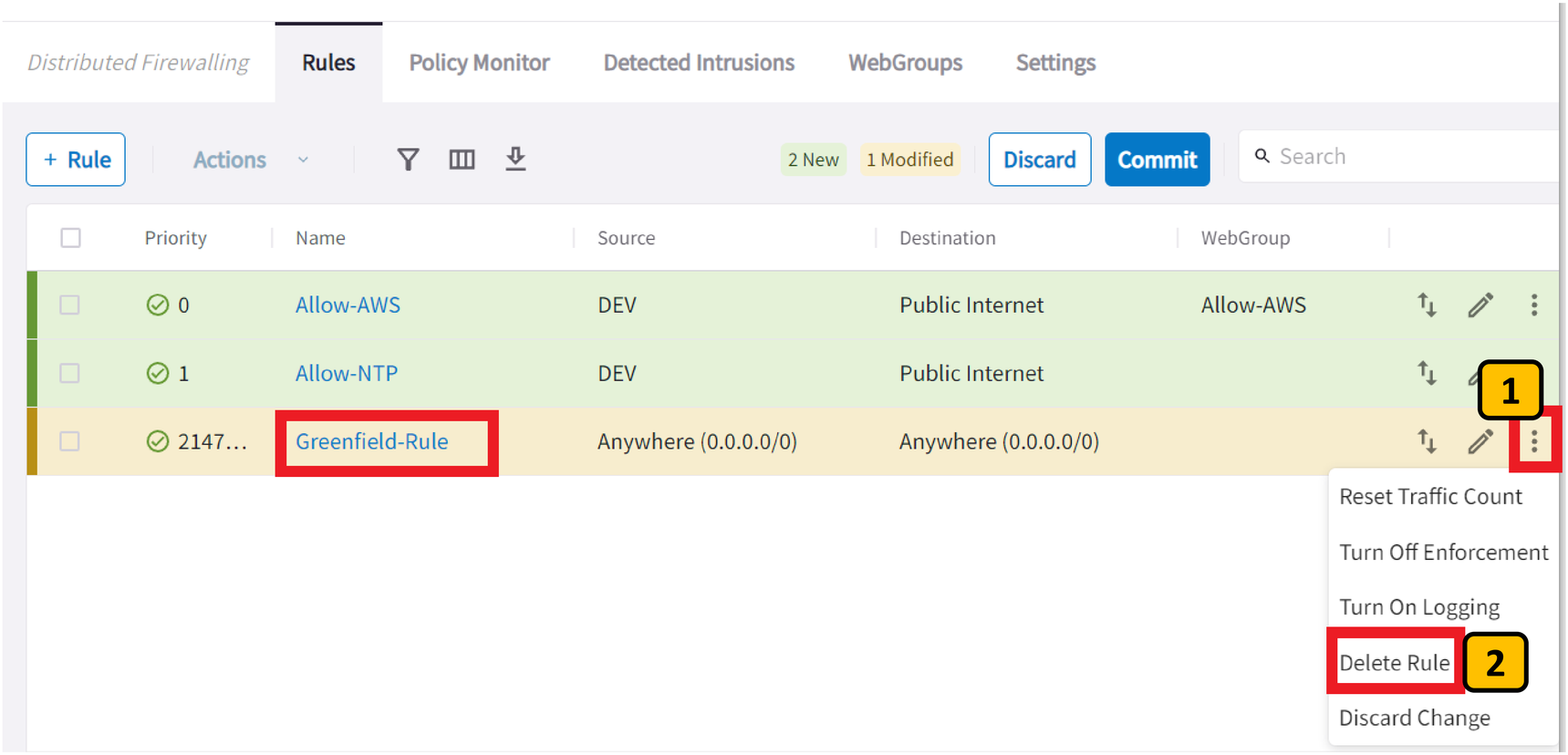
Choose the Protocol **TCP** and enter Port number **443** 5

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



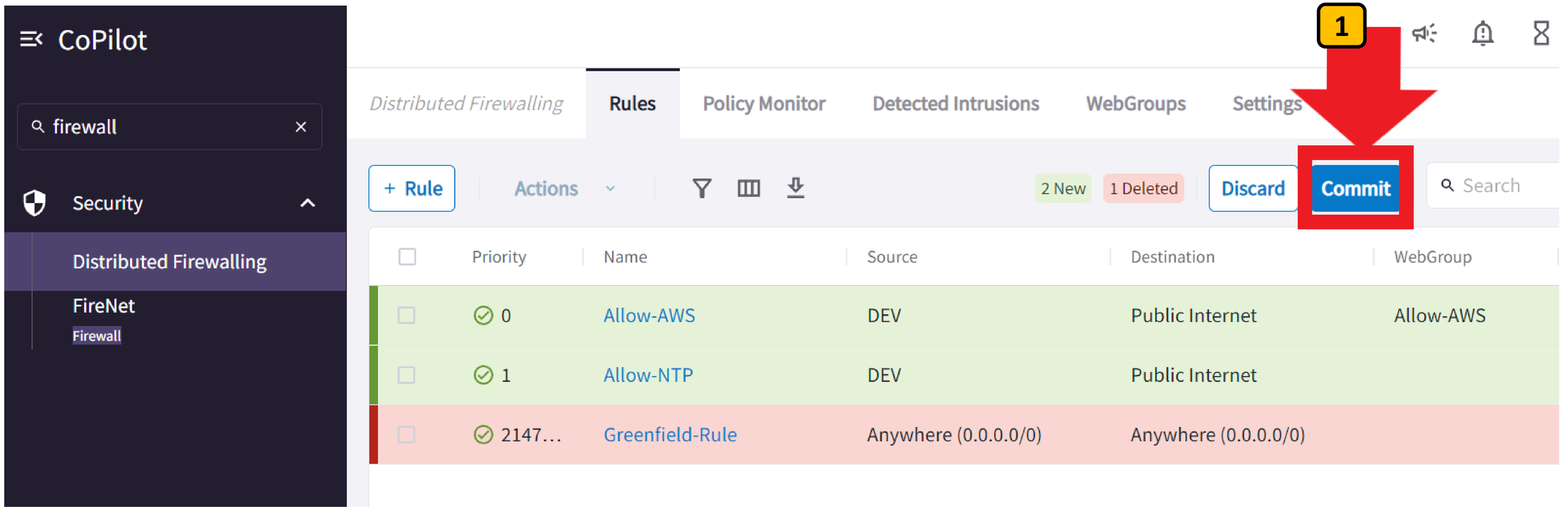
Find the **Greenfield-Rule** that permits all traffic and delete it

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



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

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

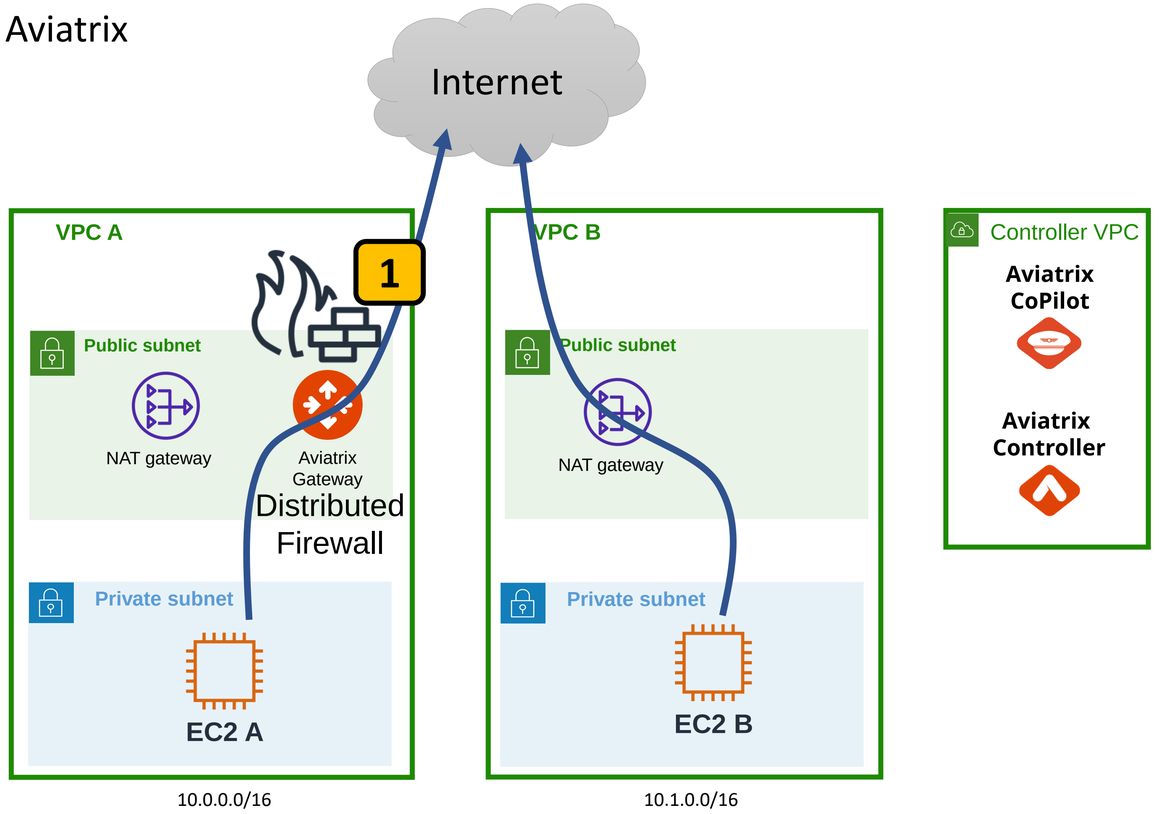
Lab 2: Checkpoint 3

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

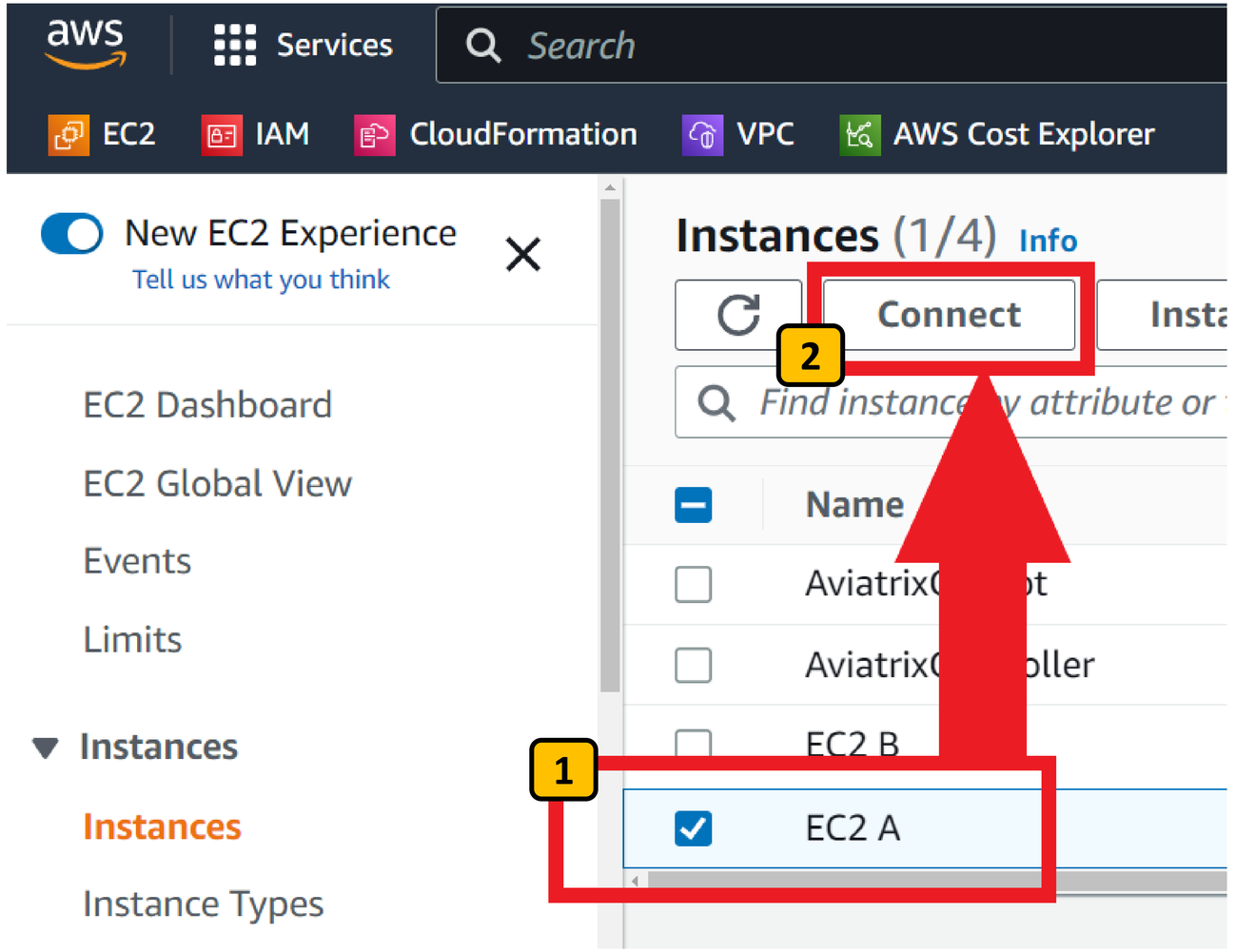
Connect to EC2 A instance console

Go the EC2 section of the AWS Console.

