Add GW to existing AWS

Tuesday, December 4, 2018 2:29 PM

Adding a Cloud Guard cluster into an existing AWS environment

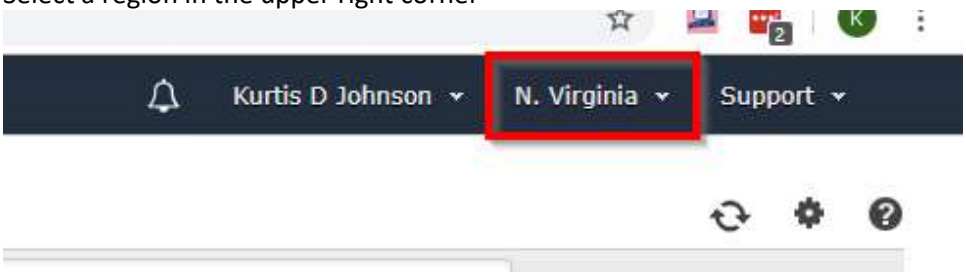
Created 11/20/2018 by Kurt Johnson (SE - KY/OH)

Section 1 - Setup or confirm the initial AWS environment

1. Log into AWS account:
   https://console.aws.amazon.com

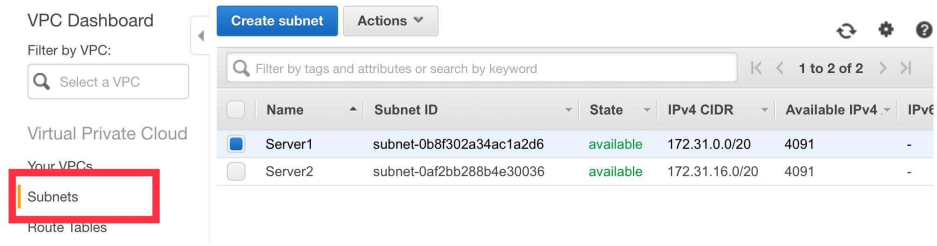
2. Add shortcuts to the top bar by clicking the pushpin. It may be helpful to add EC2, VPC, CloudFormation, and IAM

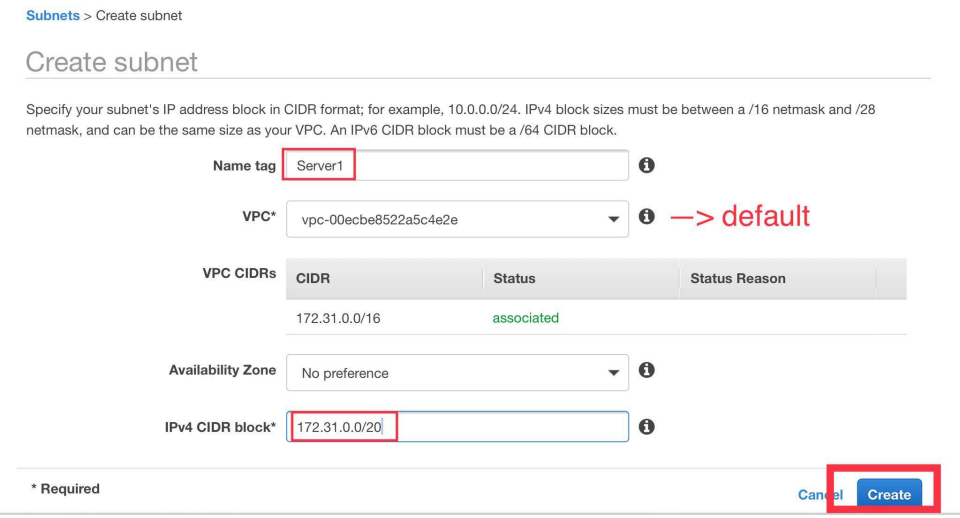
Select a region in the upper right corner



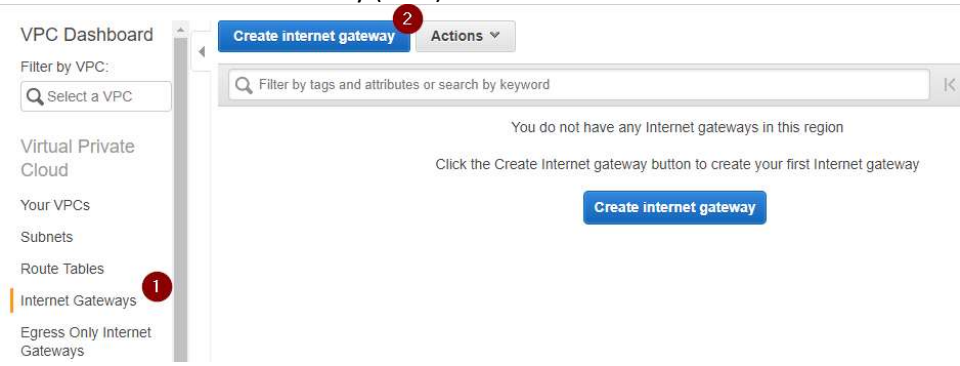
3. Review 'Your VPCs' if one doesn't exist with the CIDR 172.31.0.0/16, Create one.

