AWS Immersion Day
LAB 7

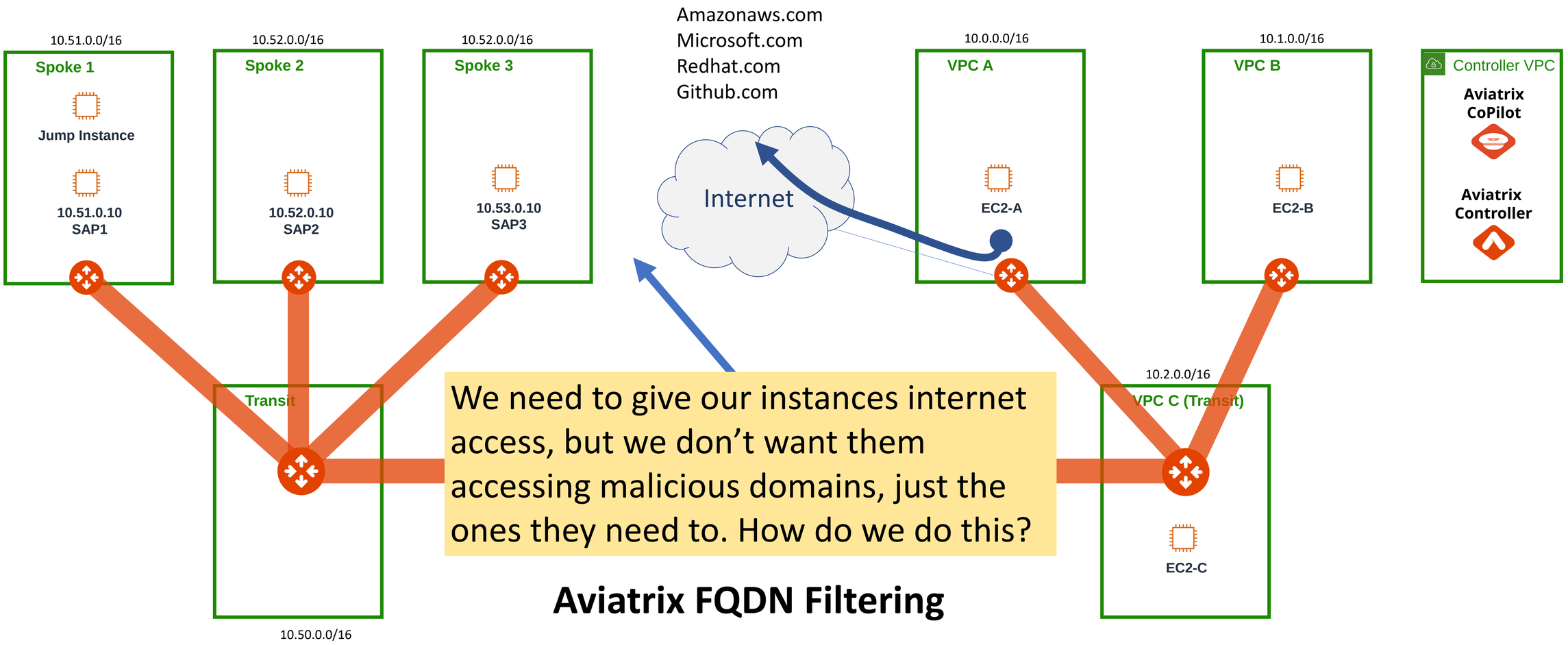
SECURITY: EGRESS FQDN FILTERING

Brad Hedlund
Principal Solutions Architect,
Aviatrix Systems

aviatrix

Lab 7 Intro pt.1

Egress FQDN Filtering



We need to give our instances internet access, but we don't want them accessing malicious domains, just the ones they need to. How do we do this?