Select **Instances**

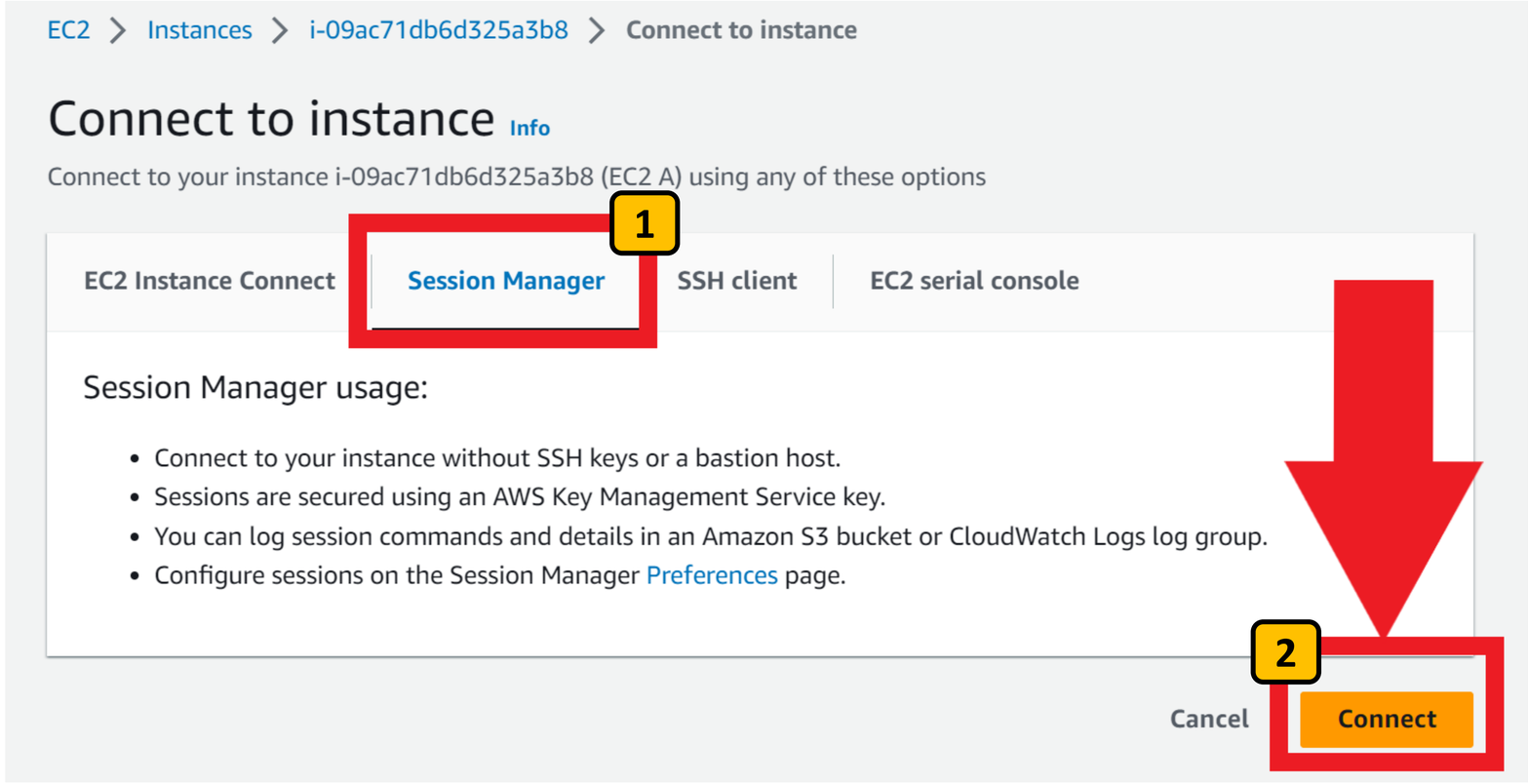
Find the **EC2 A** instance and select it. 1

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.

Next, let’s test internet egress.

Lab 2: Step 2.35

Test connections to potentially harmful domains

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 2.36

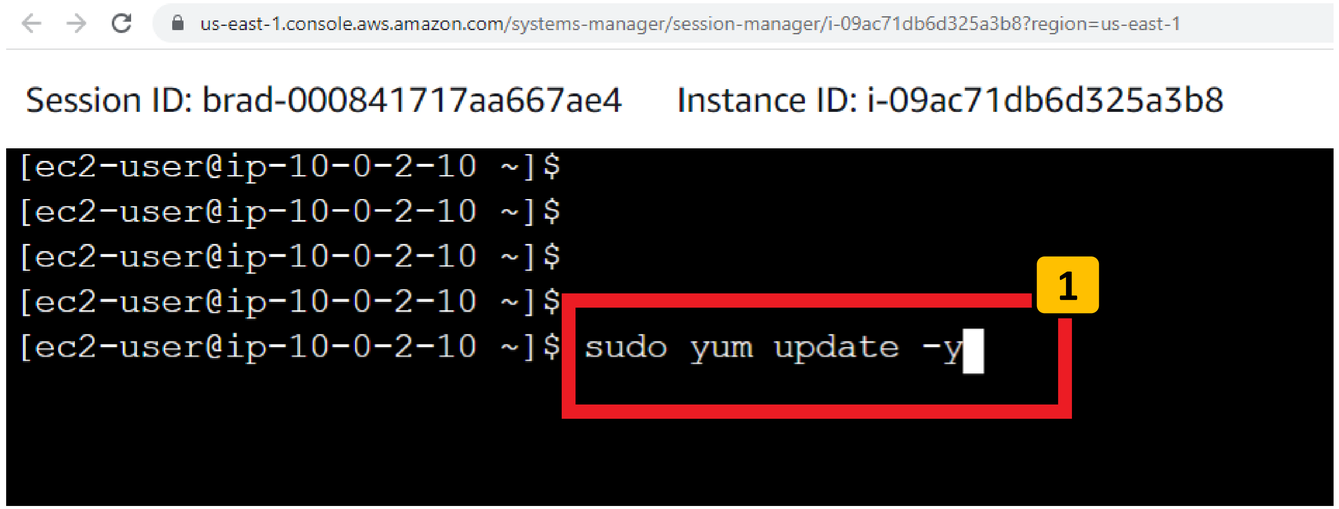
Run a software update

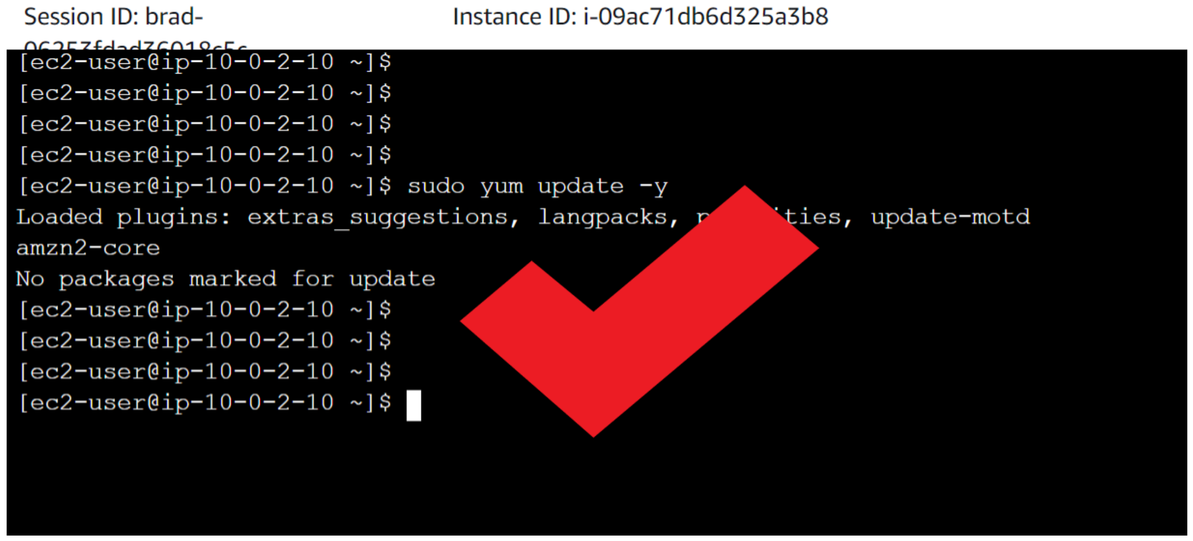
Let’s make sure we can still run a software update from the AWS repositories

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.





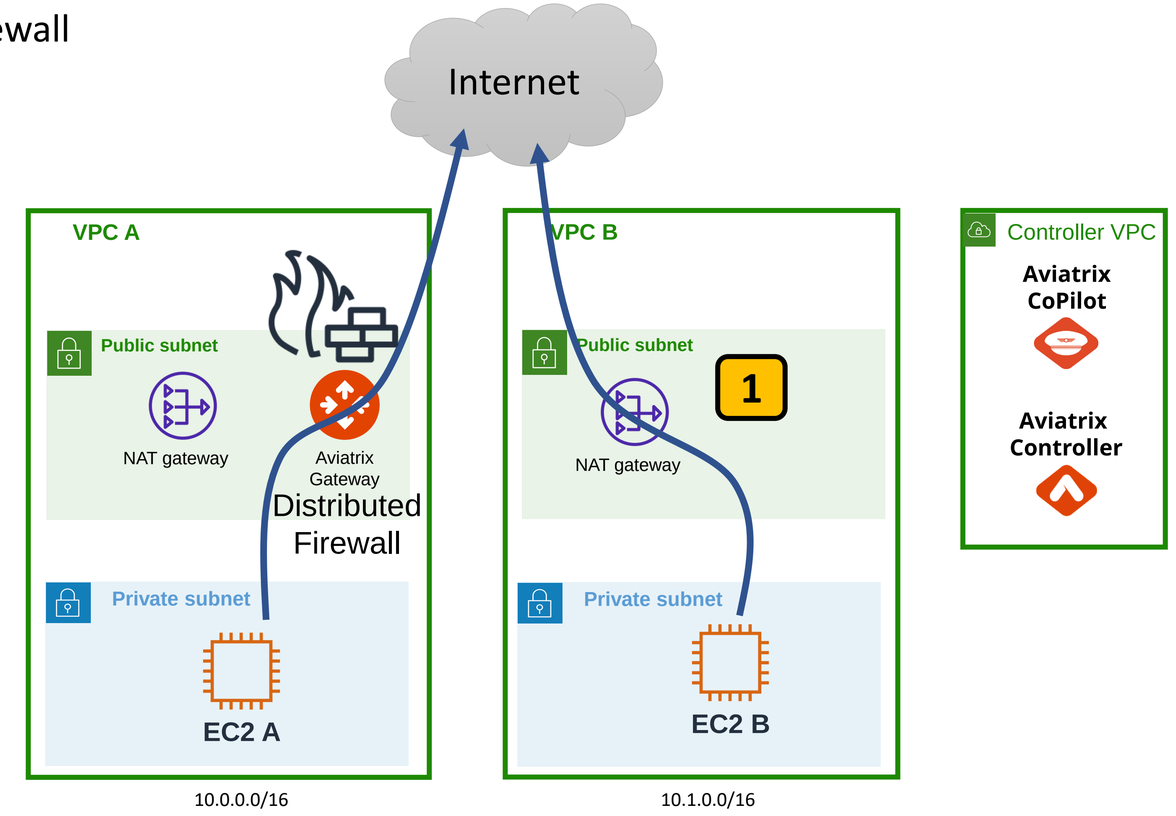
Lab 2: Checkpoint 4

Aviatrix Distributed Firewall

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

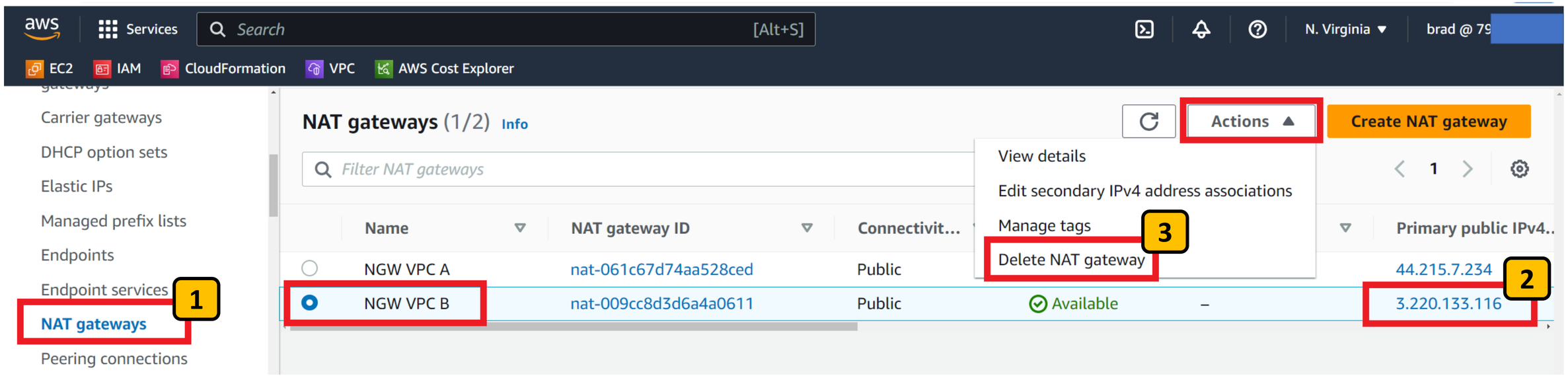
Your EC2 A instance will only be able access the internet according to the firewall rules you created.