4. Click on 'Subnets' on the left and create two of them
   Server1 - 172.31.0.0/20
   Server2 - 172.31.16.0/20



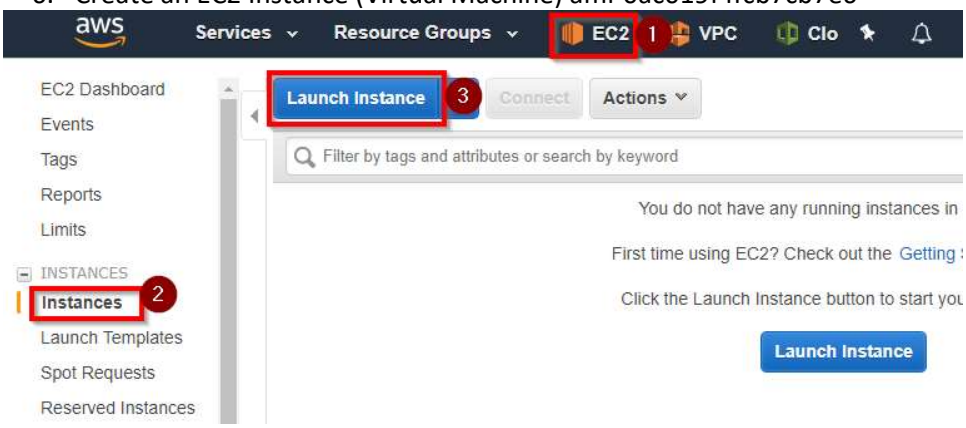


5. Create Internet Gateway (IGW)



a. Attach the IGW to the VPC

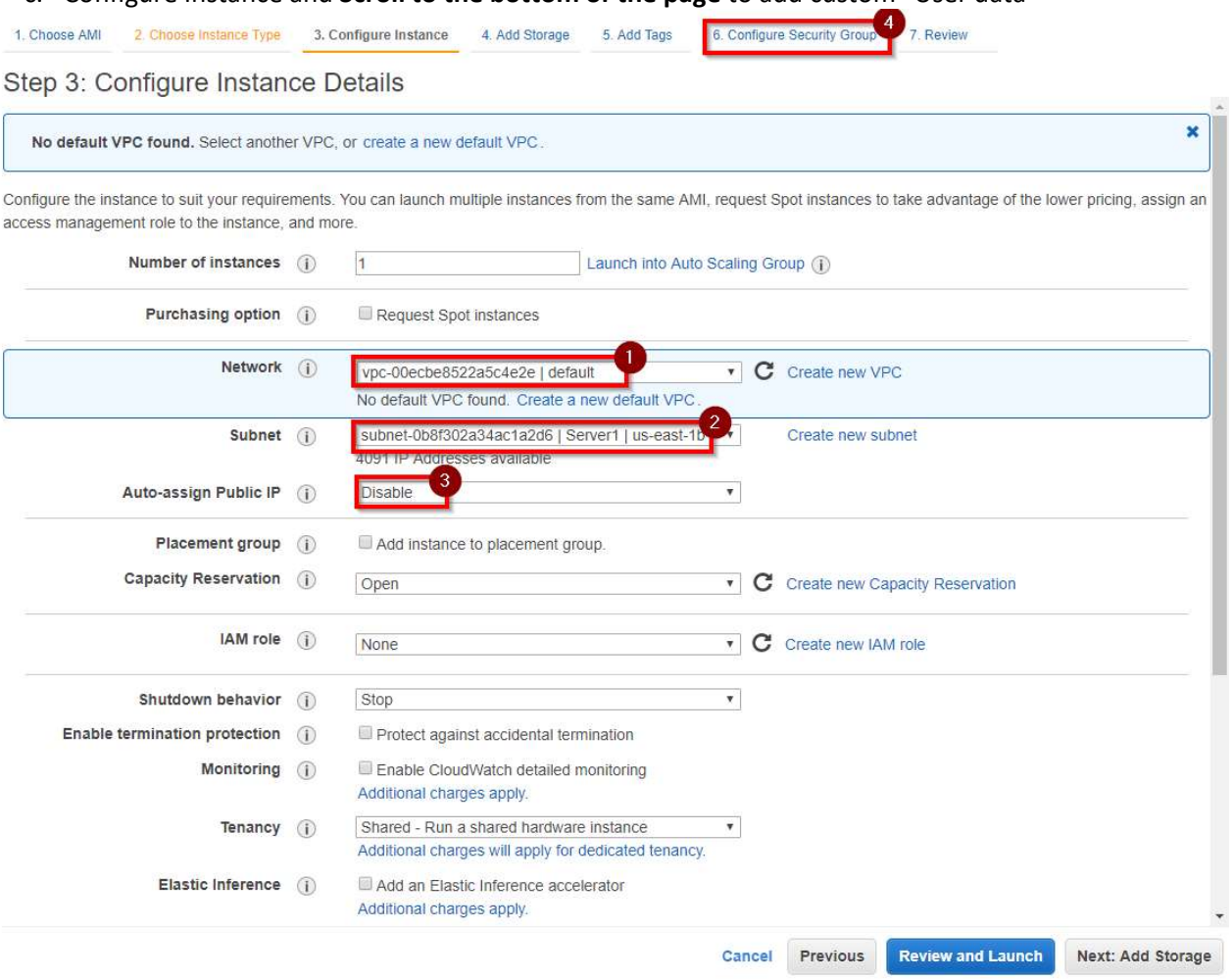
6. Create an EC2 Instance (Virtual Machine) ami-0ac019f4fcb7cb7e6



a. Search for 'ami-0ac019f4fcb7cb7e6' and select 64-bit

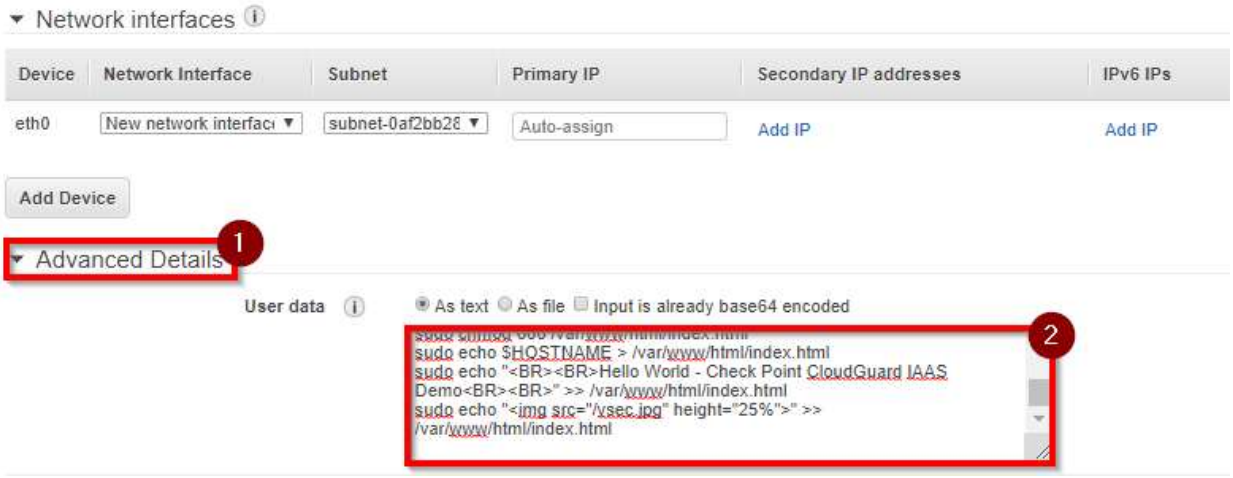
b. Select they instance type - t2.micro will be fine

c. Configure Instance and **Scroll to the bottom of the page** to add custom "User data"



d. Before Configuring Security Groups, scroll to the Advanced Details and paste the script below:

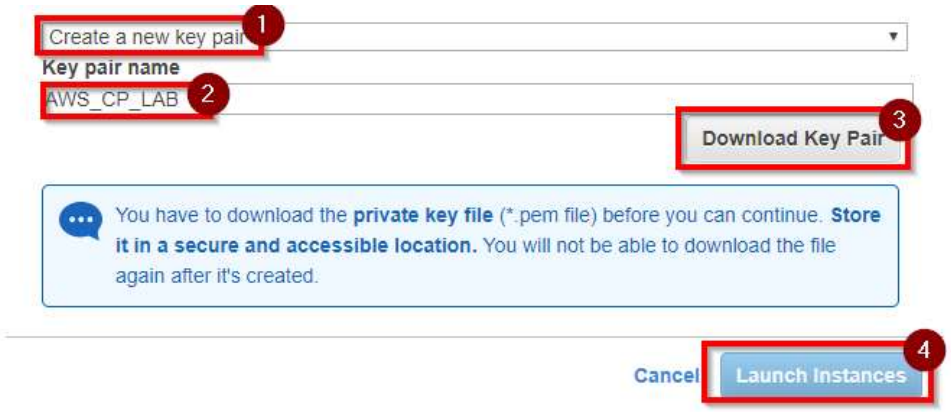
```
==begin script==
#!/bin/bash
until sudo apt update && sudo apt -y install apache2;do
sleep 1
done
until sudo curl --output /var/www/html/vsec.jpg --url https://www.checkpoint.com/wp-content/uploads/cloudguard-iaas-236x150.png ; do
sleep 1
done
sudo chmod 666 /var/www/html/index.html
sudo echo $HOSTNAME > /var/www/html/index.html
sudo echo "<BR><BR>Hello World - Check Point CloudGuard IAAS Demo<BR><BR>" >> /var/www/html/index.html
sudo echo "<img src="/vsec.jpg" height="25%">" >> /var/www/html/index.html
==end script==
```



e. Configure Security Groups: Add rules for All ICMP - IPv4 and HTTP

f. Select Launch

g. If you don't already have a Key Pair in AWS, create a new one called AWS_CP_LAB
Select 'Download Key Pair'
Select 'Launch Instances'

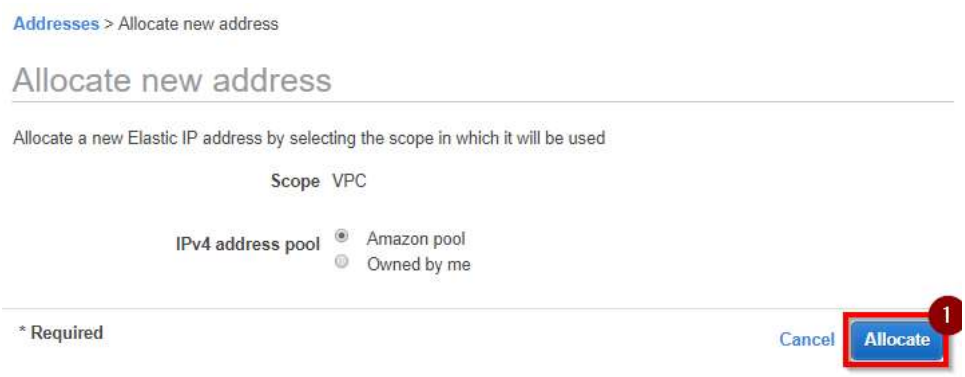


Can you PING it? What IP address?
--No Public IP has been assigned.