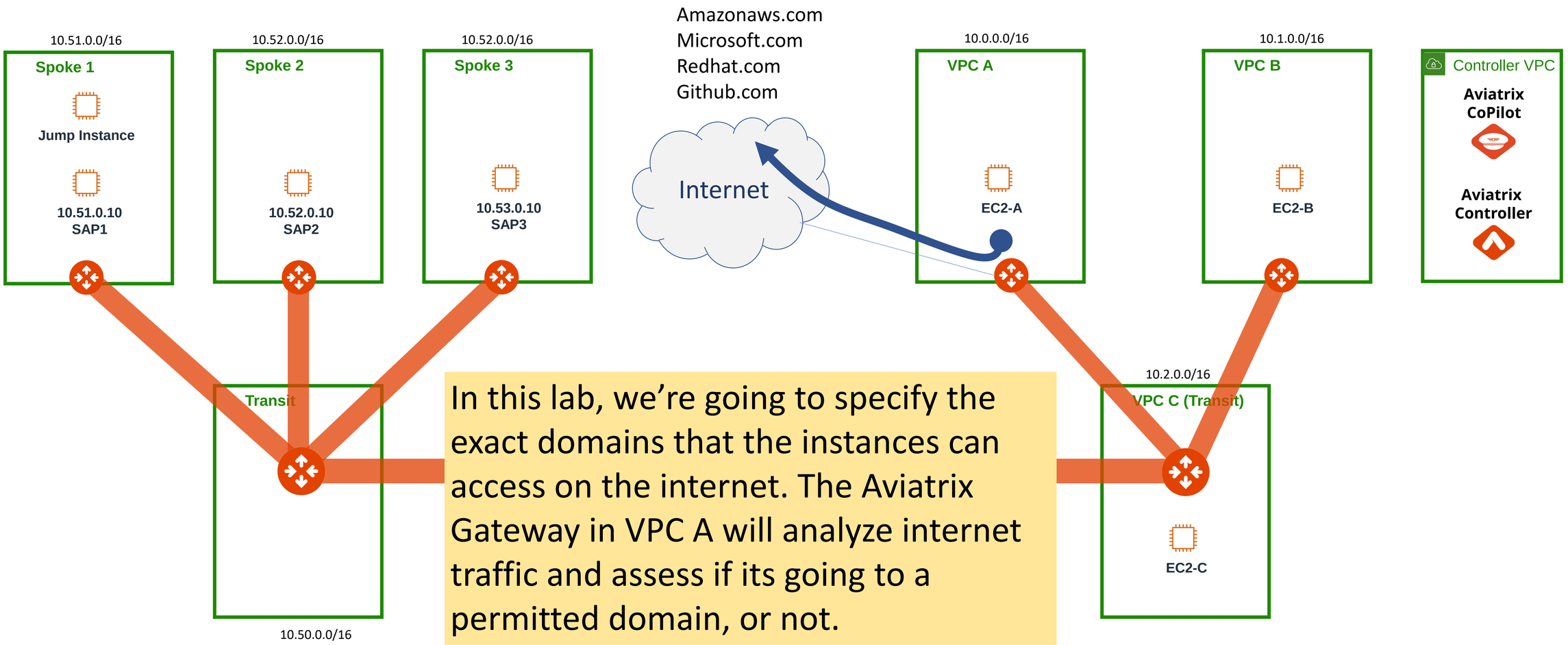
Aviatrix FQDN Filtering

AWS us-west-2

AWS us-east-1

Lab 7 Intro pt.2

Egress FQDN Filtering



In this lab, we're going to specify the exact domains that the instances can access on the internet. The Aviatrix Gateway in VPC A will analyze internet traffic and assess if its going to a permitted domain, or not.