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 using. Because maybe you have that EIP whitelisted with a partner or customer you work with, and you want to keep it. 1



Lab 2: Step 2.37

Delete your AWS NAT Gateways



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

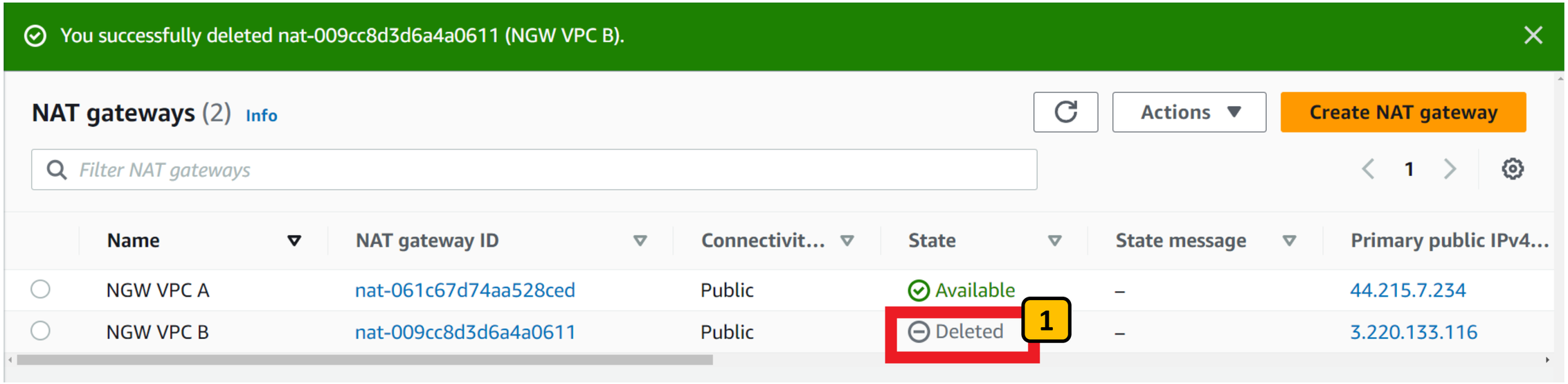
Find the NAT Gateway for VPC B and take note of the public IP address 2

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



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

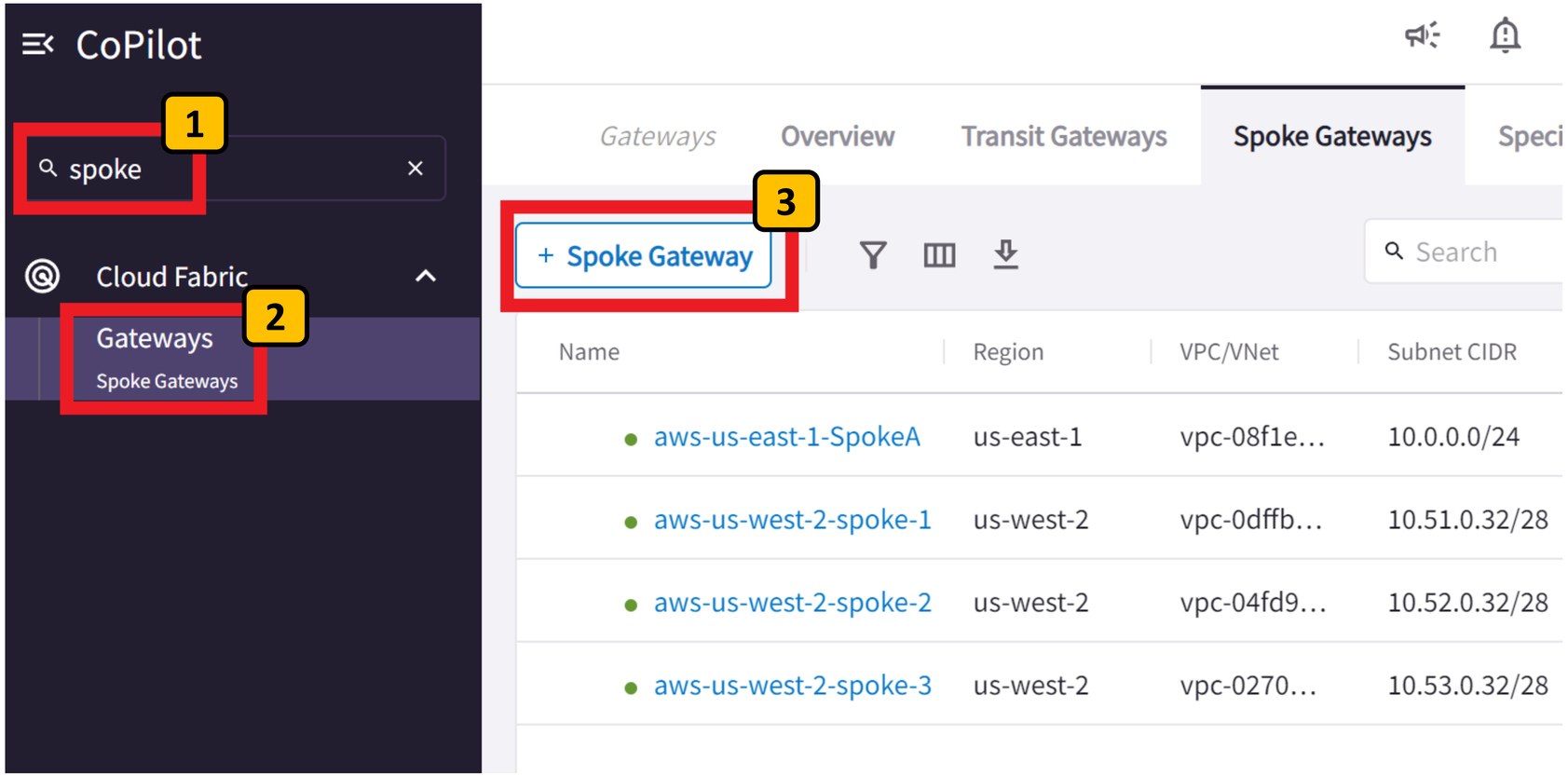
Deploy Aviatrix Spoke Gateway in VPC B Public Subnet

From the left-hand navigation in Aviatrix CoPilot…

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

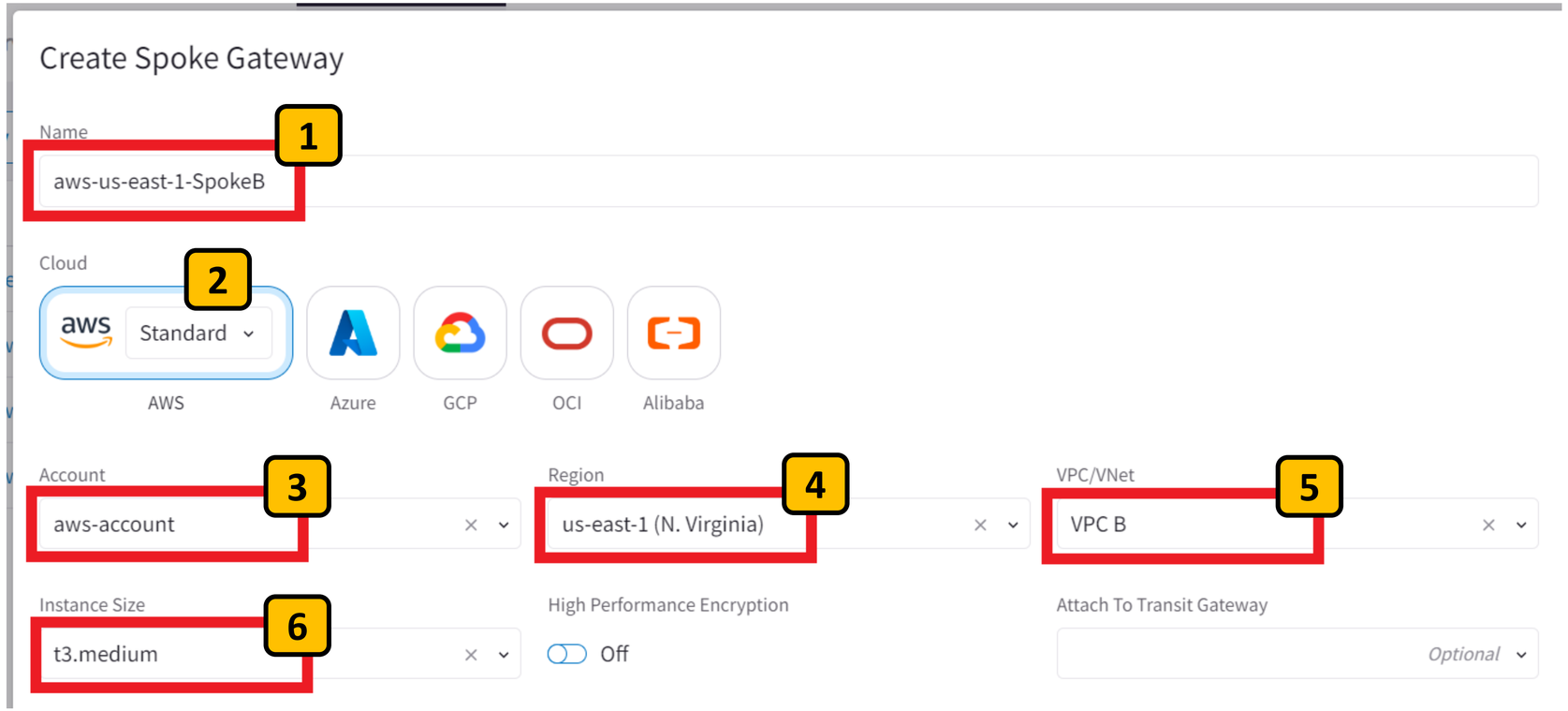
Select the **Spoke Gateways** search result 2

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



Lab 2: Step 2.40

Deploy Aviatrix Spoke Gateway in VPC B Public Subnet



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

Select the cloud AWS standard 2

Select the account **aws-account** 3

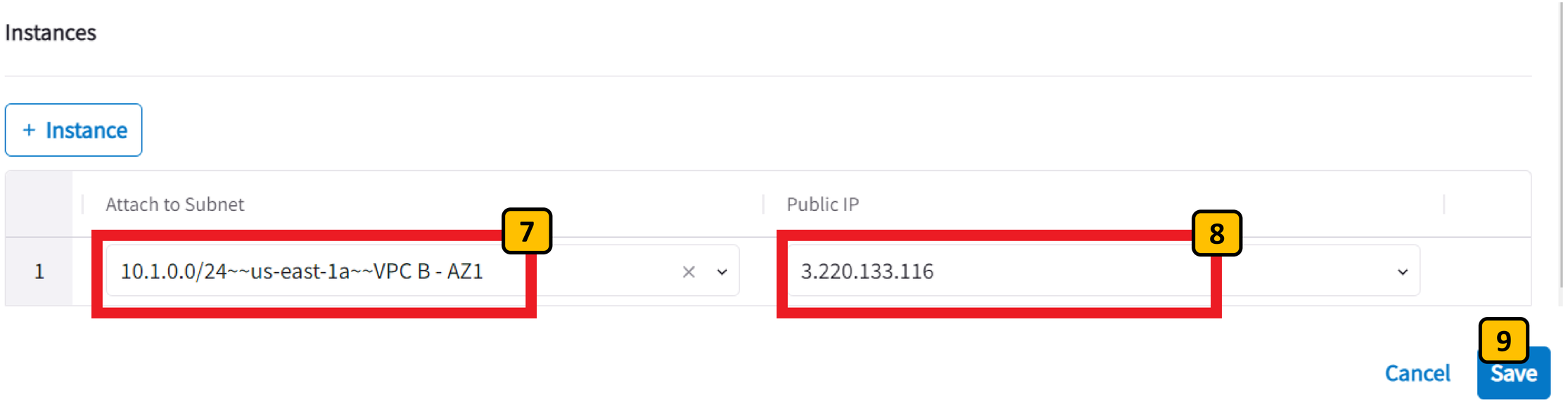
Select the region **us-east-1** 4

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



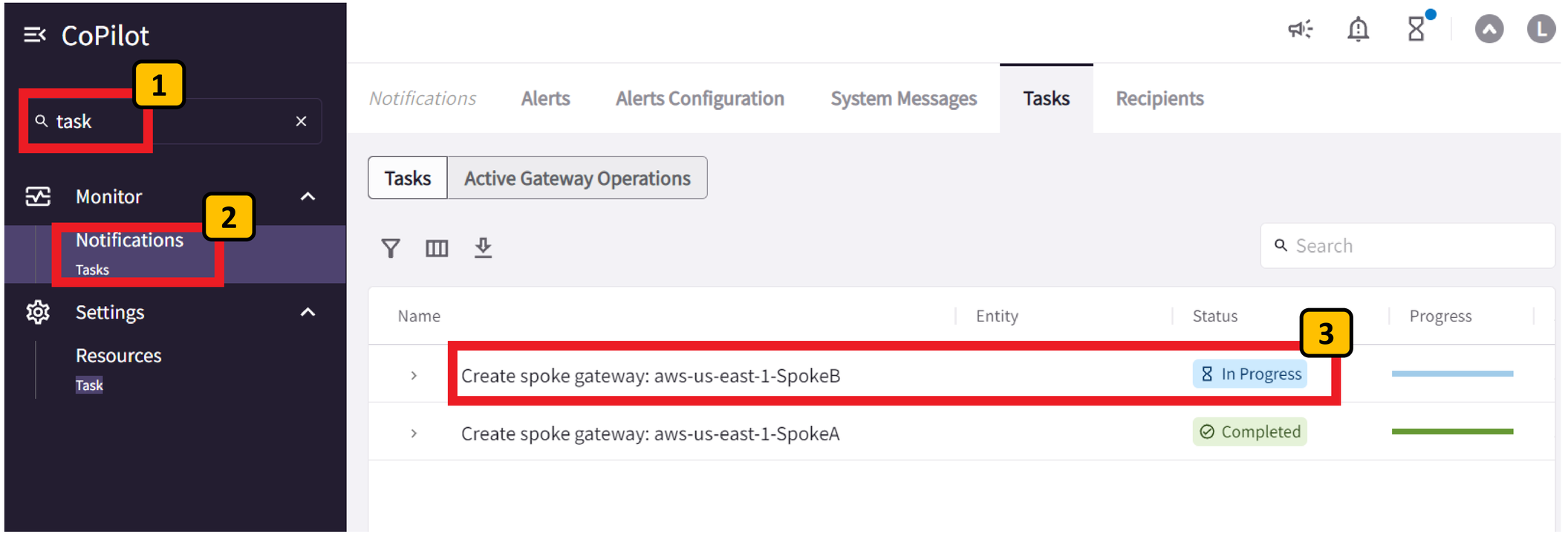
Select the 10.1.0.0/24 subnet in **AZ1** 7