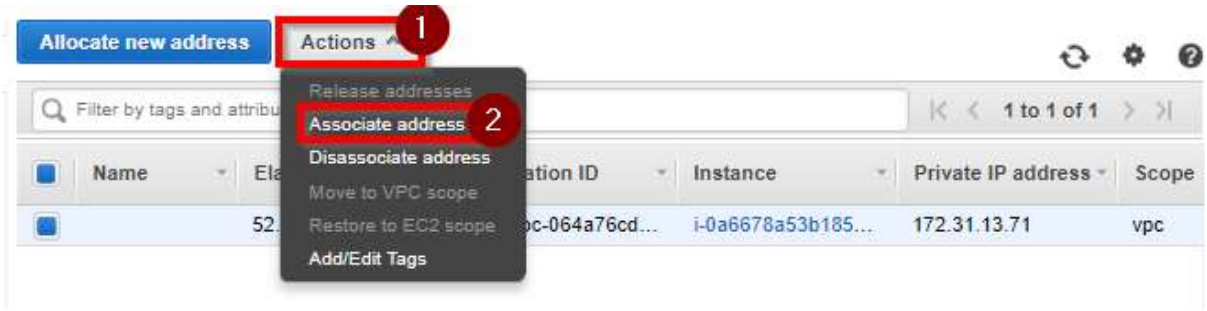
7. Create/allocate and Associate a new public address to your host

a. Click on your VPC -> Elastic IPs (on the left)

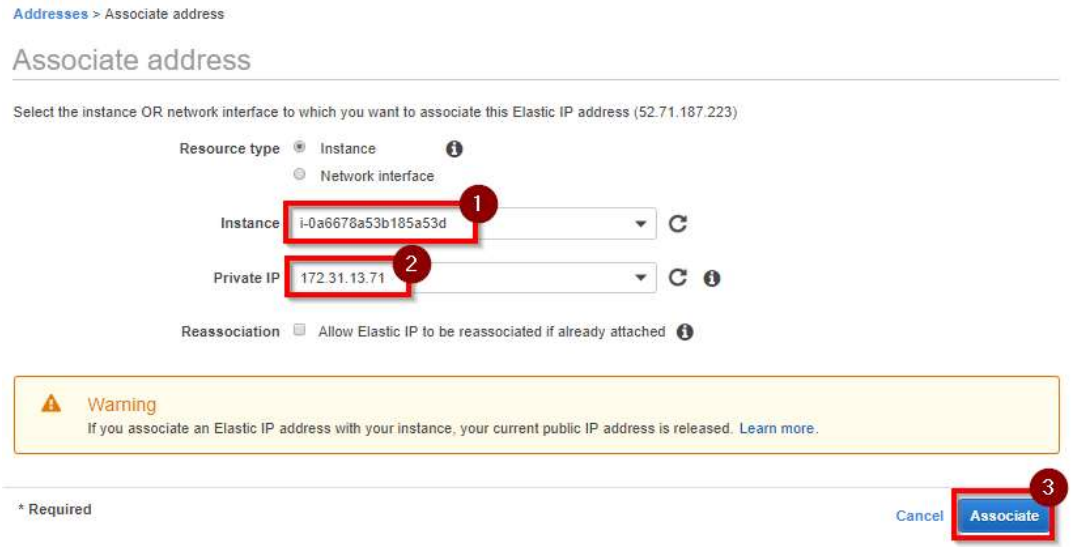
b. Allocate new address



c. Click on Actions -> Associate address



d. Bind the EIP to the Instance IP (Your IP and Instance ID will be different)



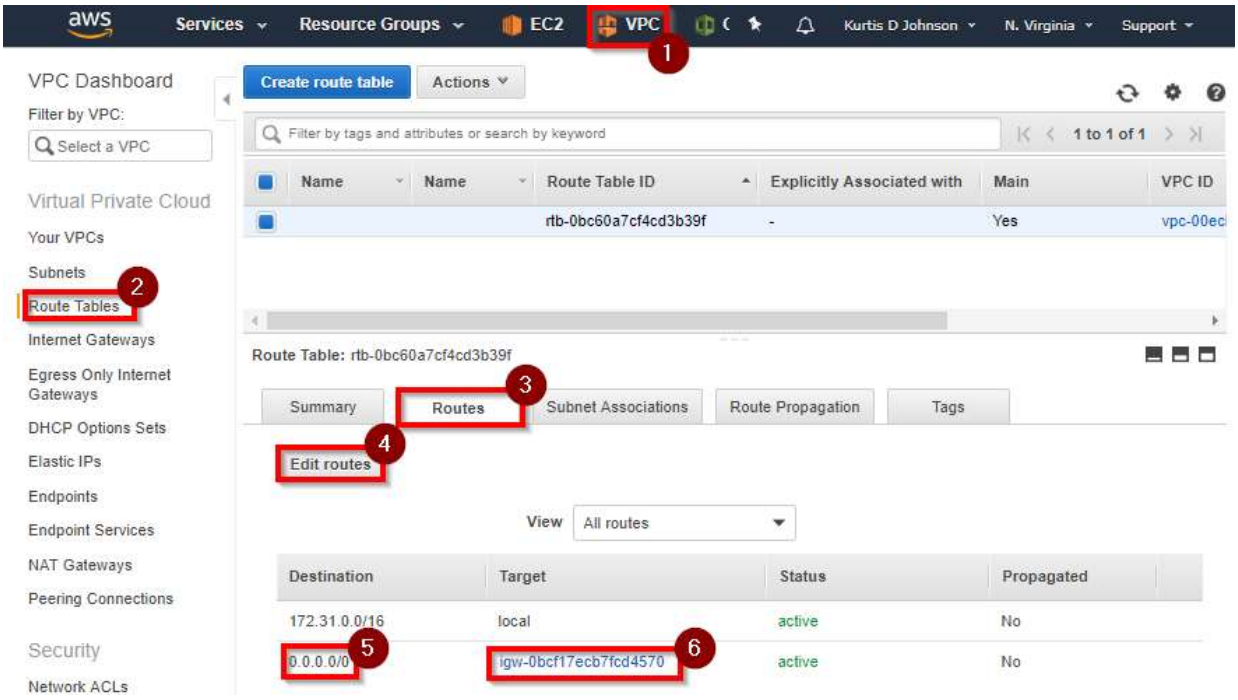
Can you PING the Elastic (public) IP now?

--Technically, the packet is getting to the host, but no packets are returning.

8. Add a default route leveraging the IGW:

a. Select VPC -> Route Tables -> Routes (tab) -> Edit routes

b. Add a 0.0.0.0/0 using the IGW created earlier



You should now receive replies from your PING.
You should also receive a response on HTTP

Congratulations!!! Your AWS instance is connected… to EVERYTHING on the Internet.