AWS us-west-2 **Aviatrix FQDN Filtering** **AWS us-east-1**

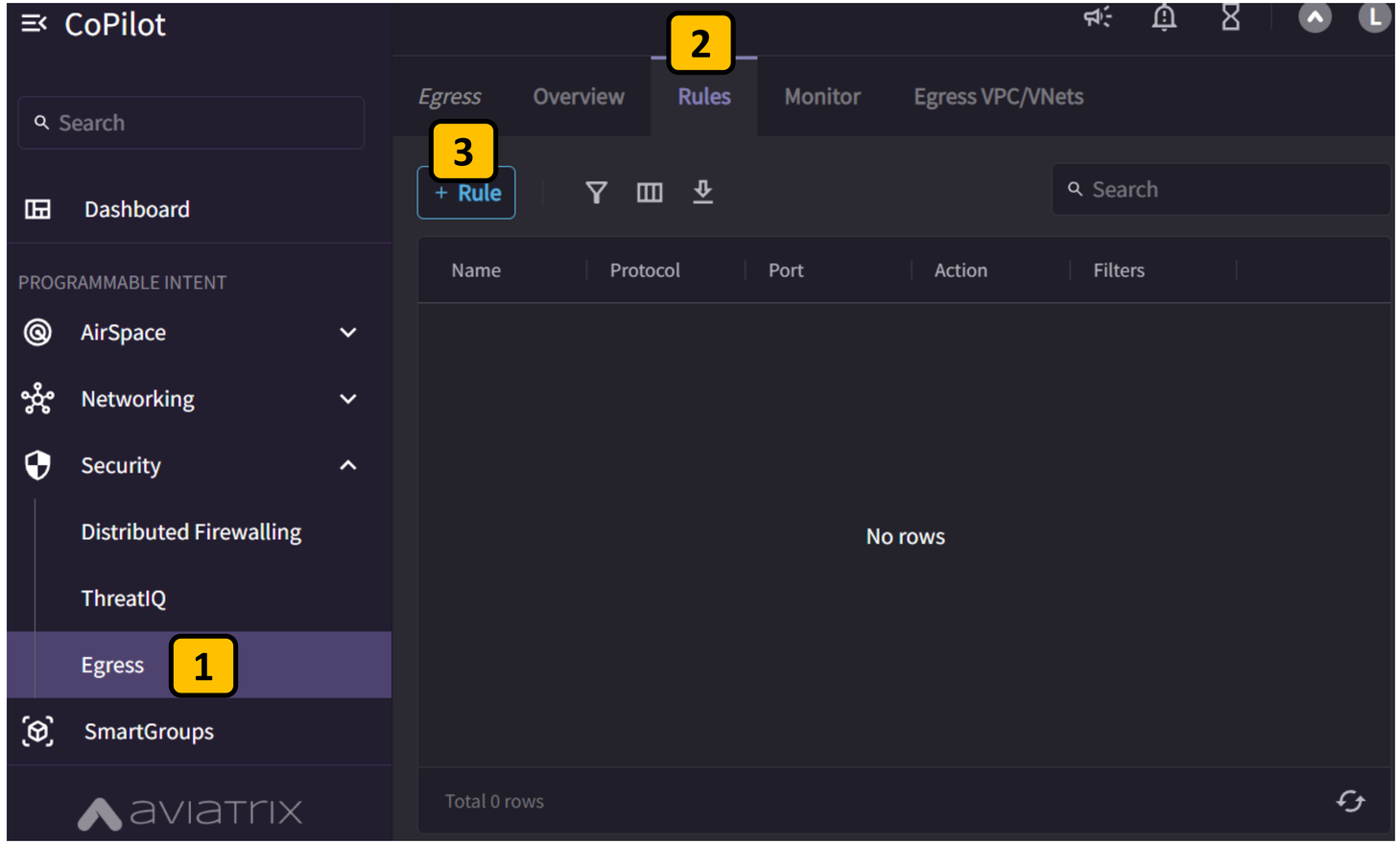
Lab 7: Egress FQDN Filtering: Step 7.0

Create domain filtering rules

To begin, navigate to Security > Egress 1

Select Rules 2

Select +Rule 3



Lab 7: Egress FQDN Filtering: Step 7.1

Create domain filtering rules

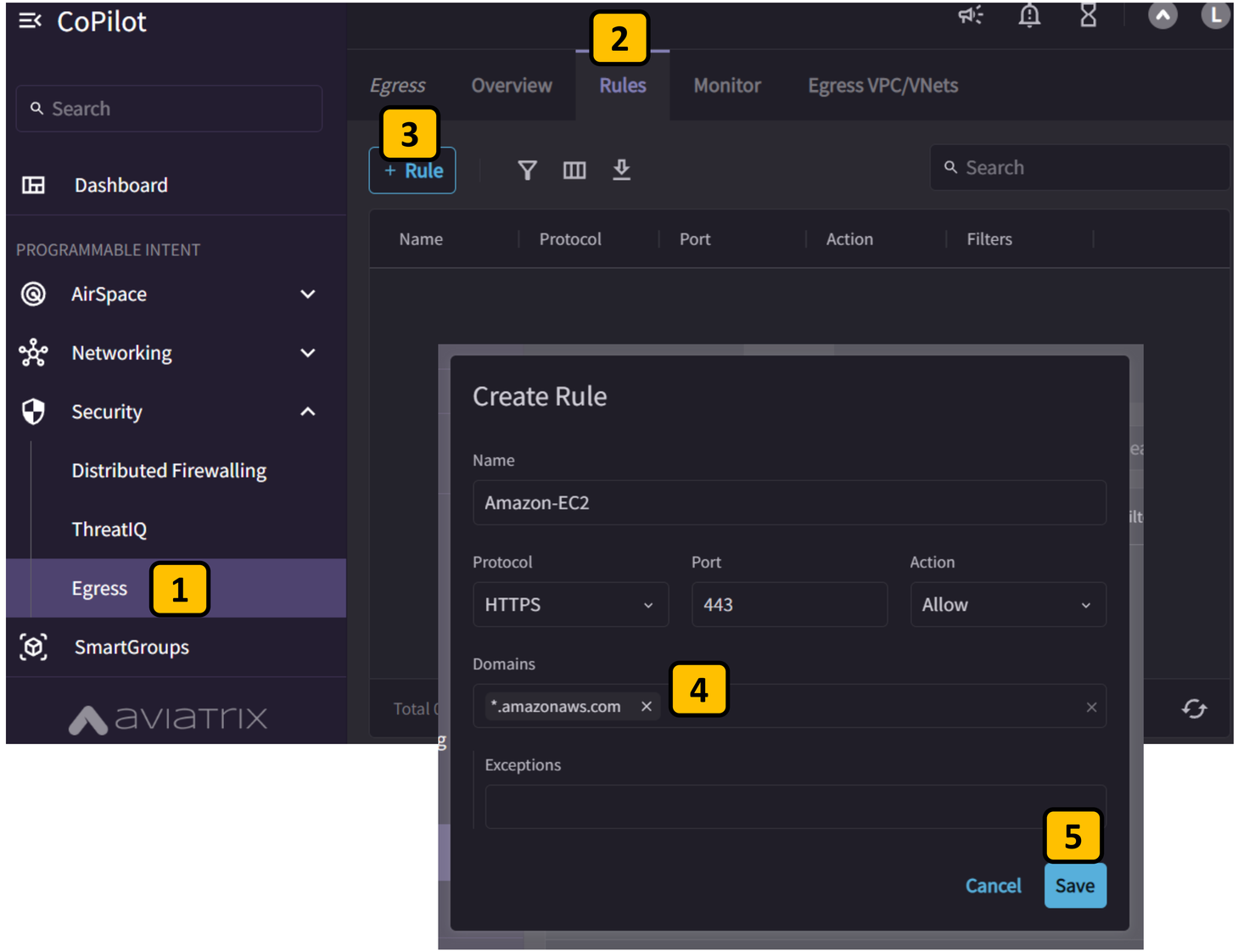
To begin, navigate to Security > Egress 1