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



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

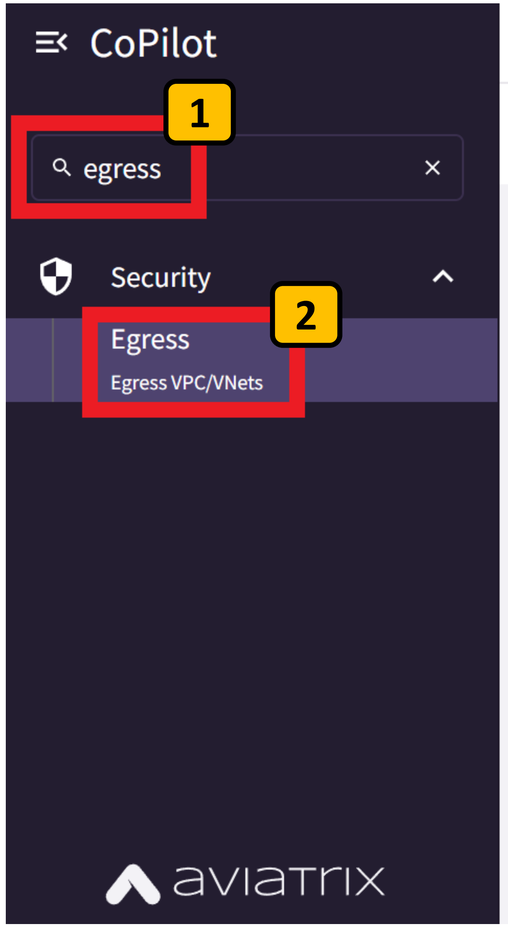
Click the search result **Notifications / Tasks** 2

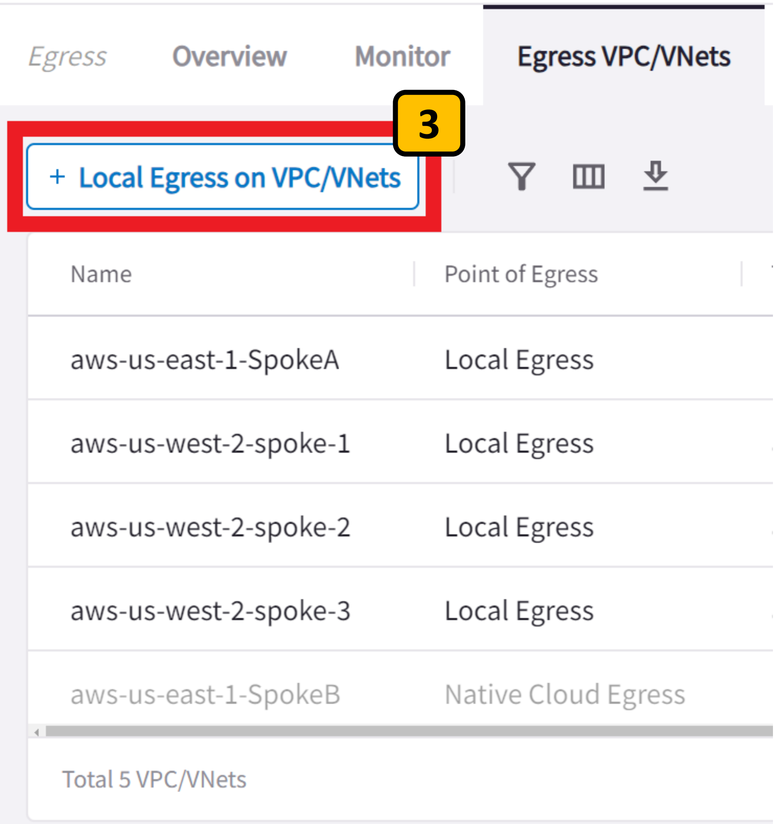
Observe the spoke gateway creation Task and wait for it to complete 3

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

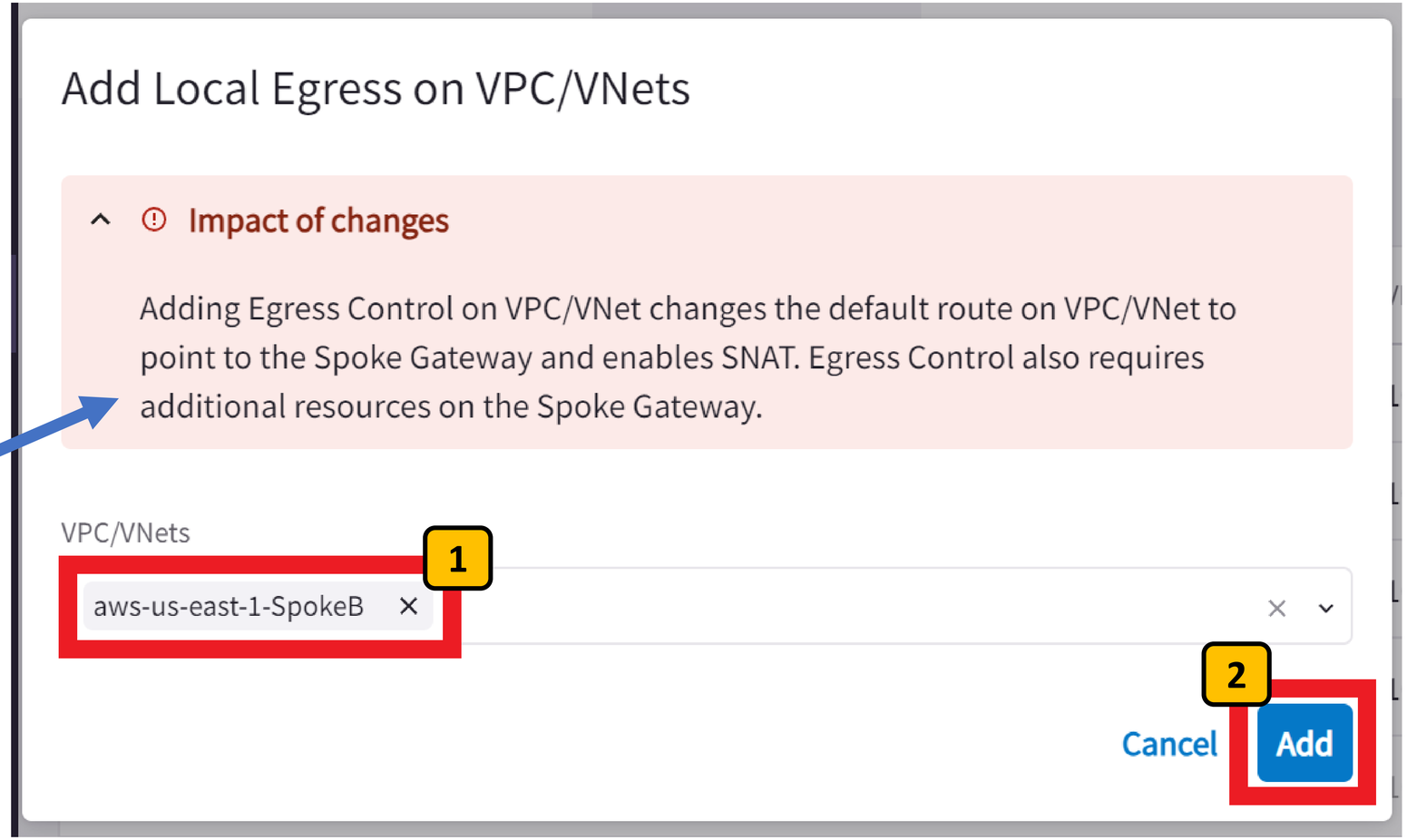
Lab 2: Step 2.44

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

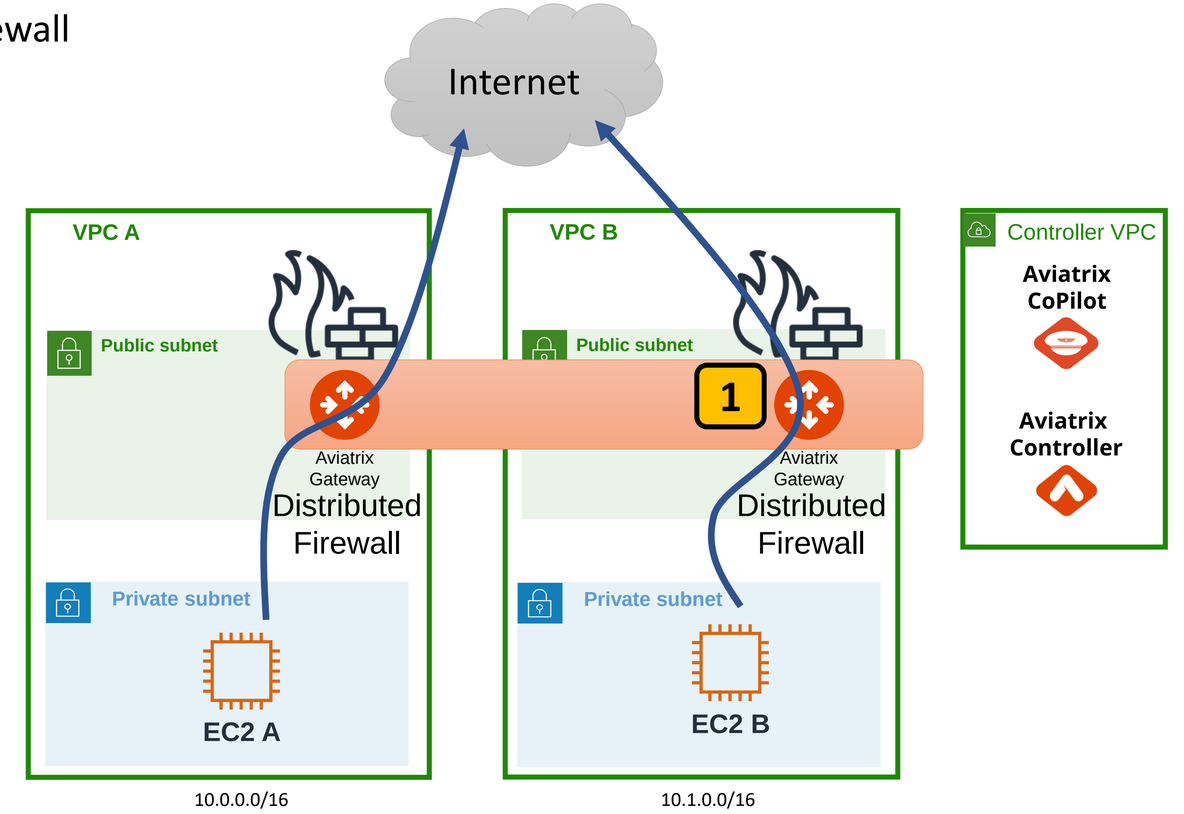
Aviatrix Distributed Firewall

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.



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

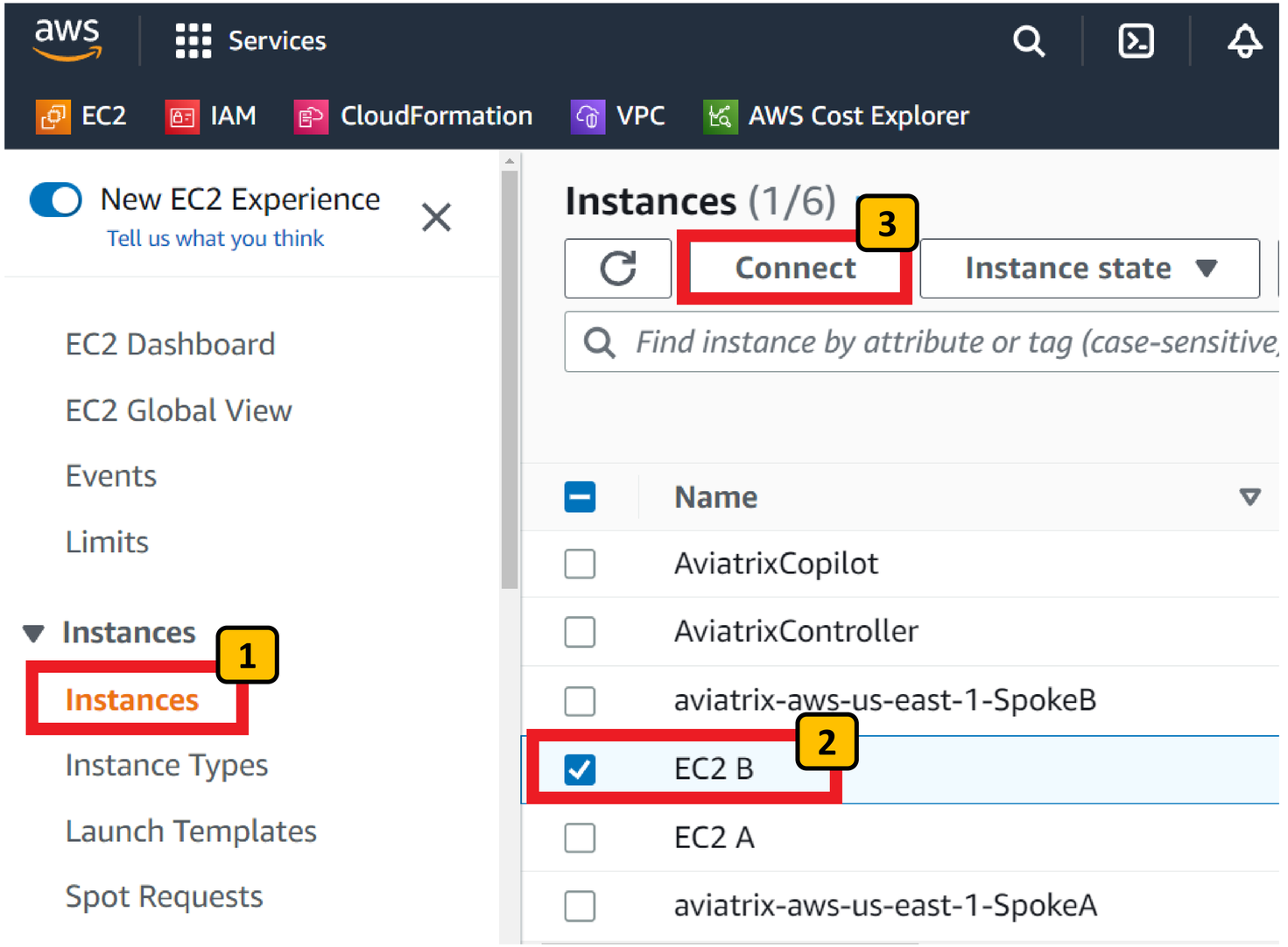
Connect to EC2 B instance console

Go the EC2 section of the AWS Console.

