Reference documentation :
https://docs.paloaltonetworks.com/prisma/prisma-access/prisma-access-panorama-integration/secure-your-public-cloud-deployment-with-prisma-access/onboard-aws-vpcs

Workflow:

In Prisma Access:

Step 1:

Create appropriate cryptography settings within Prisma Access for connection to AWS. See steps 1-3 in the doc referenced above

Step 2:

Create two IKE gateways, one for the primary and one for the secondary VPN tunnel, on Prisma Access

*directions in link above indicated that version should be set to “IKEv1 only” however we found that things only worked when we set it to IKEv2 Preferred and set the advanced options for each IKEv1 and IKEv2 as shown later in this doc

*AWS will not create an IPSec tunnel until it has an IP to target on the Prisma Access side. So we must first create, on the Prisma Access side, a “dynamic” address type IKE gateway with phony values. This will make Prisma Access create a service connection node and assign it a public IP.

We will come back and correct the settings later, for now you can create the IKE gateways according to Step 4 in the above linked doc

Step 3:

Create the IPSec Tunnels according to Steps 5-6 in the referenced doc

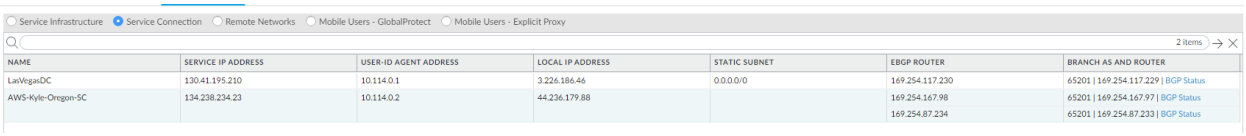
Step 4:

Set up a Service Connection (instead of a Remote Network as referenced in the doc). Specify your two IPSec tunnels created in the last step as the primary and secondary WAN connections.

Enable BGP using dummy values according to step 7

Commit, wait up to 30 minutes for the nodes to build

Grab the “Service IP Address” from Panorama > Cloud Services > Status > Network Details > Service Connection. In our case this is 134.238.234.23



In AWS:

Step 1:

If you have not already, create VPC, Route table, subnets. If you are using a Transit Gateway, create this as well

Follow steps 1-2 to create a Customer Gateway. This is an AWS object that holds information about the Prisma Access node that AWS will interact with. The IP Address used here is the Service IP you just pulled from Prisma Access

Step 2:

Create a Virtual Private Gateway as per the document. This defines how the gateway on the AWS side should behave

When setting routing options, select BGP and make sure you provide the ASN of the Prisma Access Infrastructure subnet. You can find this in Panorama > Cloud Services > Configuration > Service Infrastructure

Attach this VPG to the VPC **or to a Transit Gateway** if you are using that

*side note - only use a VPG attached to a VPC if you are only trying to access resources in that VPC. If you want to access resources in multiple VPCs from Prisma Access, you must use a Transit Gateway and/or deploy a VM in your AWS network to terminate the Service Connection. This is an AWS networking limitation of how it handles S2S connections direct to an individual VPC

Step 3:

Create a VPN Connection (“Site to Site VPN Connection” in the AWS console, under VPC dashboard)

Set it up to use BGP if asked, and create + record a PSK

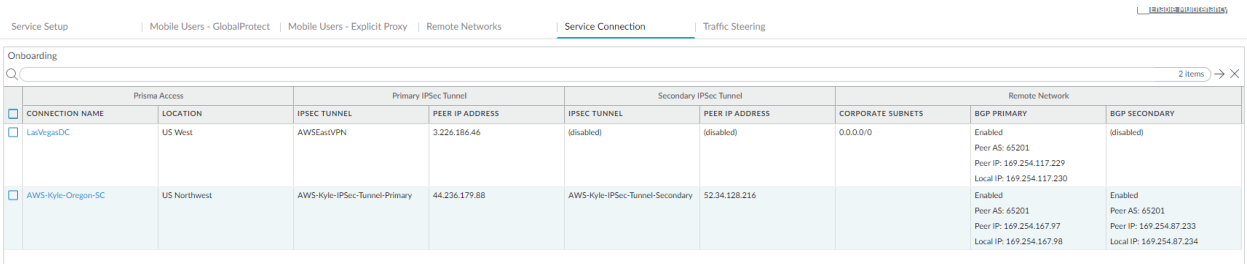
Once completed, go to the tunnel details tab and you can find the Outside IP Address and Inside IP CIDR of the active and passive tunnels. Here is what ours looked like