Select Rules 2

Select +Rule 3

Create a rule that allows HTTPS to the domain ***.amazonaws.com** 4

Click Save 5



Lab 7: Egress FQDN Filtering: Step 7.2

Create domain filtering rules

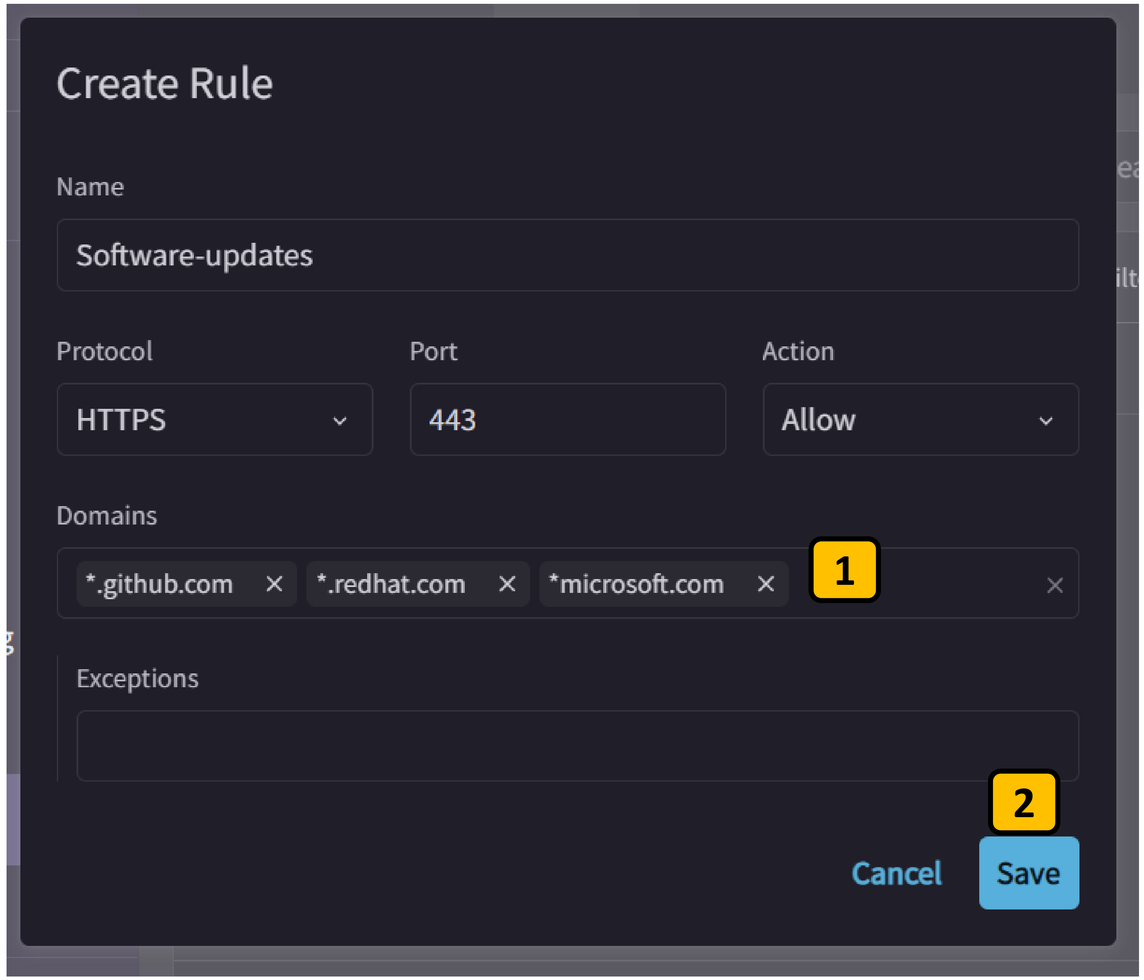
Create additional rules that allow the domains for HTTPS:

***.gitub.com**

***.redhat.com** 1

***.microsoft.com**

Click Save 2



Lab 7: Egress FQDN Filtering: Step 7.3

Enable Secure Egress on a spoke Gateway

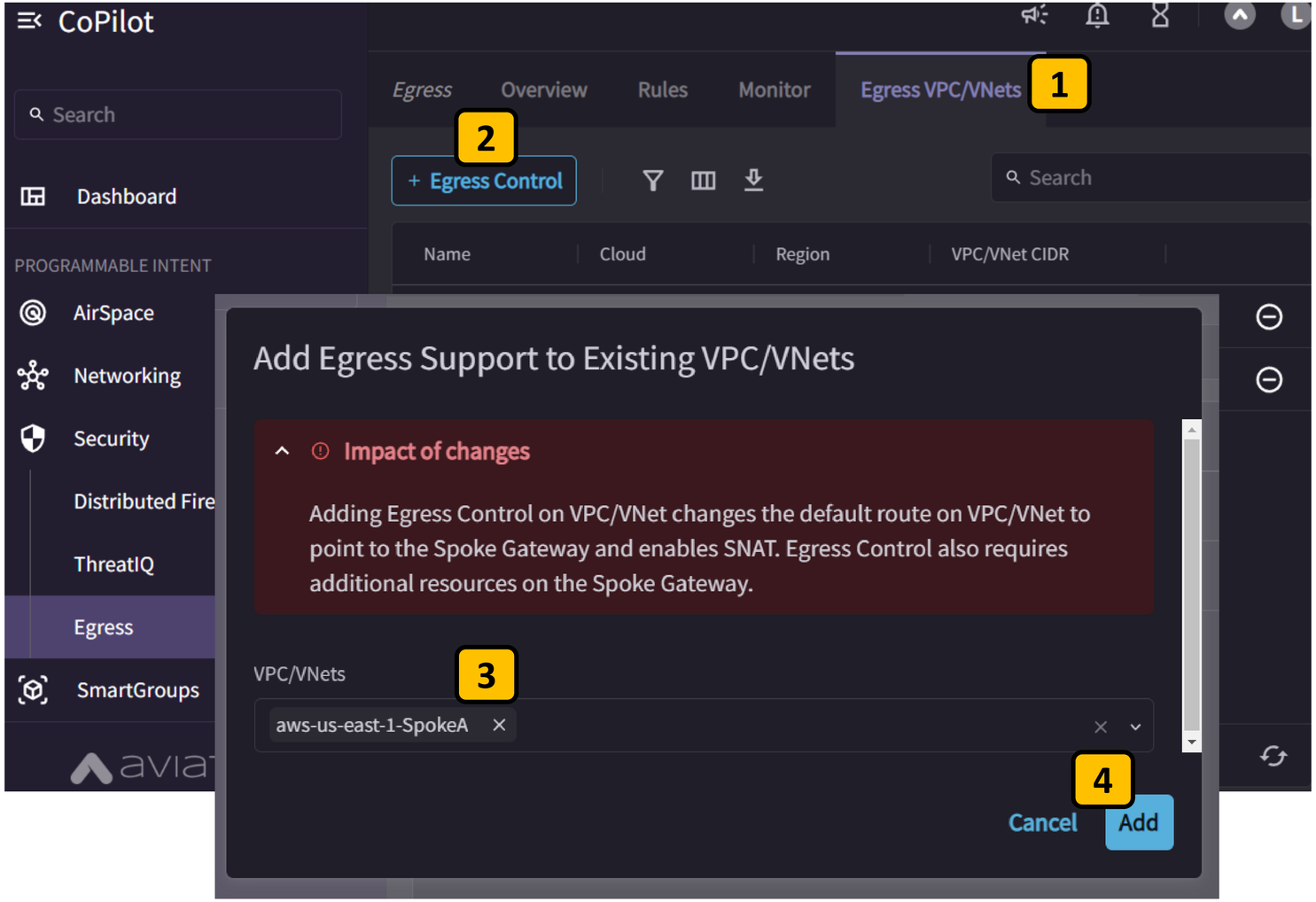
Now let's add Egress Control for the **aws-us-east-1-SpokeA** gateway