Section 2 - Add a Cloud Guard cluster

1. Add two new subnets: (VPC -> Subnets)
   FW_outside - 172.31.254.0/24 - Availability Zone A
   FW_inside - 172.31.253.0/24 - Availability Zone A

2. Launch a CloudFormation Template to deploy a cluster in an existing VPC
https://console.aws.amazon.com/cloudformation/home?#/stacks/create/review?templateURL=https%3A%2F%2Fs3.amazonaws.com%2FCloudFormationTemplate%2Fcluster-into-vpc.json&stackName=Check-Point-Cluster
   a. Enter the information below

VPC Network Configuration

VPC	<the one you created earlier>
Availability Zone	A <same as the Subnets>
External subnet	FW_outside
Internal subnet	FW_inside

Cluster Network Configuration

Cluster external address	172.31.254.10
Member A external address	172.31.254.20
Member B external address	172.31.254.30

Cluster internal address	172.31.253.10
Member A internal address	172.31.253.20
Member B internal address	172.31.253.30

EC2 Instance Configuration

Instance type	C4.xlarge <default>
Key name	<same from Section 1-6-g>

Check Point Settings

License	R80.10-BYOL
Admin shell	/bin/bash
Password hash	**1mX1Pn5NV$nK6n18yxt1AvfgITZpDn.1**
---	That is the hash of **zaq1@WSXcde3**
SIC key	abcd1234
Allow upload & download	true <default>
Primary NTP server	169.254.169.123 <default>
Secondary NTP server	0.pool.ntp.org <default>

Capabilities

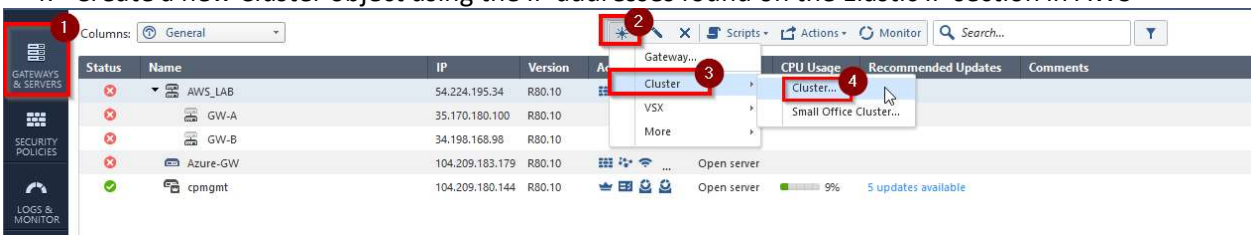
☑ I acknowledge that AWS CloudFormation might create IAM resources with custom names
☑ I acknowledge that AWS CloudFormation might require the following capability: ------

b. Click Create

3. Wait for approximately 5 minutes and record the Elastic IP information

Cluster info	Private IP address	Public IP address
VIP	172.31.254.10	100.26.128.32
Member A	172.31.254.20	54.84.135.202
Member B	172.31.254.30	54.152.247.164

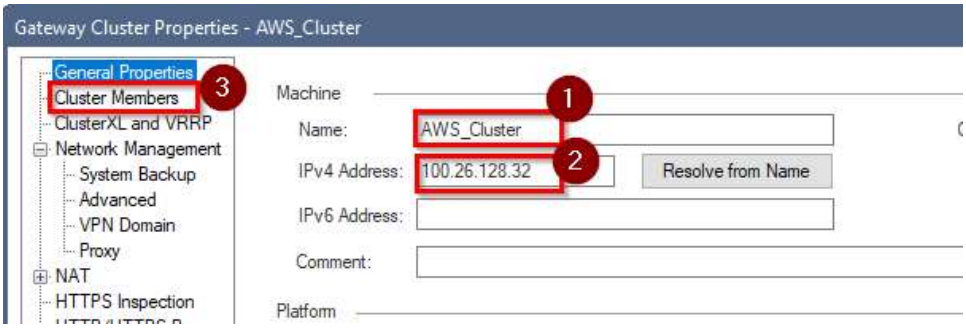
4. Create a new Cluster object using the IP addresses found on the Elastic IP section in AWS



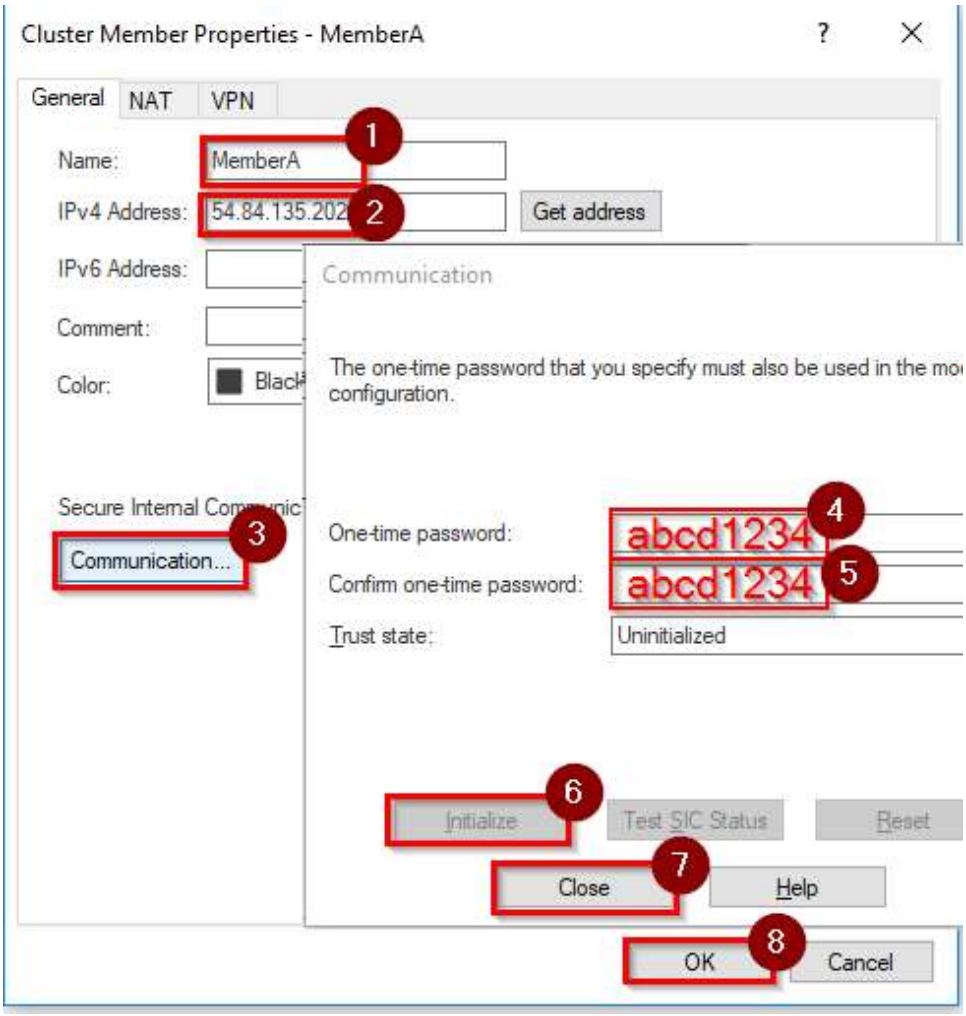
a. Select Classic mode

b. Input a Name and the public IP address tied to the VIP 172.31.254.10 from your Elastic IP information.