Select **Instances** 1

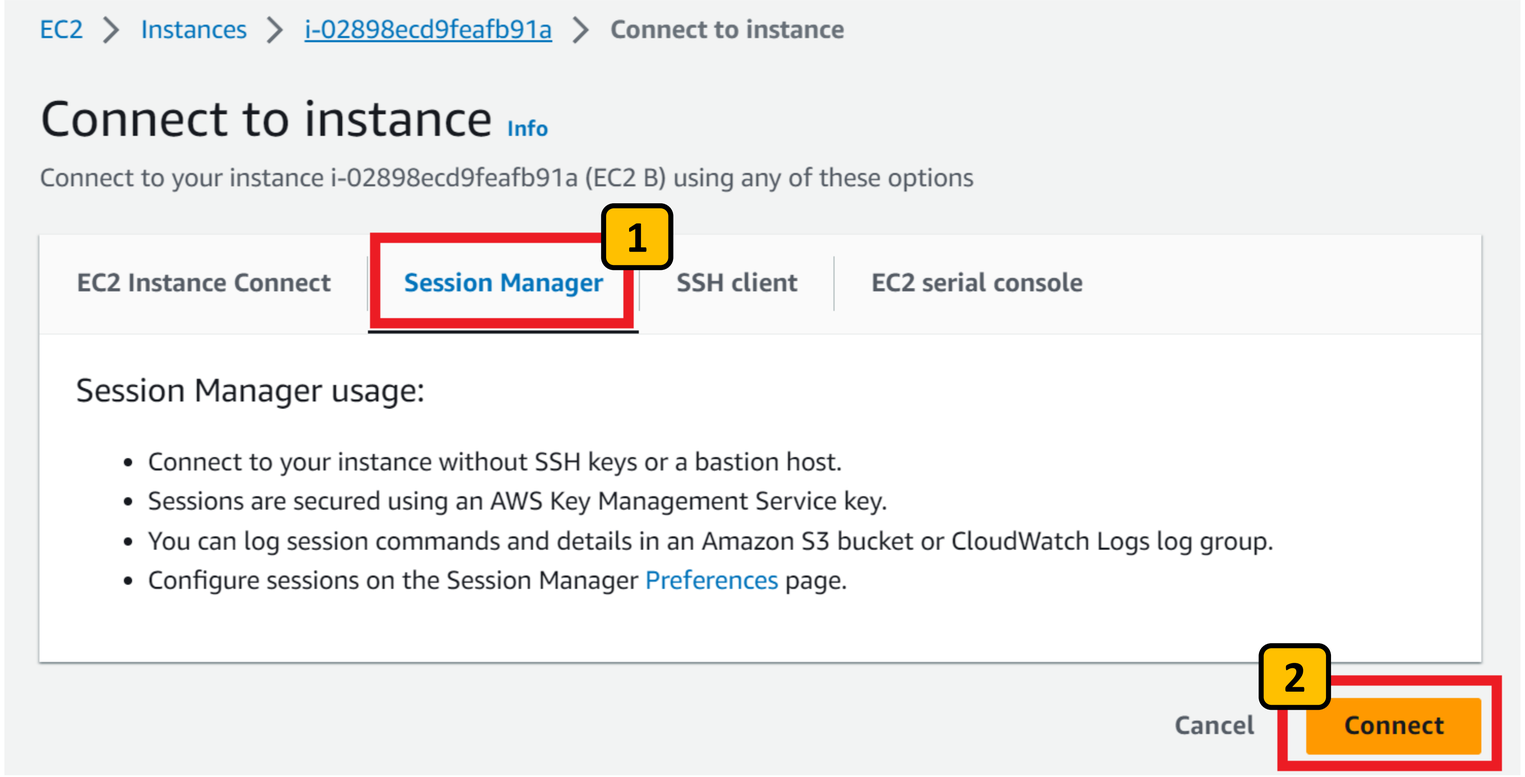
Find the **EC2 B** instance and select it. 2

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

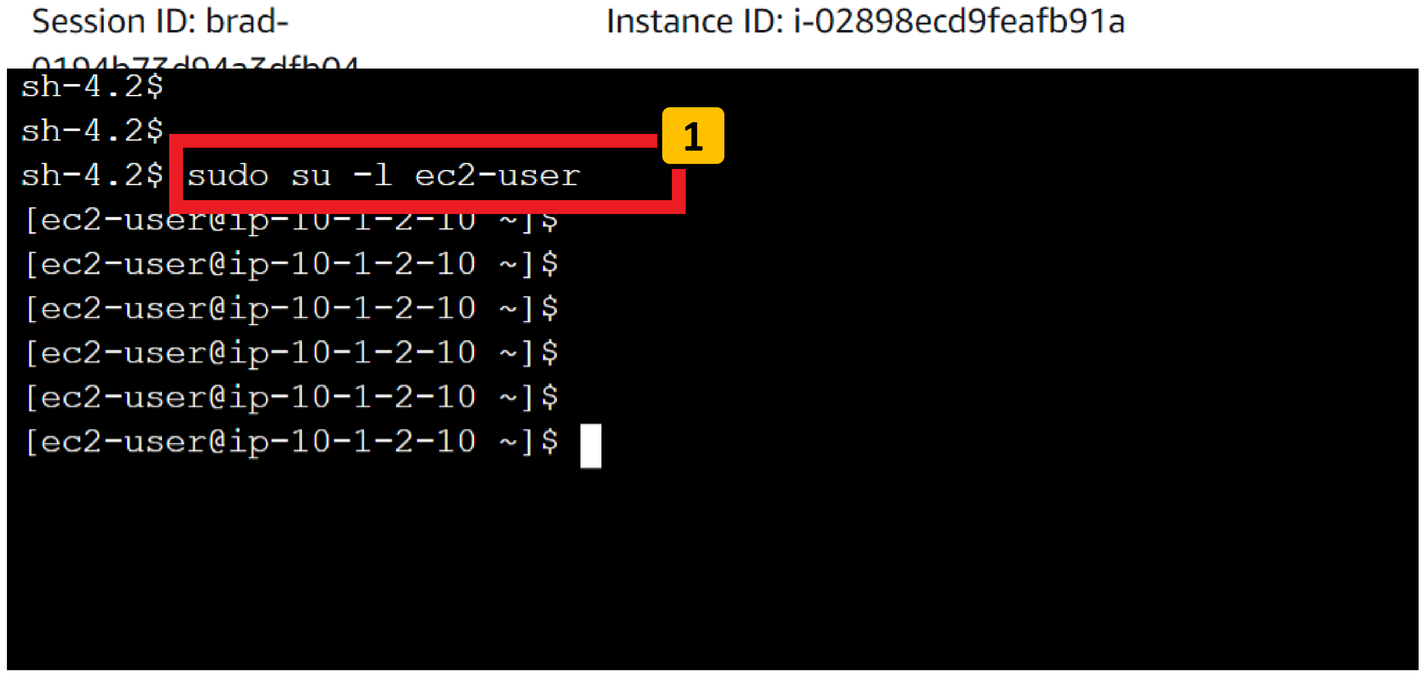
Note: If you get an error here, try rebooting EC2 B to accelerate the process of Session Manager reconnecting through the Aviatrix Gateway

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.

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

Lab 2: Step 2.49

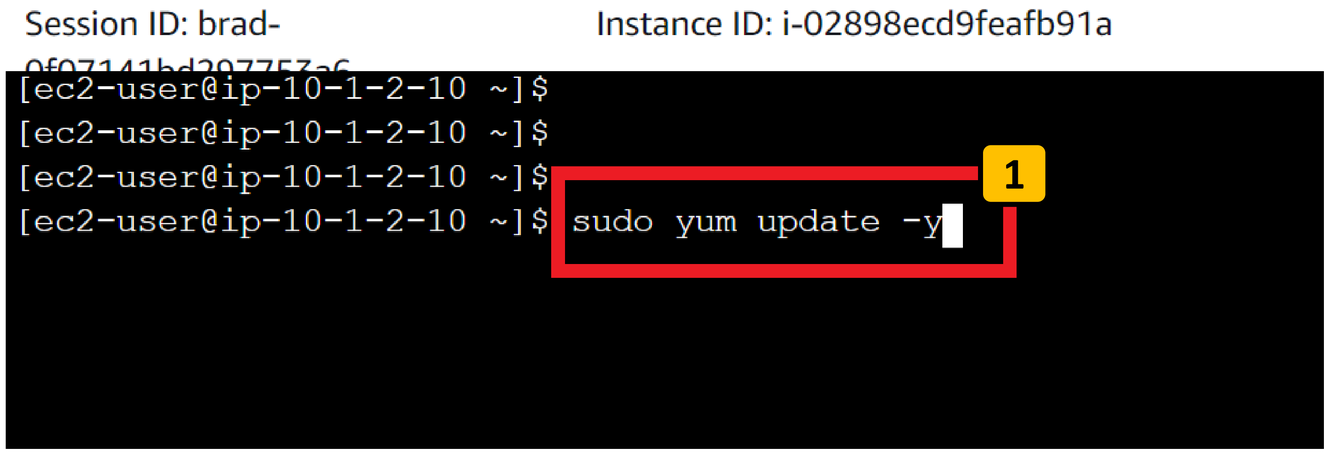
Run a software update

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.



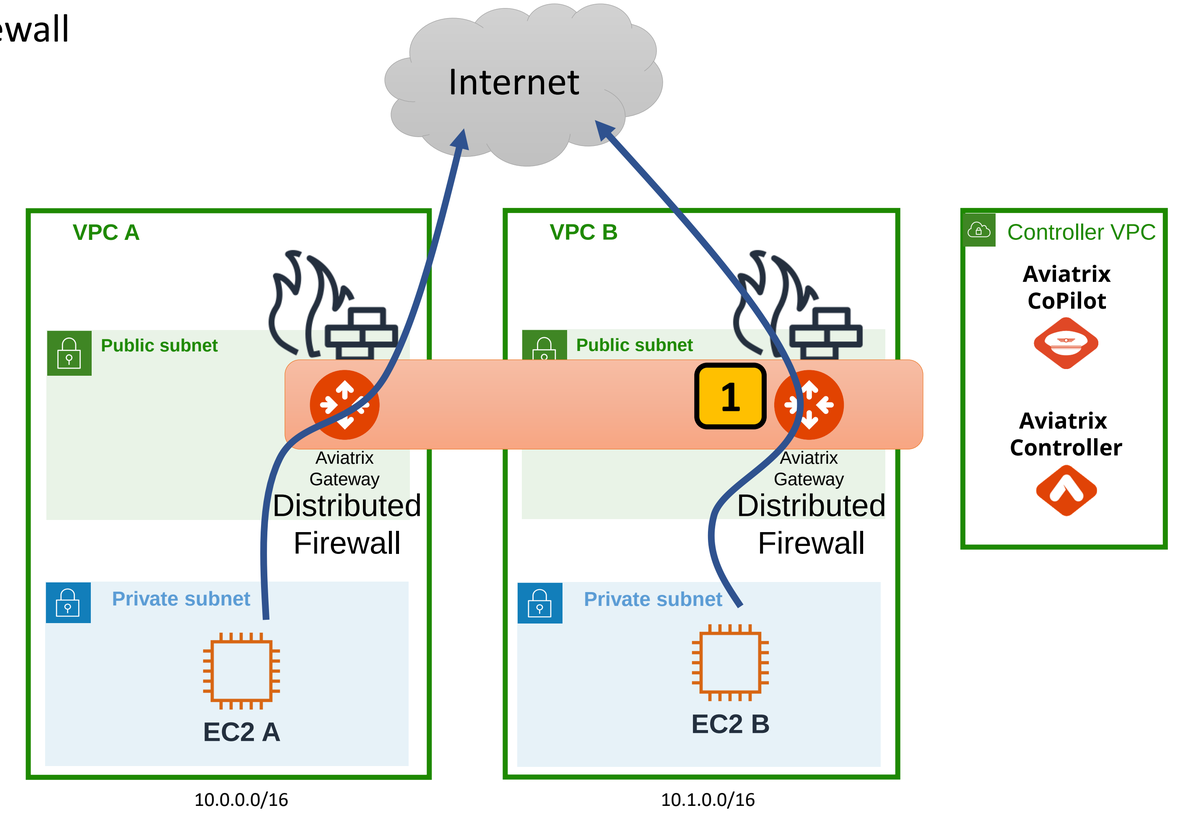
Lab 2: Checkpoint 6

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.