Select Egress VPC/VNets 1

Select +Egress Control 2

In the popup, select the gateway **aws-us-east-1-SpokeA** 3

Click Add 4



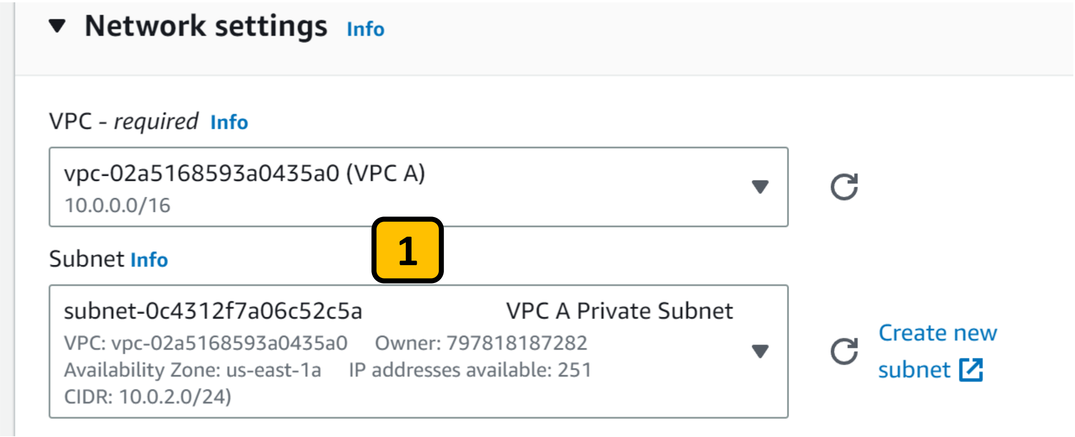
Lab 7: Egress FQDN Filtering: Step 7.4

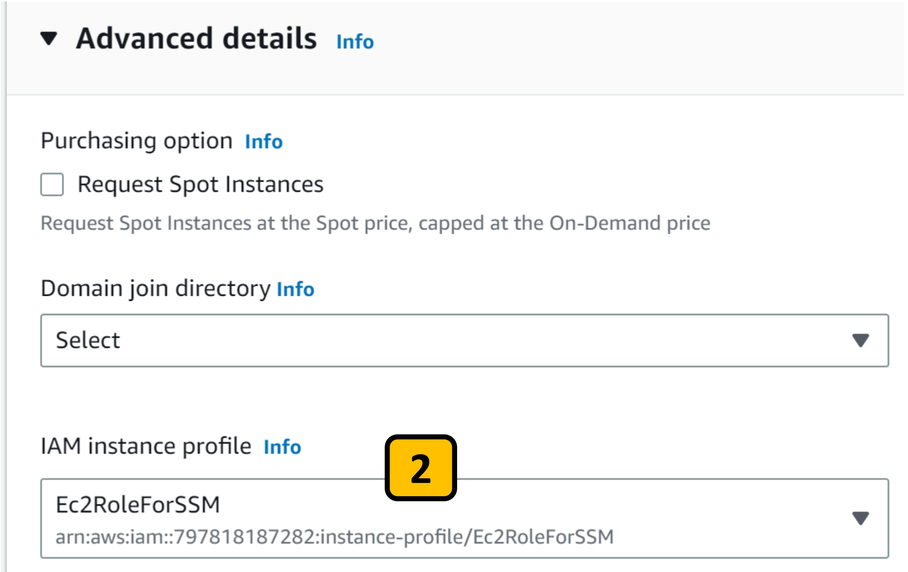
Launch instance in private subnet on VPC A

Now let's launch a new instance in a private subnet in VPC A and test our egress rules we just created.