c. Then select Cluster Members

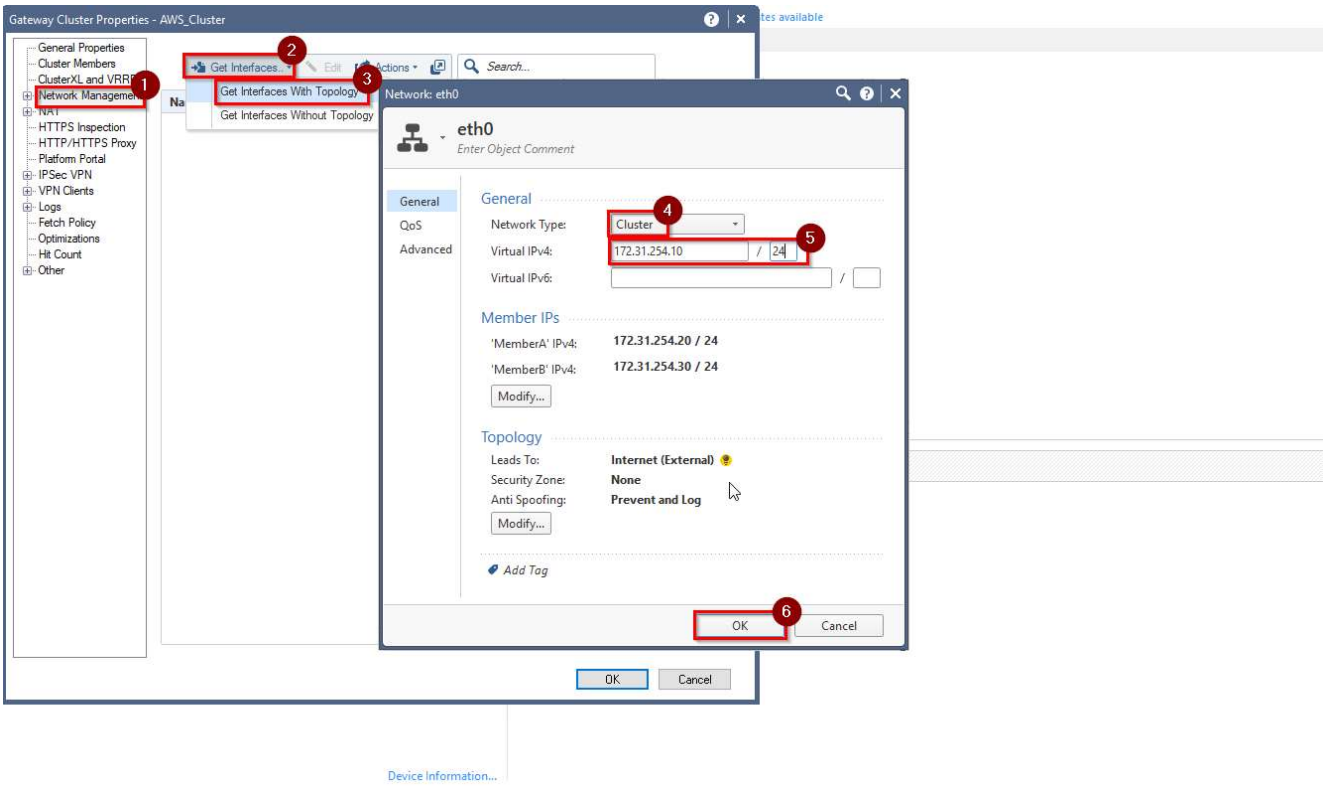


d. Add -> New Cluster Member…

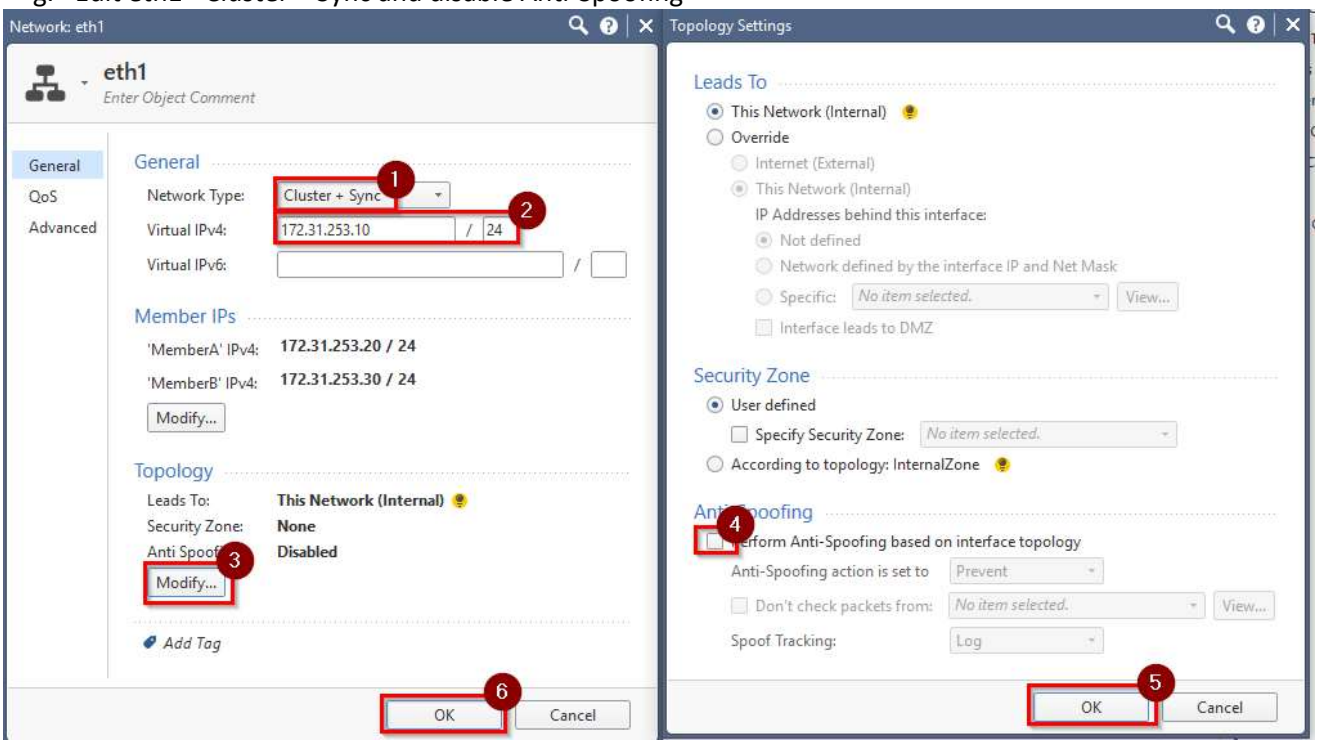


e. Repeat the above step for Member B

f. Configure Network Management for eth0 as a Cluster interface

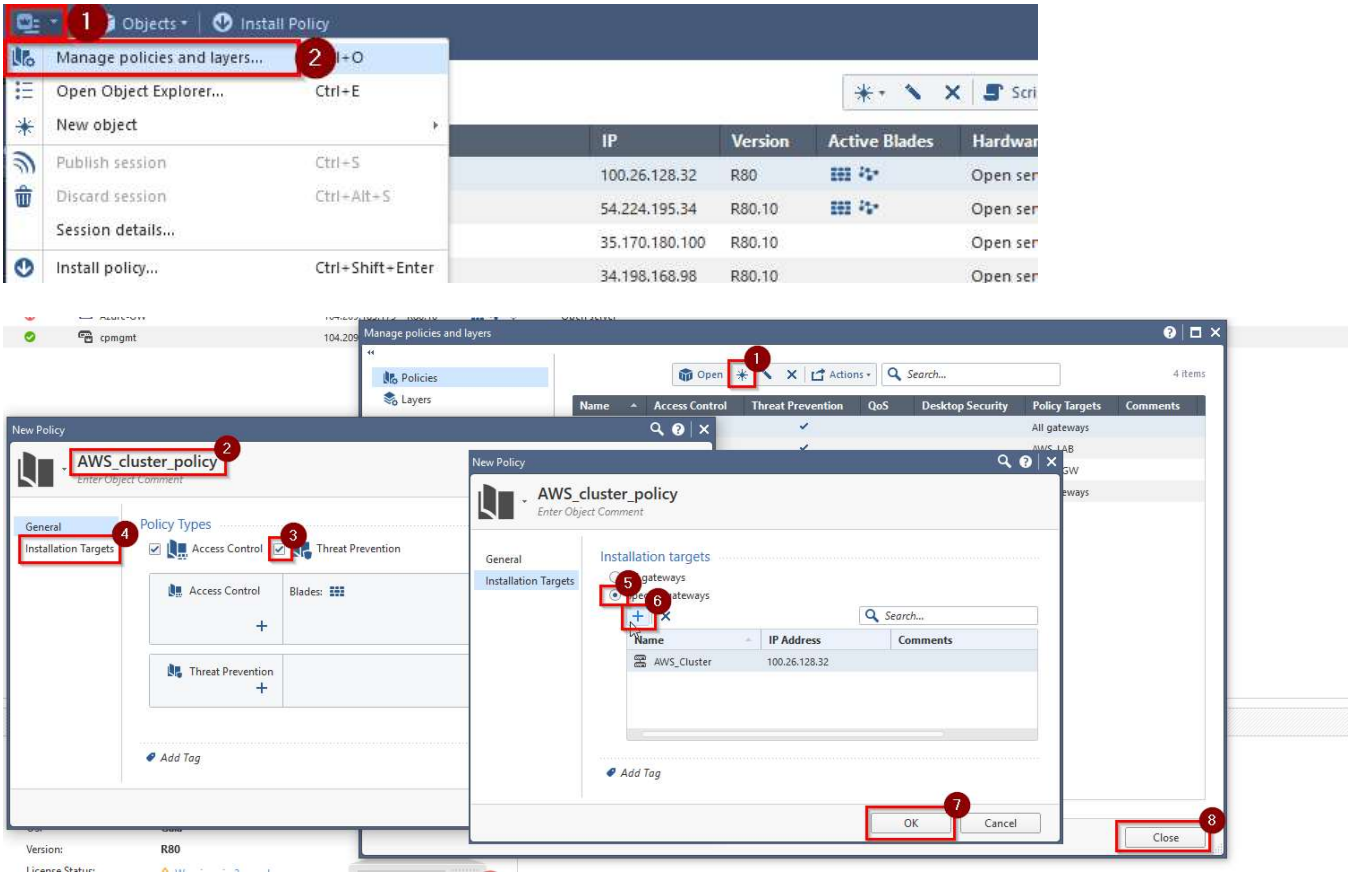


g. Edit eth1 - Cluster + Sync and disable Anti-Spoofing

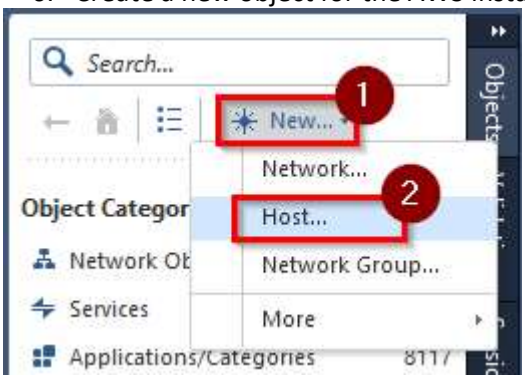


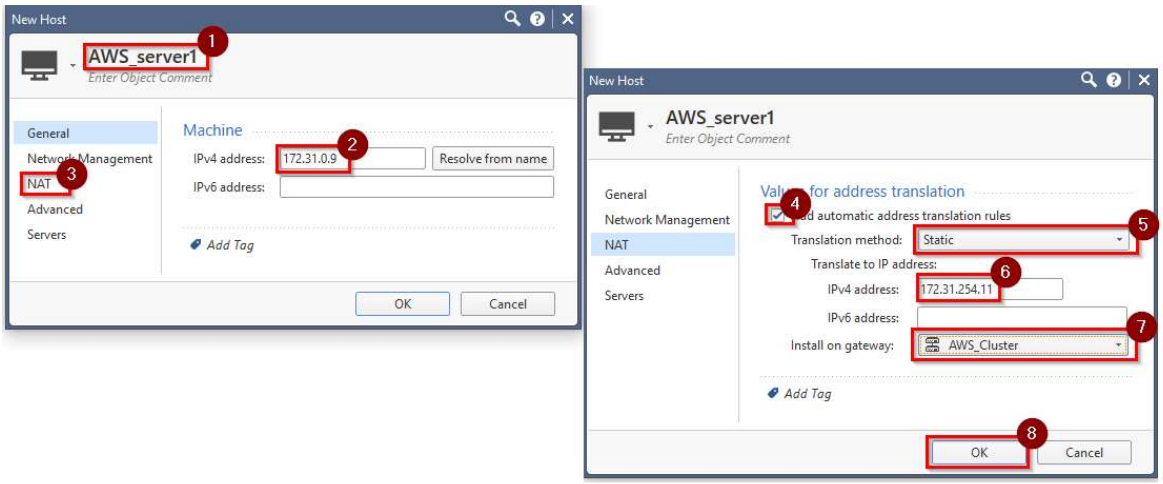
h. Click OK to complete the Cluster setup
i. Click Yes to the warning dialog box - VPN Link Selection

5. Create a new policy:



6. Create a new object for the AWS instance and add a static NAT as 172.31.254.11