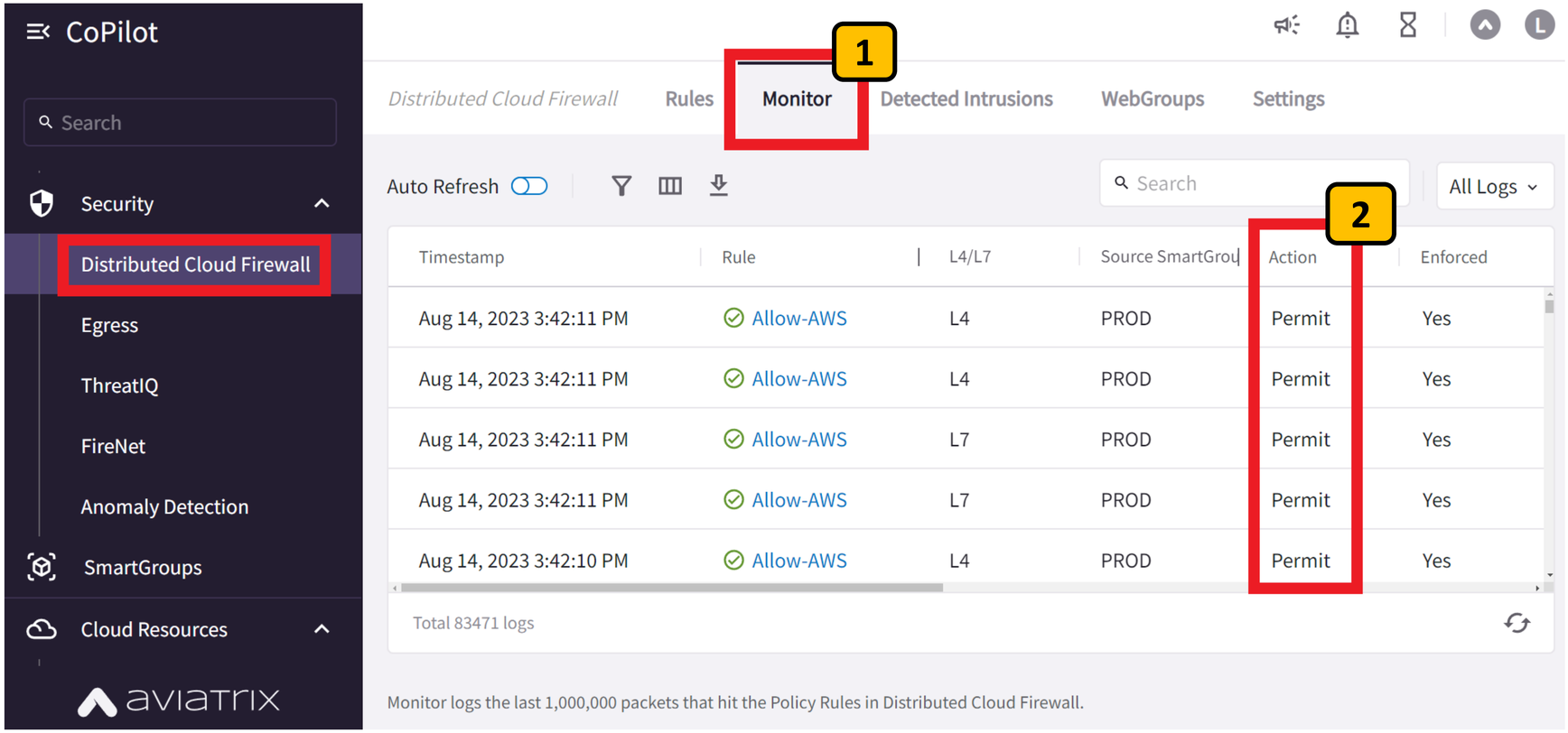
In addition to security, The Aviatrix Distributed Firewall also provides deep traffic visibility, monitoring, alerting, and scaling policies.



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

Lab 2: Step 2.50

See the Policy actions

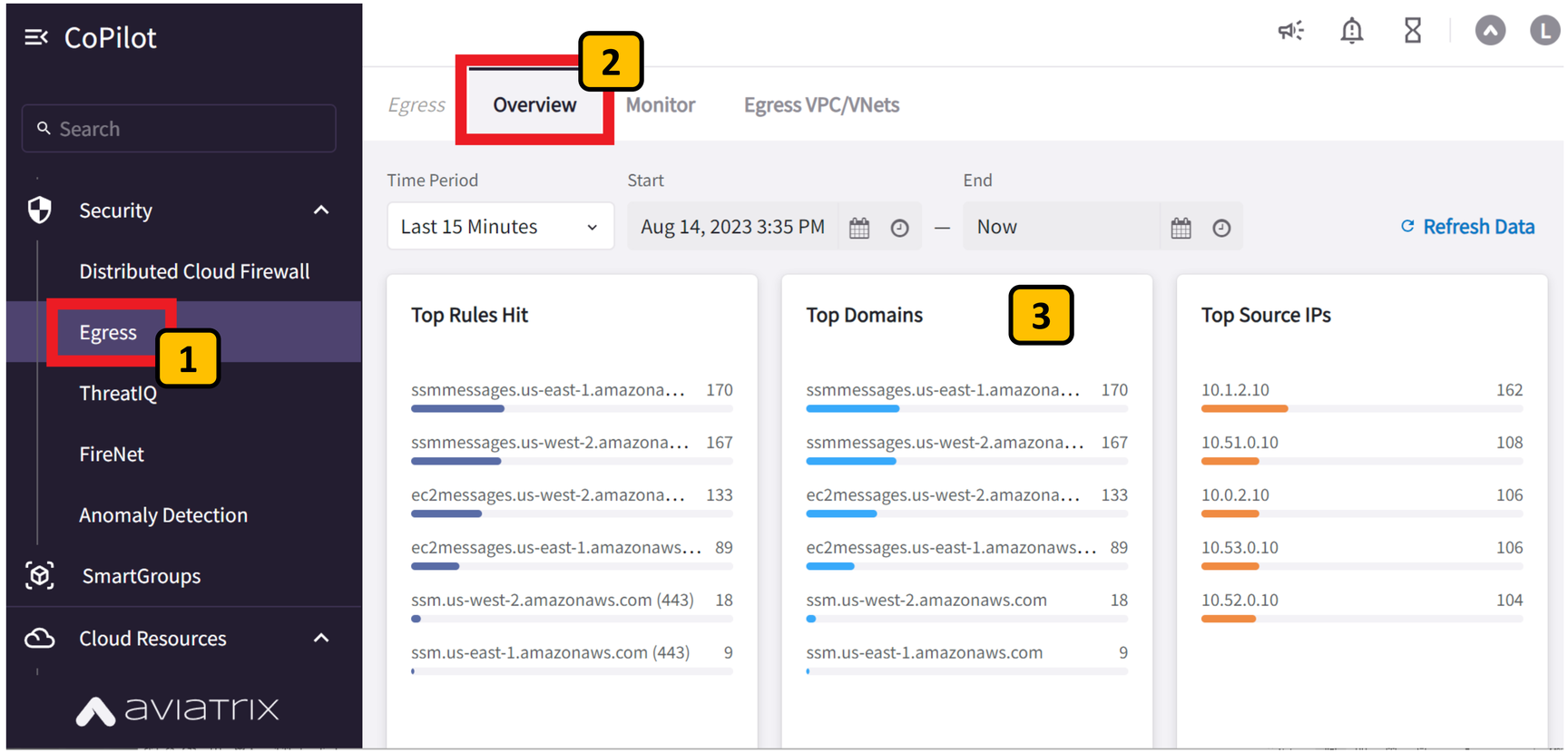


In the Distributed Cloud Firewall section of CoPilot select **Monitor** tab 1

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

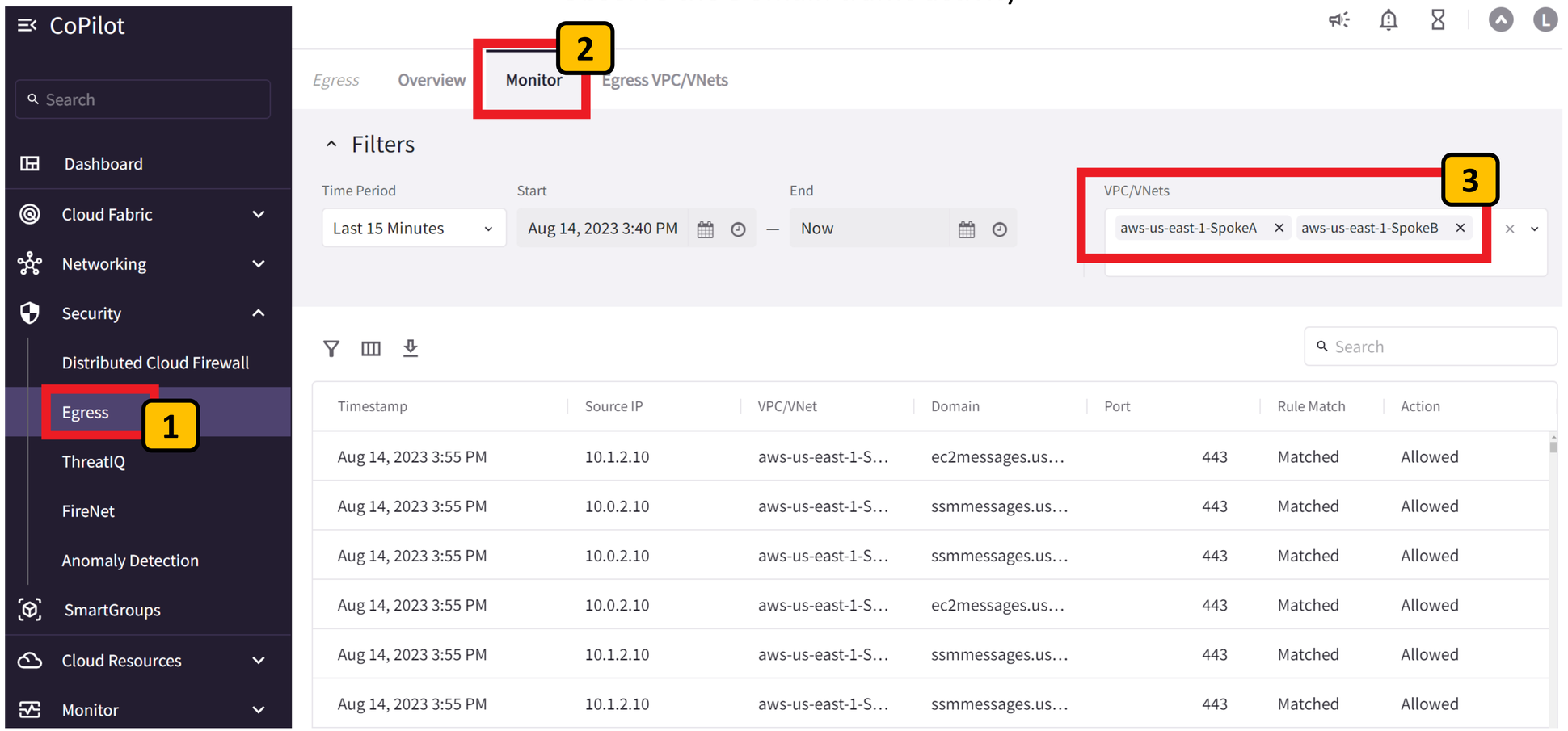
Select the **Overview** tab 2

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



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

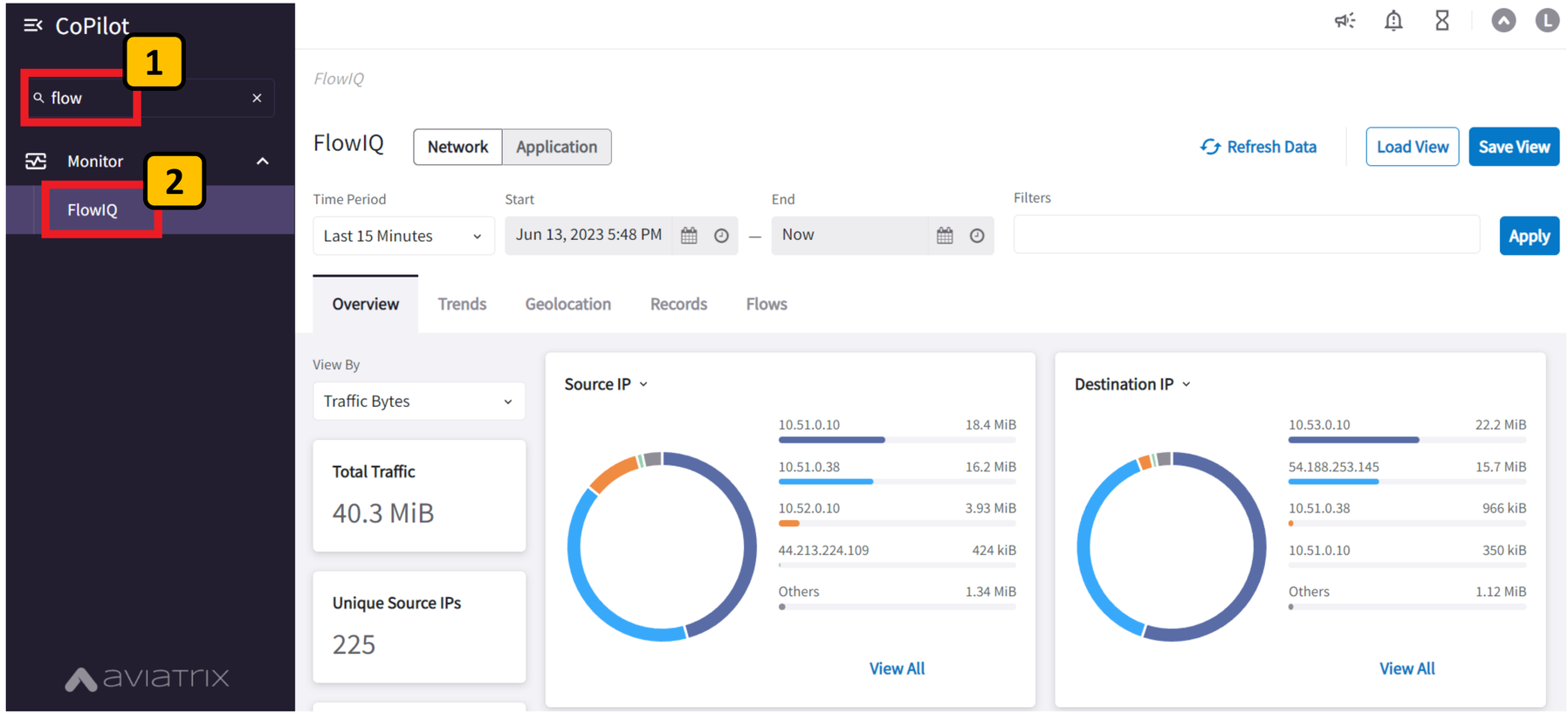
Select the **Monitor** tab 2

Select **SpokeA** and **SpokeB** gateways in the VPC/VNets drop-down 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