Launch an Amazon Linux t2.micro instance in **us-east-1**, in **VPC A**, in the **VPC A Private Subnet** 1

Under Advanced Details, assign it to the IAM instance profile **Ec2RoleForSSM** 2



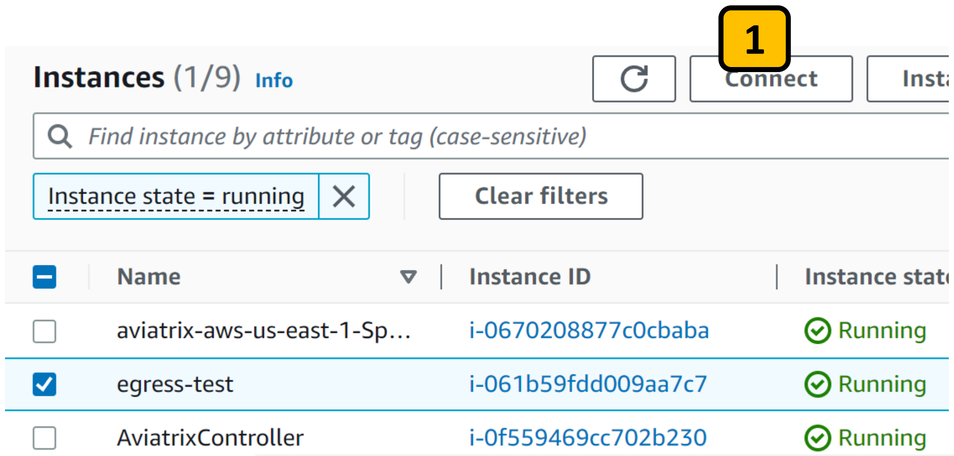


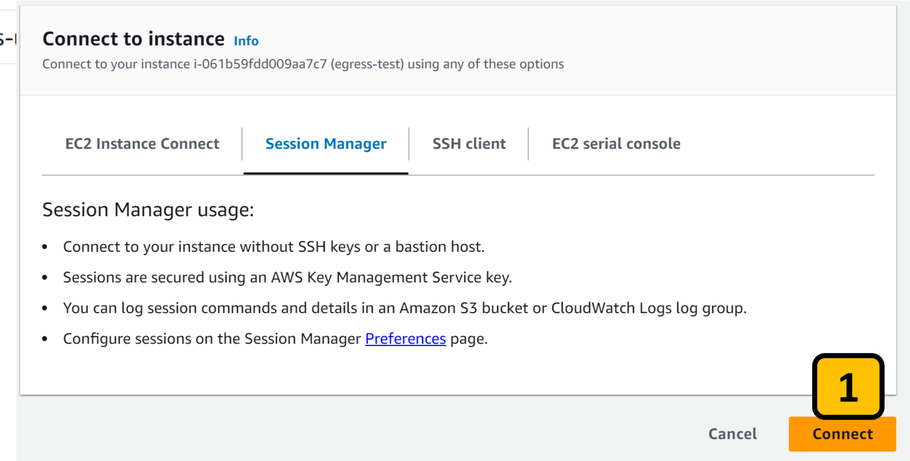
Lab 7: Egress FQDN Filtering: Step 7.5

Access the new instances terminal

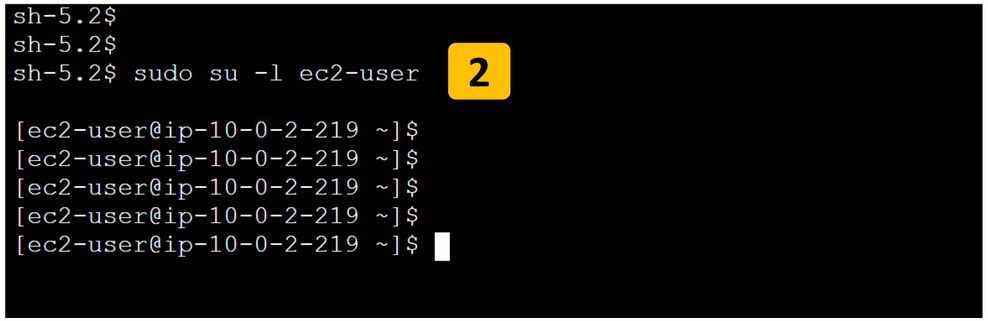
Once your instance is in the running state, access its console using **Session Manager**. 1

Login as ec2-user with the command sudo su –l ec2-user 2





Session ID: brad-0b93adefc7ee7dee0 Instance ID: i-061b59fdd009aa7c7



Lab 7: Egress FQDN Filtering: Step 7.5

Test your domain rules

Let's test our ***.amazonaws.com** rule by running an update for Amazon Linux.