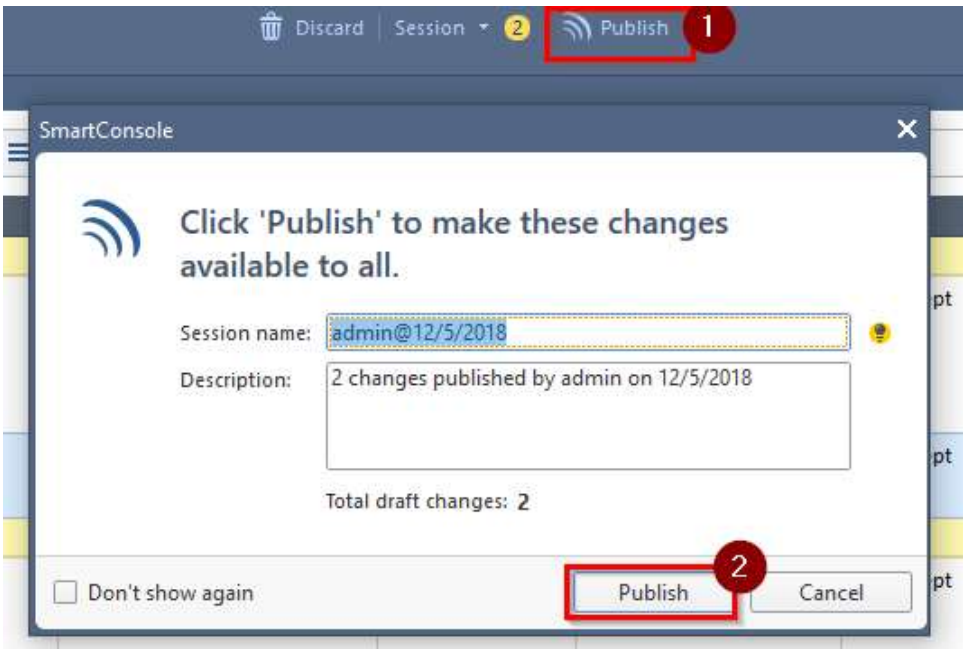




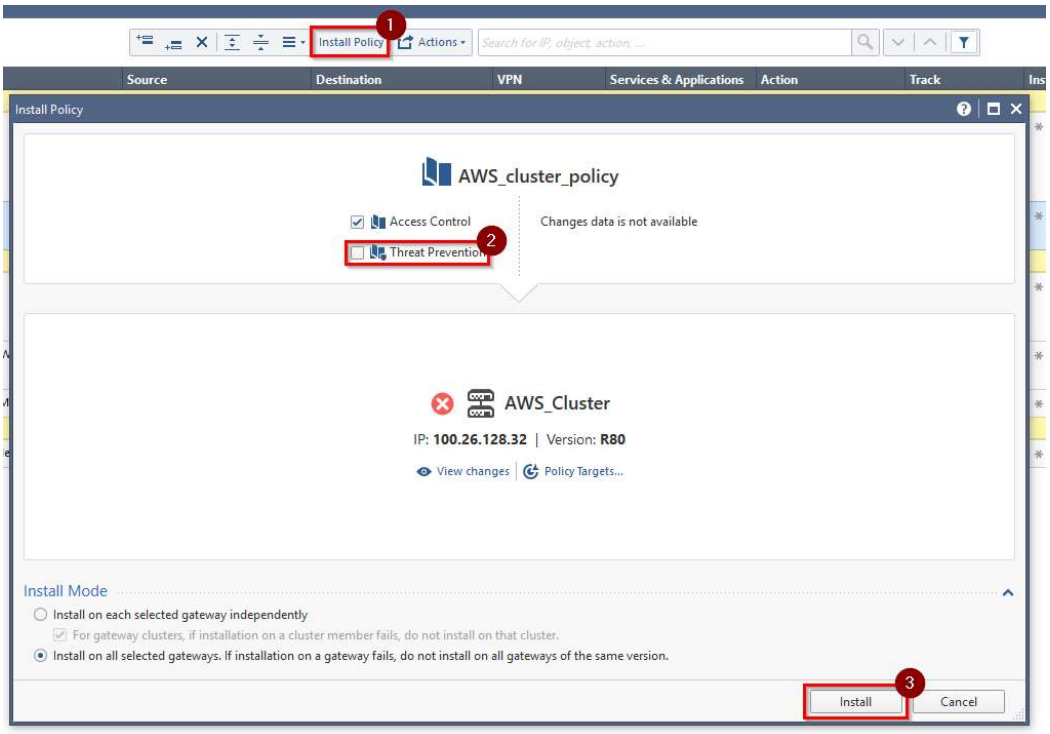
7. Create the following rulebase

No.	Name	Source	Destination	VPN	Services & Applications	Action	Track
Mgmt (1-2)							
1		Any	AWS_Cluster	Any	http ssh_version_2 https echo-request	Accept	Log
2		AWS_Cluster	Any	Any	https echo-request	Accept	Log
Traffic (3-5)							
3		Any	AWS_server1	Any	http ssh_version_2 echo-request	Accept	Log
4	Allowed WWW out	AWS_server1	Any	Any	http https	Accept	Log
5	Allowed ICMP out	AWS_server1	Any	Any	echo-request	Accept	Log
Cleanup (6)							
6	Cleanup rule	Any	Any	Any	Any	Drop	Log