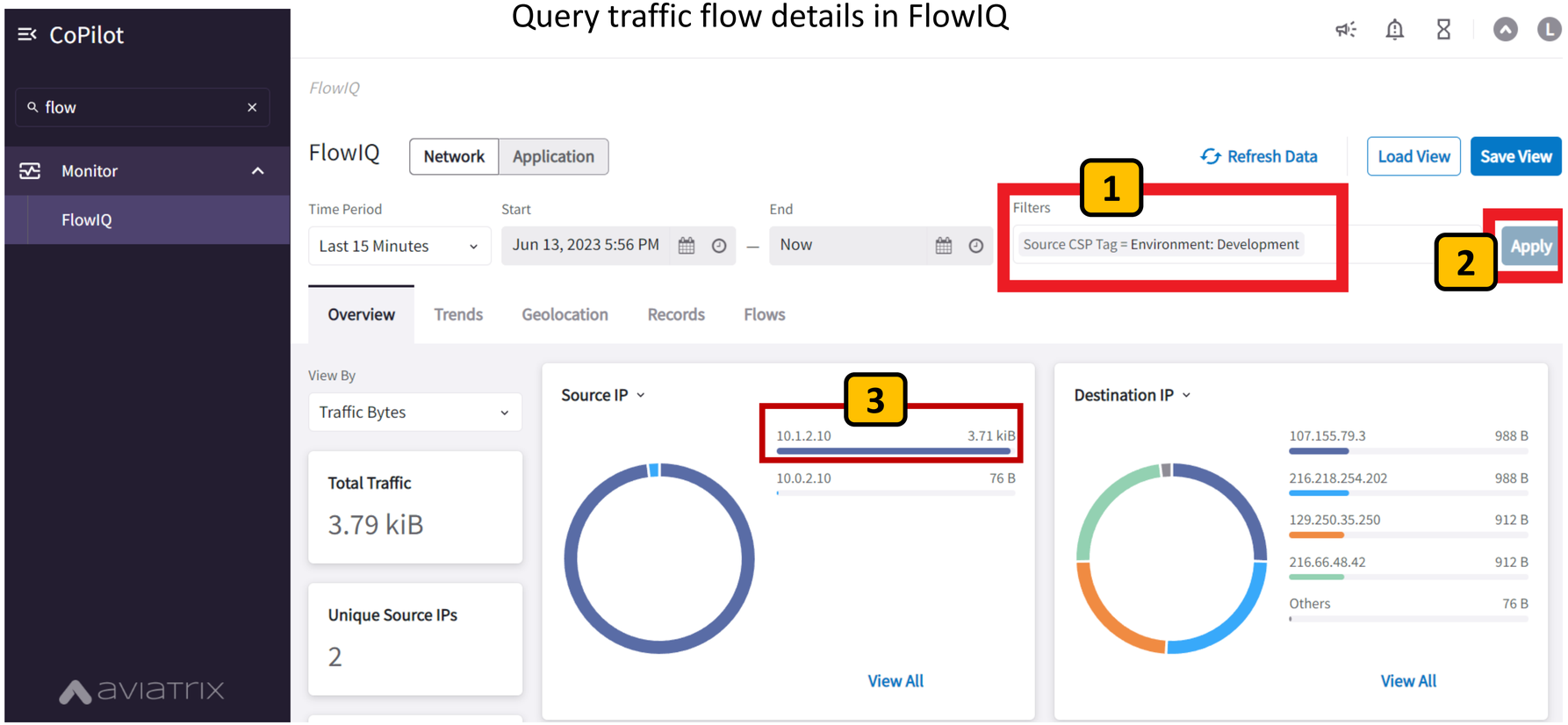
In the CoPilot search bar type **flow** 1

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

Query traffic flow details in FlowIQ



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

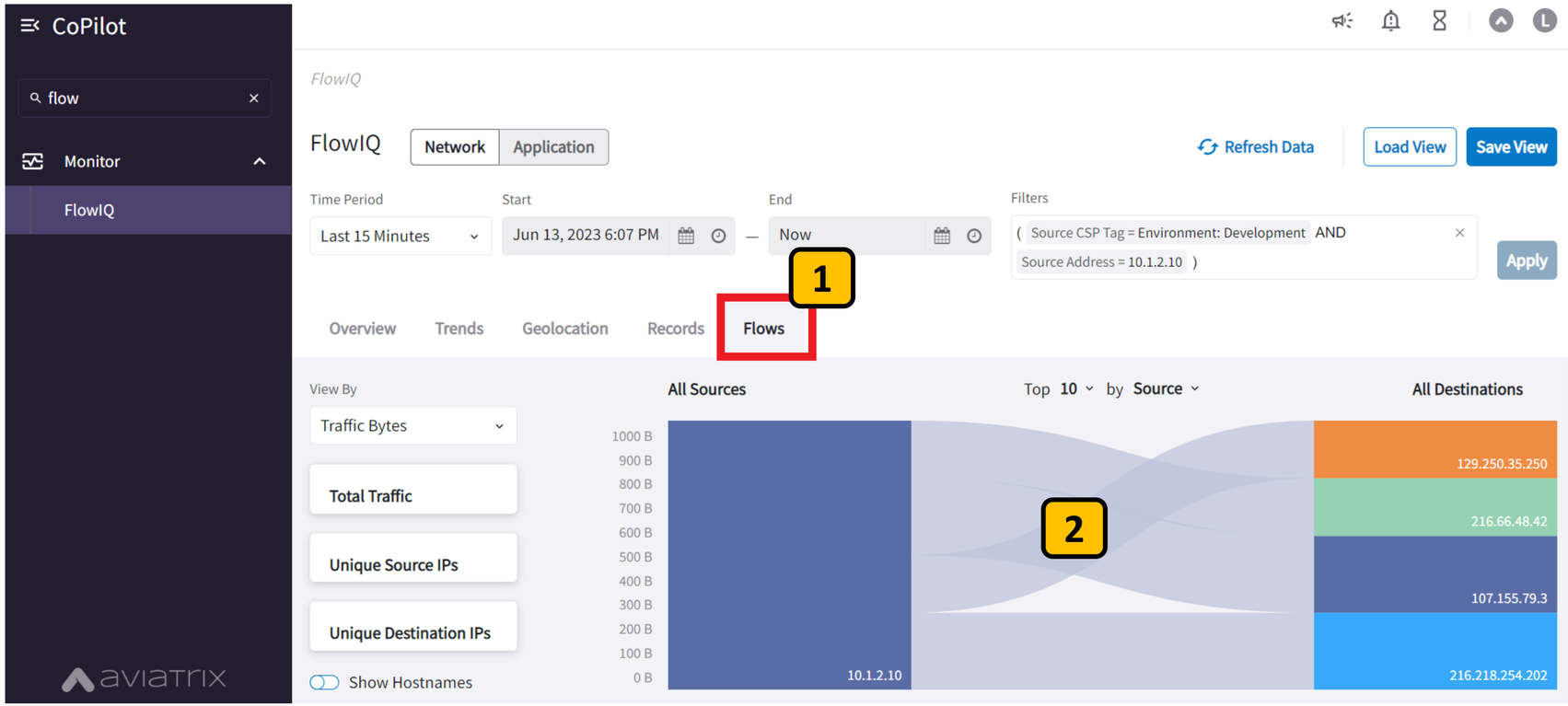
Click **Apply** to apply the filter 2

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

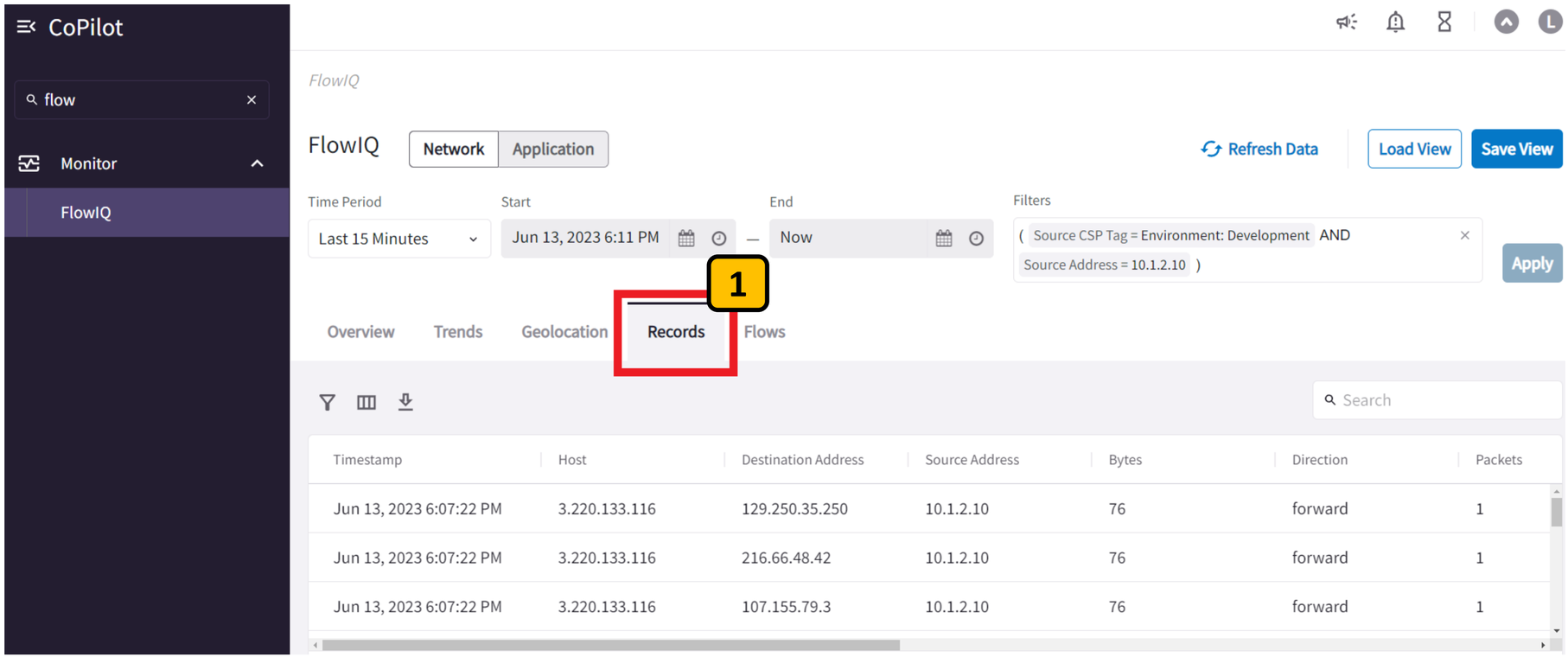


Select the Flows tab to see a flow summary from this top talker in Development 1

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

Lab 2: Step 2.54

Query traffic flow details in FlowIQ

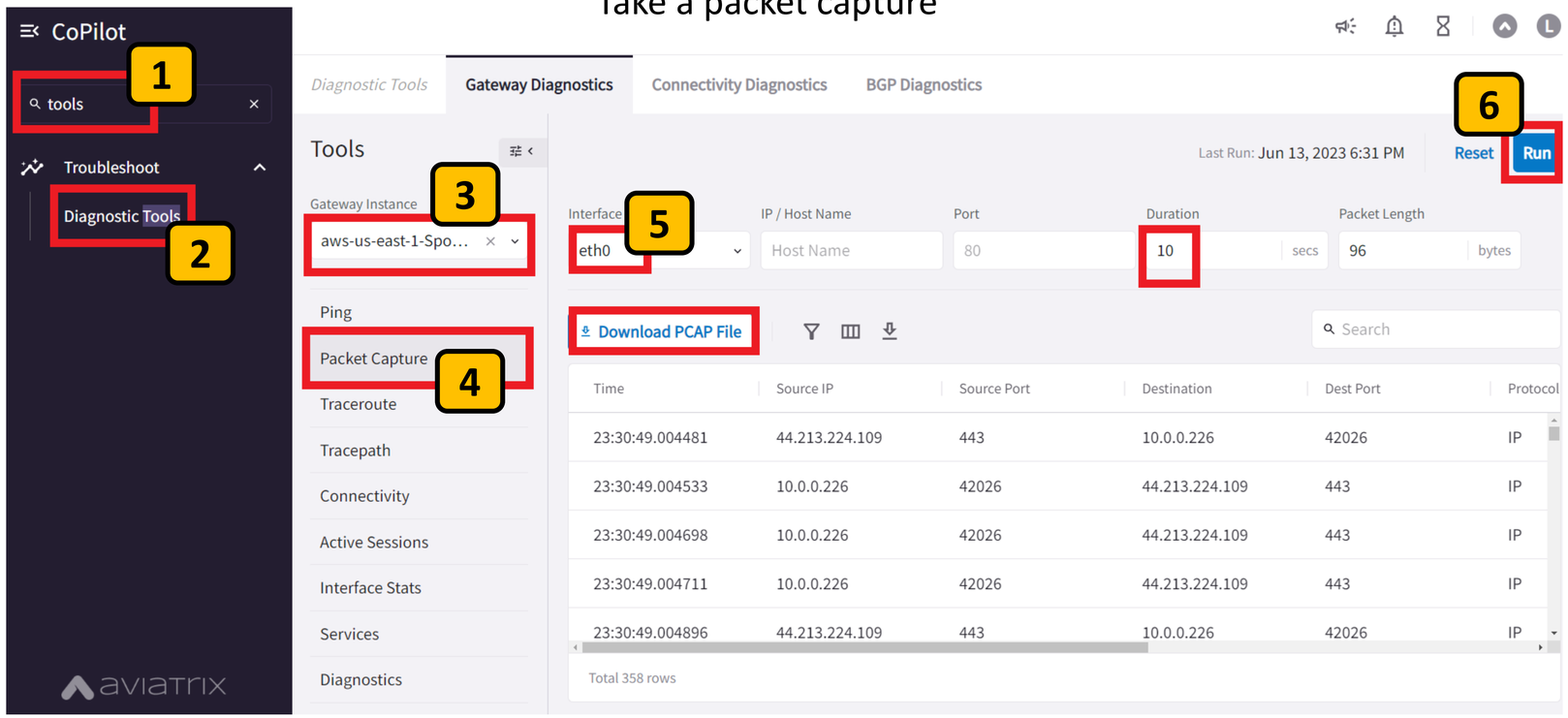


Select the **Records** tab to see details of each traffic flow record from this top talker in Development 1