Run an update with the command
sudo yum update –y 1

Test a connection to www.github.com using the command
curl https://www.github.com -v 2

Test a connection to google.com using the command
curl https://google.com -v 3

1

```
[ec2-user@ip-10-0-2-219 ~]$
[ec2-user@ip-10-0-2-219 ~]$ sudo yum update -y
Last metadata expiration check: 0:07:36 ago on Wed Apr 12 01:16:21 2023.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-2-219 ~]$
```

2

```
[ec2-user@ip-10-0-2-219 ~]$
[ec2-user@ip-10-0-2-219 ~]$ curl https://www.github.com -v
*   Trying 140.82.114.3:443...
* Connected to www.github.com (140.82.114.3) port 443 (#0)
* ALPN: offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: /etc/pki/tls/certs/ca-bundle.crt
*  CApath: none
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
```

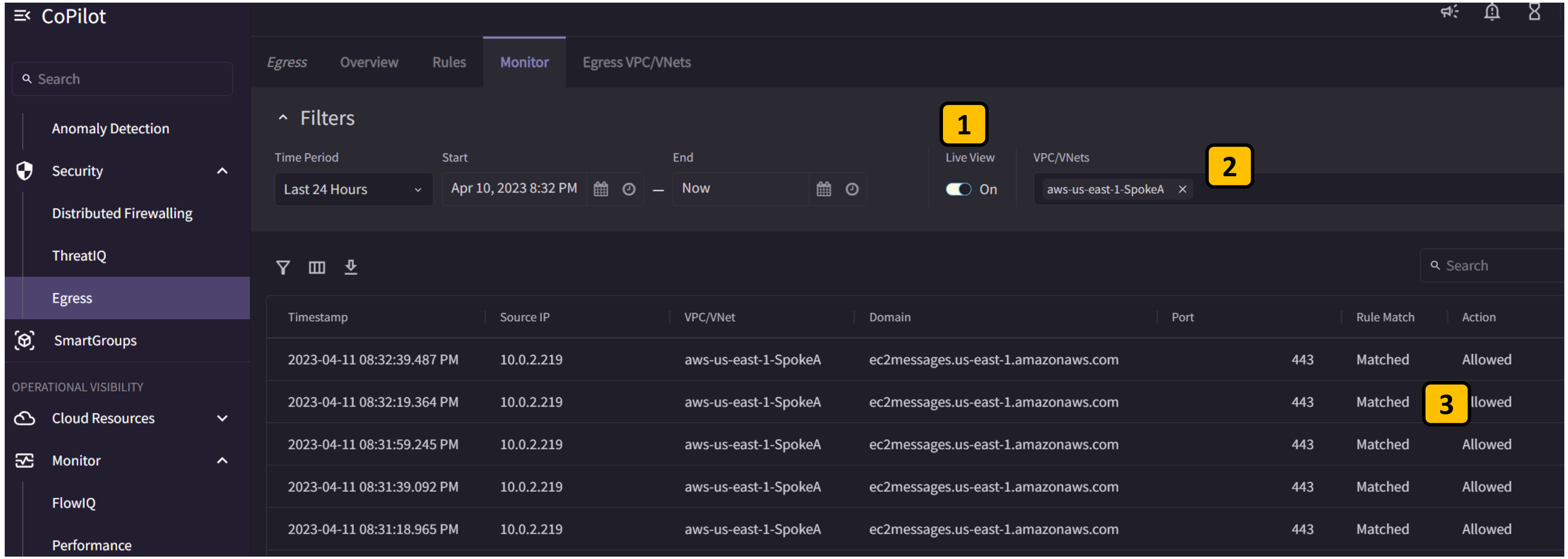
3

```
[ec2-user@ip-10-0-2-219 ~]$ curl https://google.com -v
*   Trying 142.251.16.101:443...
* Connected to google.com (142.251.16.101) port 443 (#0)
* ALPN: offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: /etc/pki/tls/certs/ca-bundle.crt
*  CApath: none
* OpenSSL SSL_connect: SSL_ERROR_SYSCALL in connection to google.com:443
* Closing connection 0
curl: (35) OpenSSL SSL_connect: SSL_ERROR_SYSCALL in connection to google.com:443
[ec2-user@ip-10-0-2-219 ~]$
```

Did you get the results that you expected??

Lab 7: Egress FQDN Filtering: Step 7.5

View domain access activity in real time



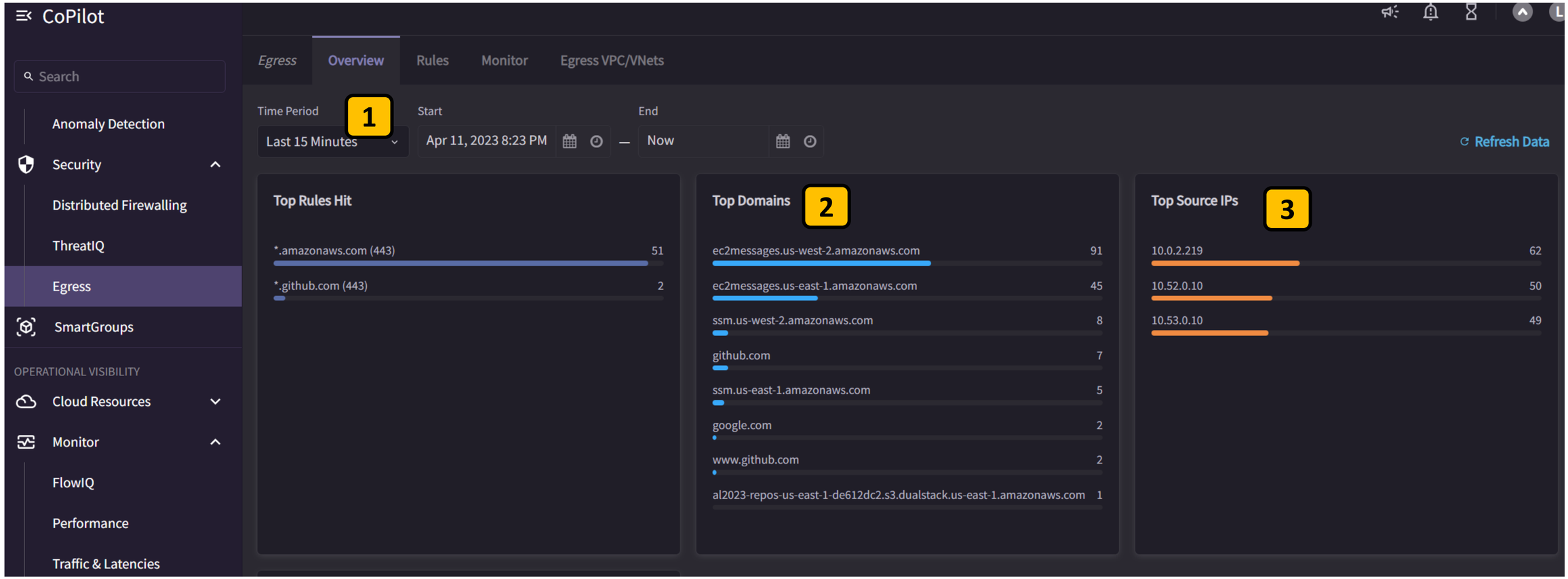
Select the Monitor tab and enable **Live View** 1