8. Publish



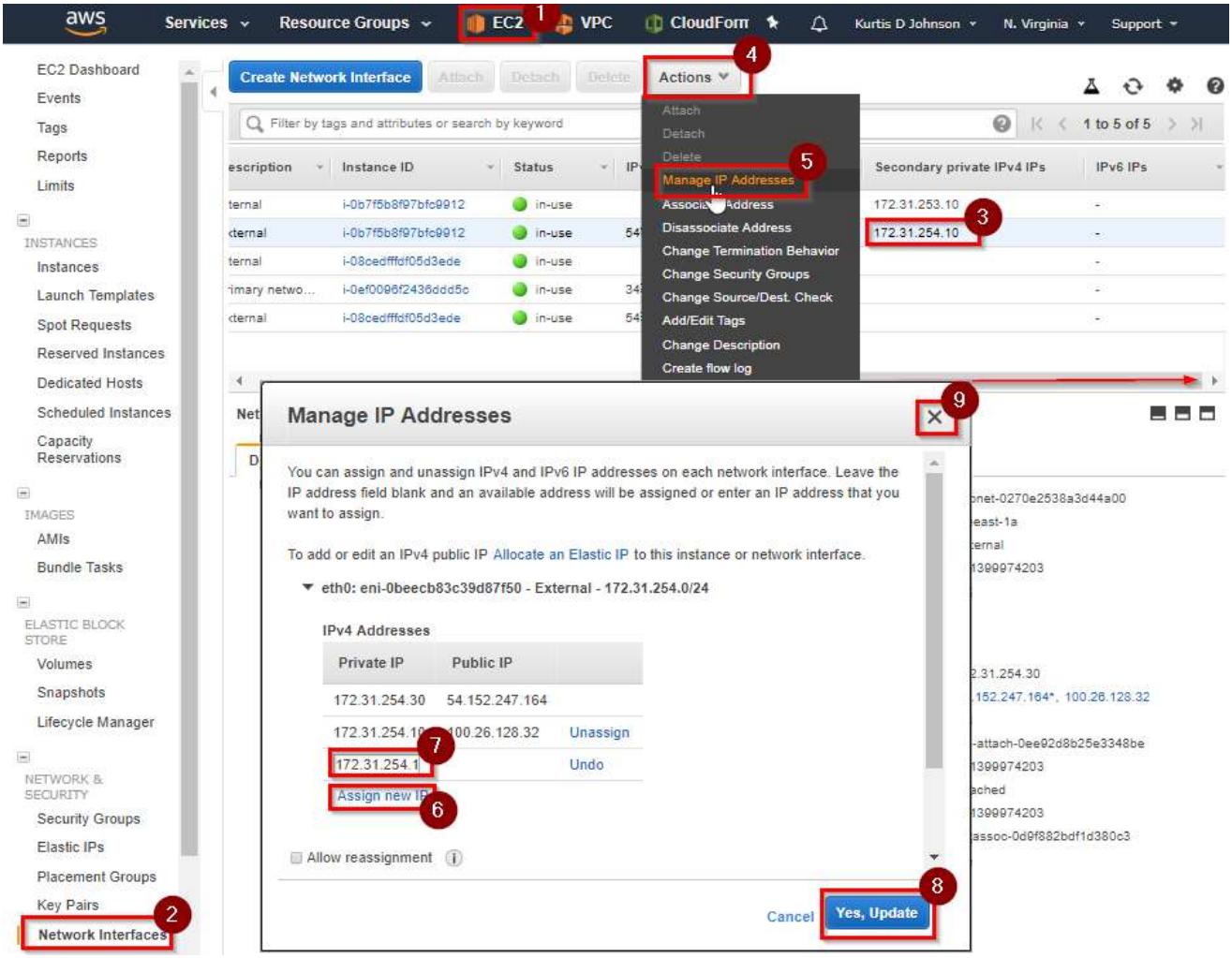
9. Install (Clear Threat Prevention for first policy install)



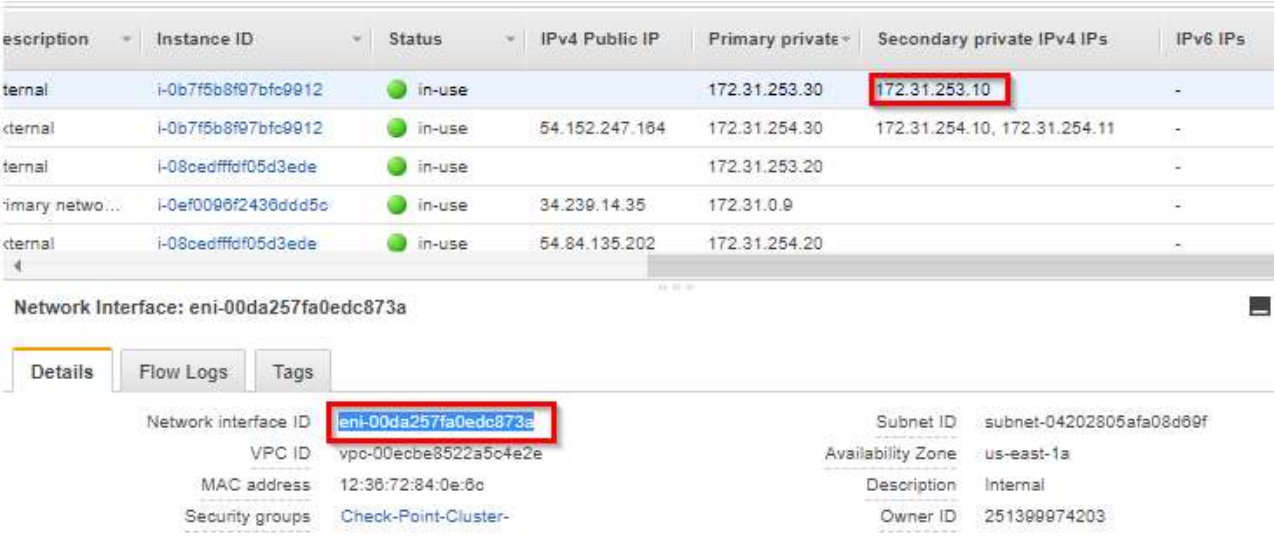
Section 3 - Add NAT and Routes

1. Add new address to the outside for NAT

a. AWS -> EC2 ->Network Interfaces (scroll to the right and edit the interface holding the 172.31.254.10 address)
   - At click 7 <in the image below> input 172.31.254.11