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

Lab 2: Step 2.55

Take a packet capture



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

Select the **Diagnostic Tools** search result 2

Select a gateway to capture from 3

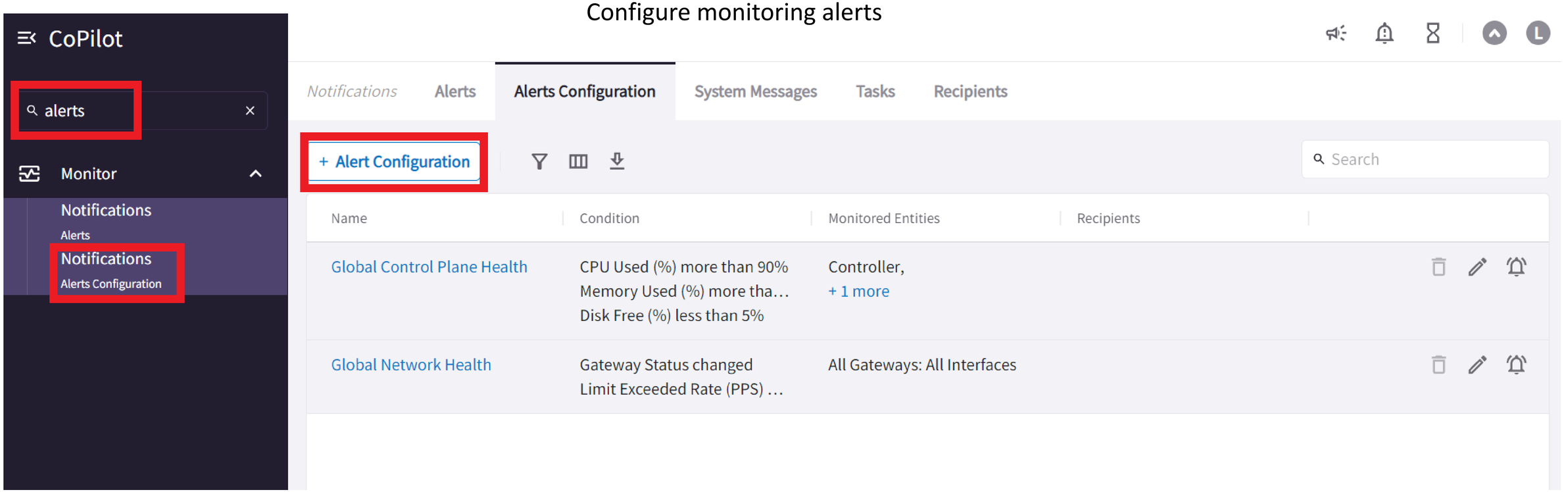
Select **Packet Capture** 4

Select the **eth0** interface 5

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

Lab 2: Step 2.56

Configure monitoring alerts



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

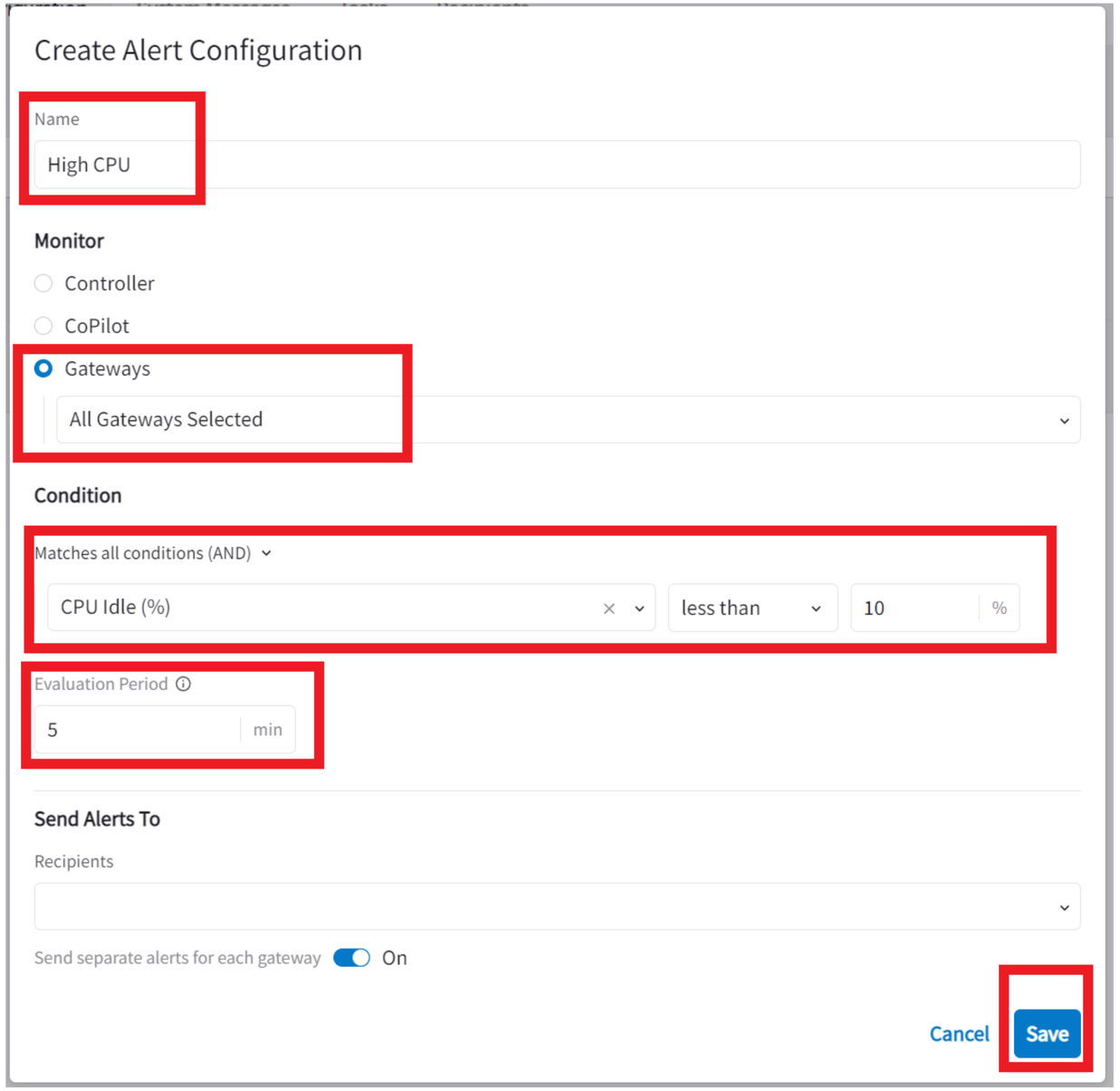
Name the Alert **High CPU** 1

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

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

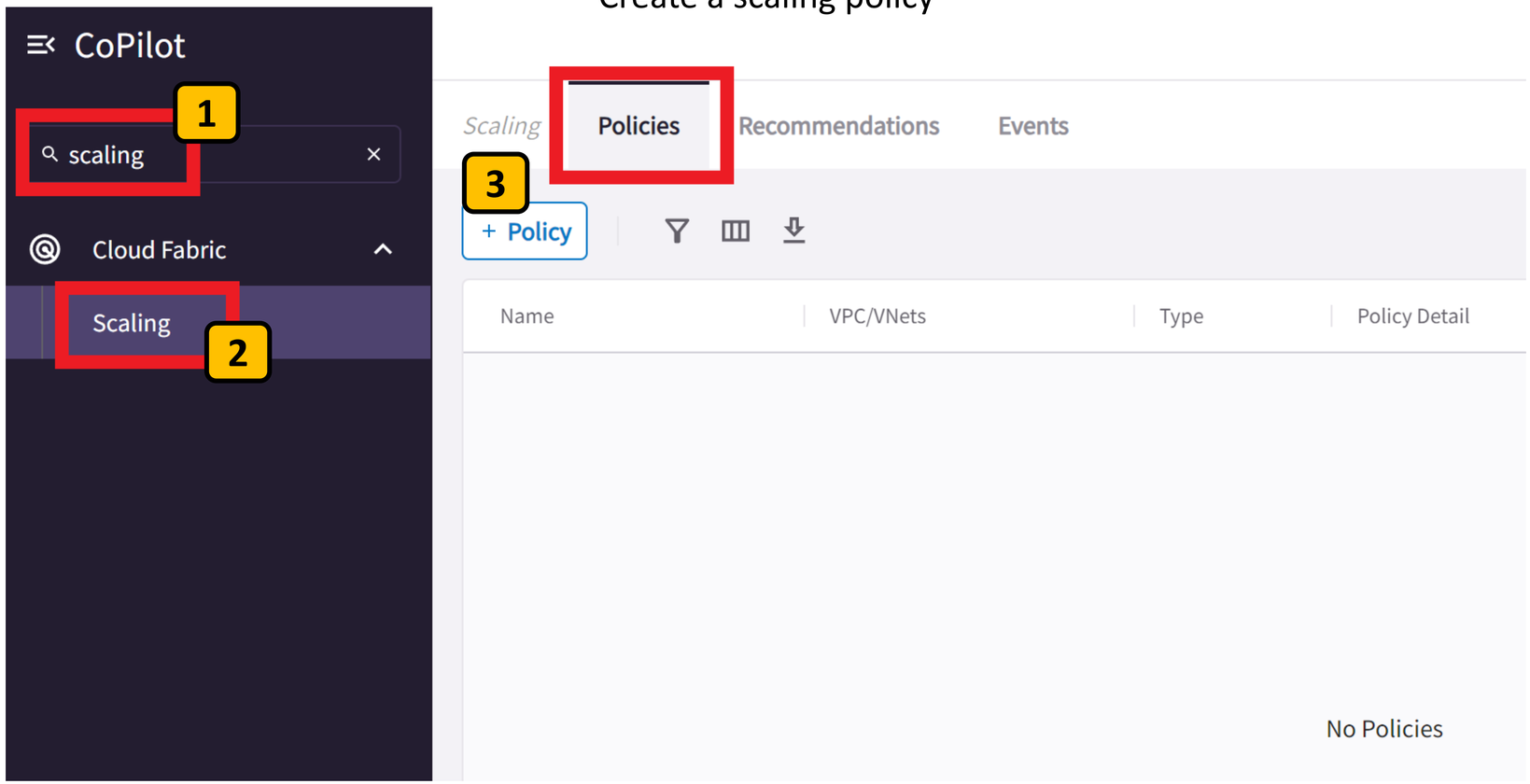
Set the Evaluation Period to 5 min 4

Click **Save** 5



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

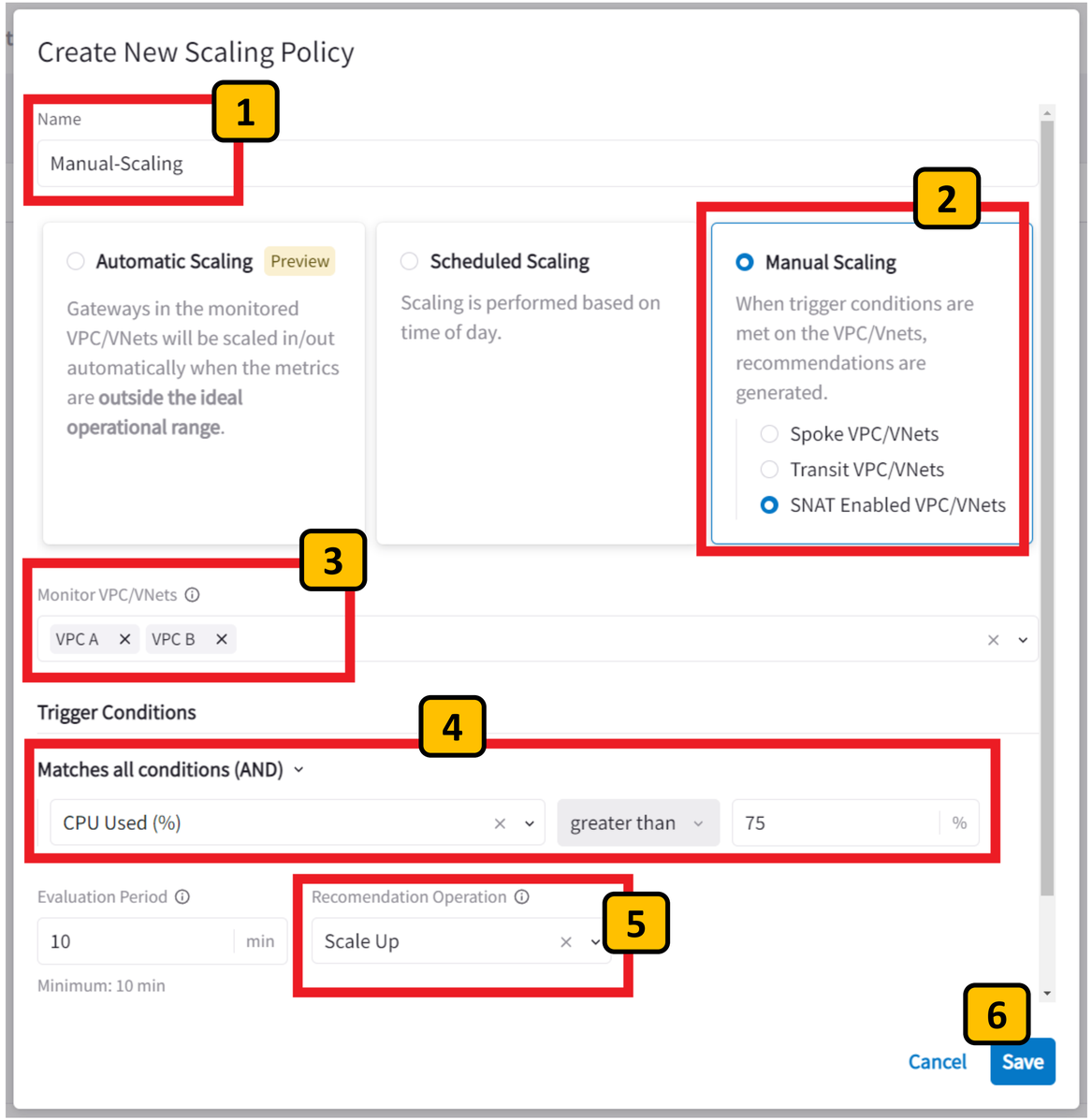
Select the **Manual Scaling** type and select **SNAT Enabled VPC** 2

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.