Select the gateway aws-us-east-1-SpokeA gateway in the VPC/Vnets dropdown 2

Observe the domain-based traffic in real time and notice Rule Match and Action columns. 3

Lab 7: Egress FQDN Filtering: Step 7.5

View domain access activity in real time



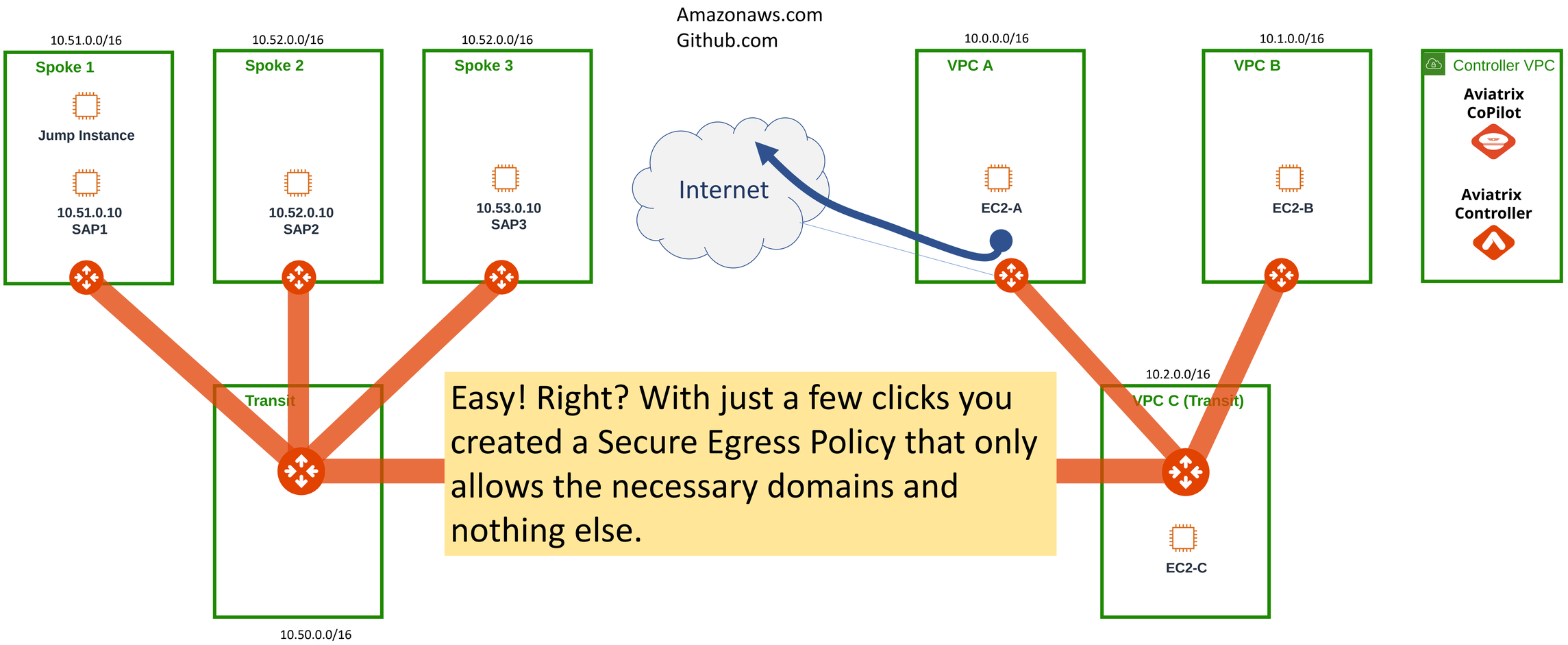
Select the Overview tab and select the Last 15 Minutes of history 1

Observe the summary top accessed domains 2

Observe the top talkers of this domain-based Egress traffic. 3

Lab 7 Success

Egress FQDN Filtering



Easy! Right? With just a few clicks you created a Secure Egress Policy that only allows the necessary domains and nothing else.

AWS us-west-2 **Aviatrix FQDN Filtering** **AWS us-east-1**