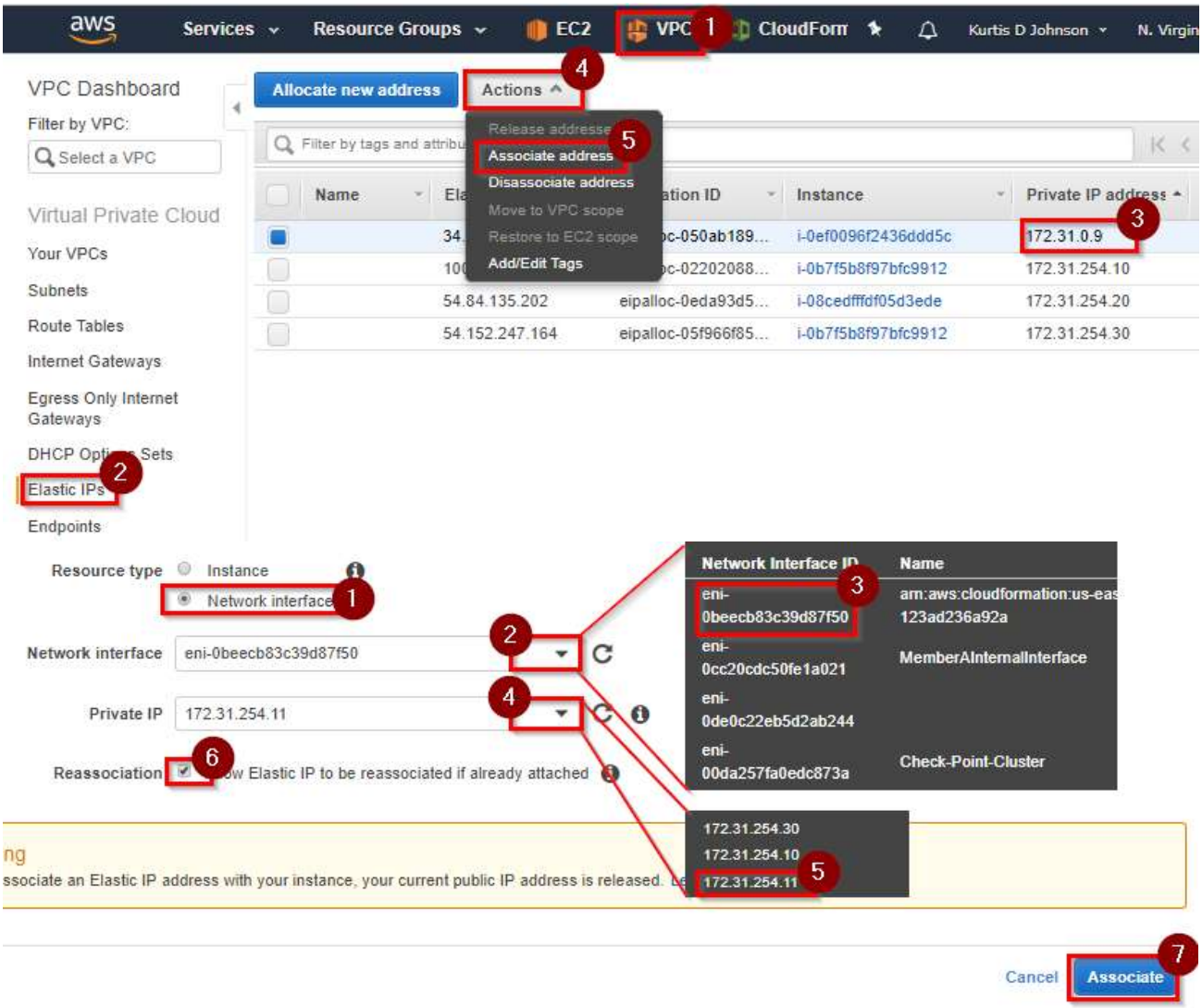


b. Record the ENI of the Internal Cluster VIP 172.31.253.10 to be used in routes eni-00da257fa0edc873a

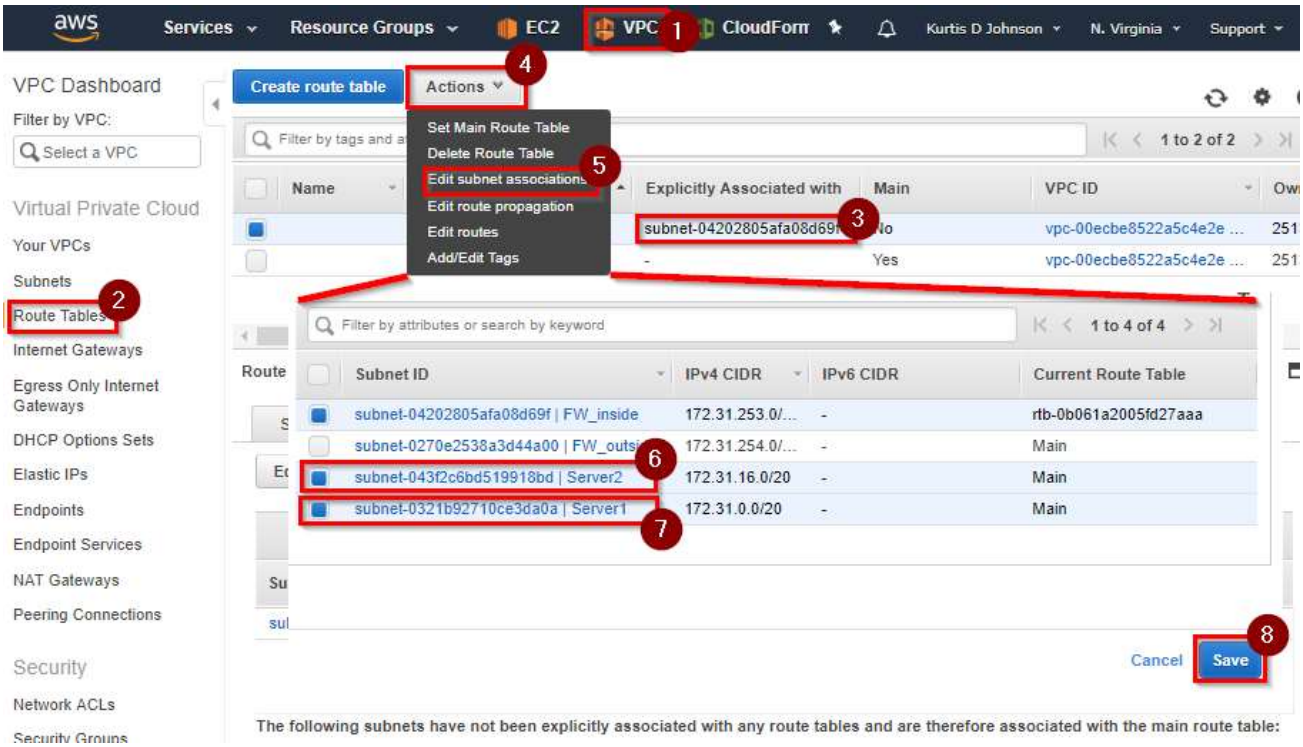


NEXT STEPS WILL CAUSE A DISCONNECT UNTIL WE ROUTE TRAFFIC BACK THROUGH THE FIREWALL

2. Re-associate our Elastic IP from the Server1 instance to the Firewalls External Interface.



3. A new Route table was created from the CloudFormation Template. Modify the Internal Routing table and edit the Subnet Associations to include Server1 - 172.31.0.0/20 and Server2 - 172.31.16.0/20. FW_inside - 172.31.253.0/24 should ALREADY be associated.



Traffic should now be flowing through the Firewall

Extra step 1:

Enable IPS, Anti-Bot, and Anti-Virus in PREVENTION mode

Install Threat Prevention Policy

SSH into the Ubuntu server (Elastic IP)
Test AV from the Ubuntu server with CURL:
curl http://www.eicar.org/download/eicar.com -o eicar.com.txt

Do you see a drop in the logs?

Extra step 2:

SSH into the Ubuntu server (Elastic IP) - execute:

while true; do curl www.google.com --max-time 5 >/dev/null; sleep 5; done

In a new window, SSH into the Firewall Cluster VIP - execute:

clusterXL_admin down

AWS interfaces will automatically be moved from one member to the other, but your Ubuntu session should come back with approximately 5 CURLs timed out.

Clean up:

DELETE the cluster via CloudFormation
DELETE the instance via EC2