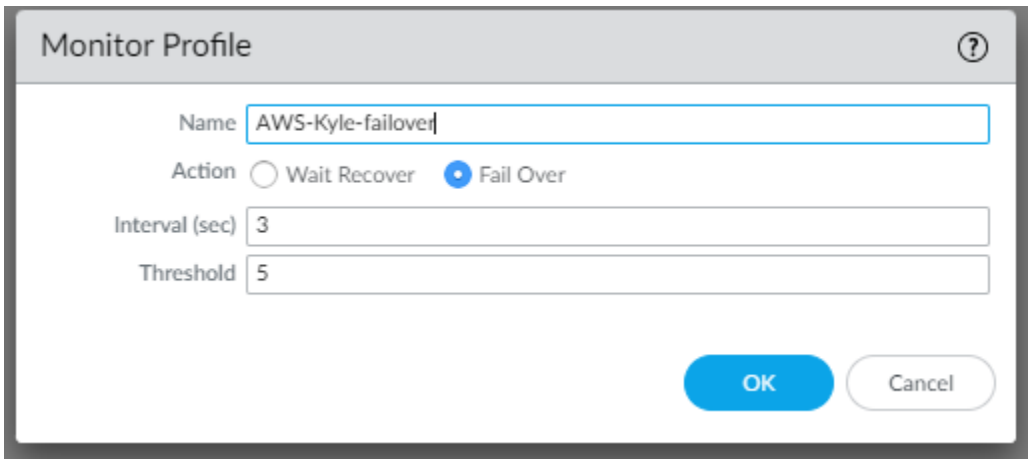
Tunnel state

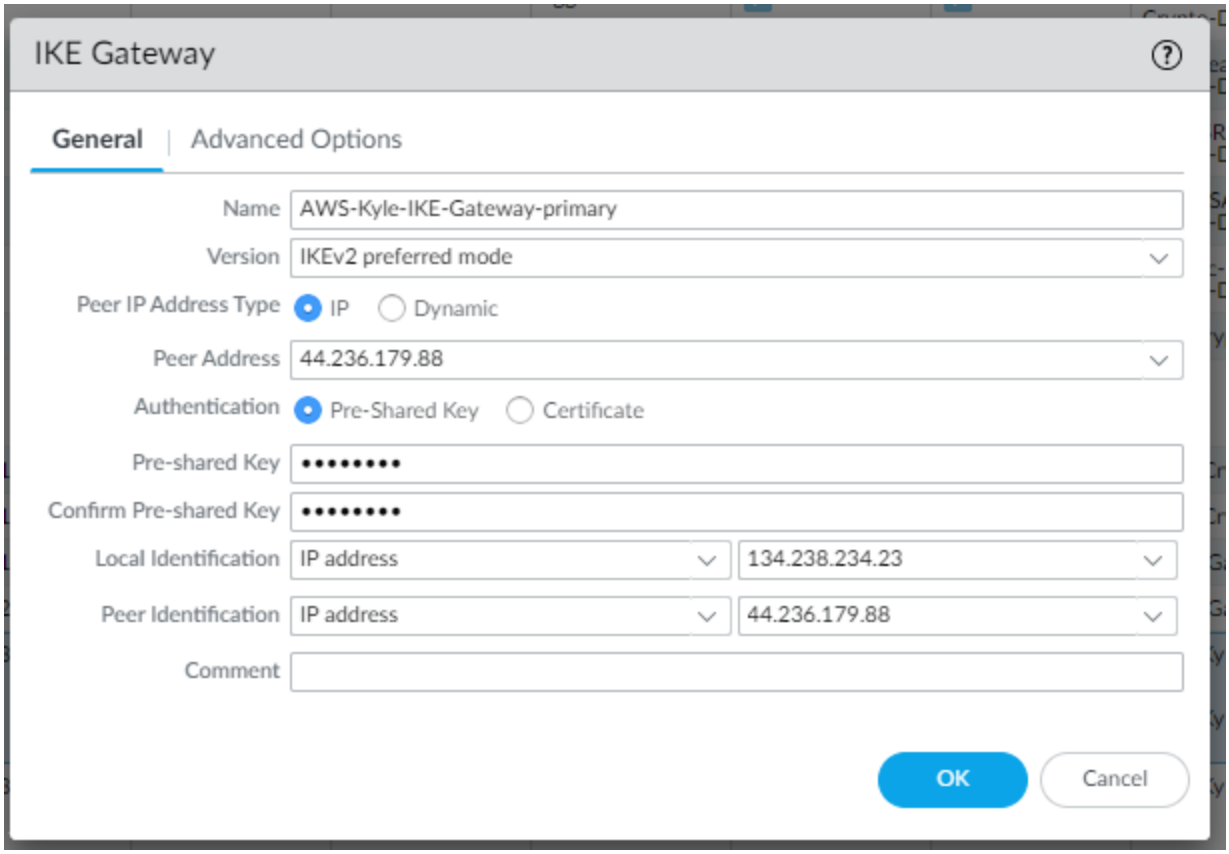
Tunnel number	Outside IP address	Inside IPv4 CIDR
Tunnel 1	44.236.179.88	169.254.167.96/30
Tunnel 2	52.34.128.216	169.254.87.232/30

Back to Prisma Access:

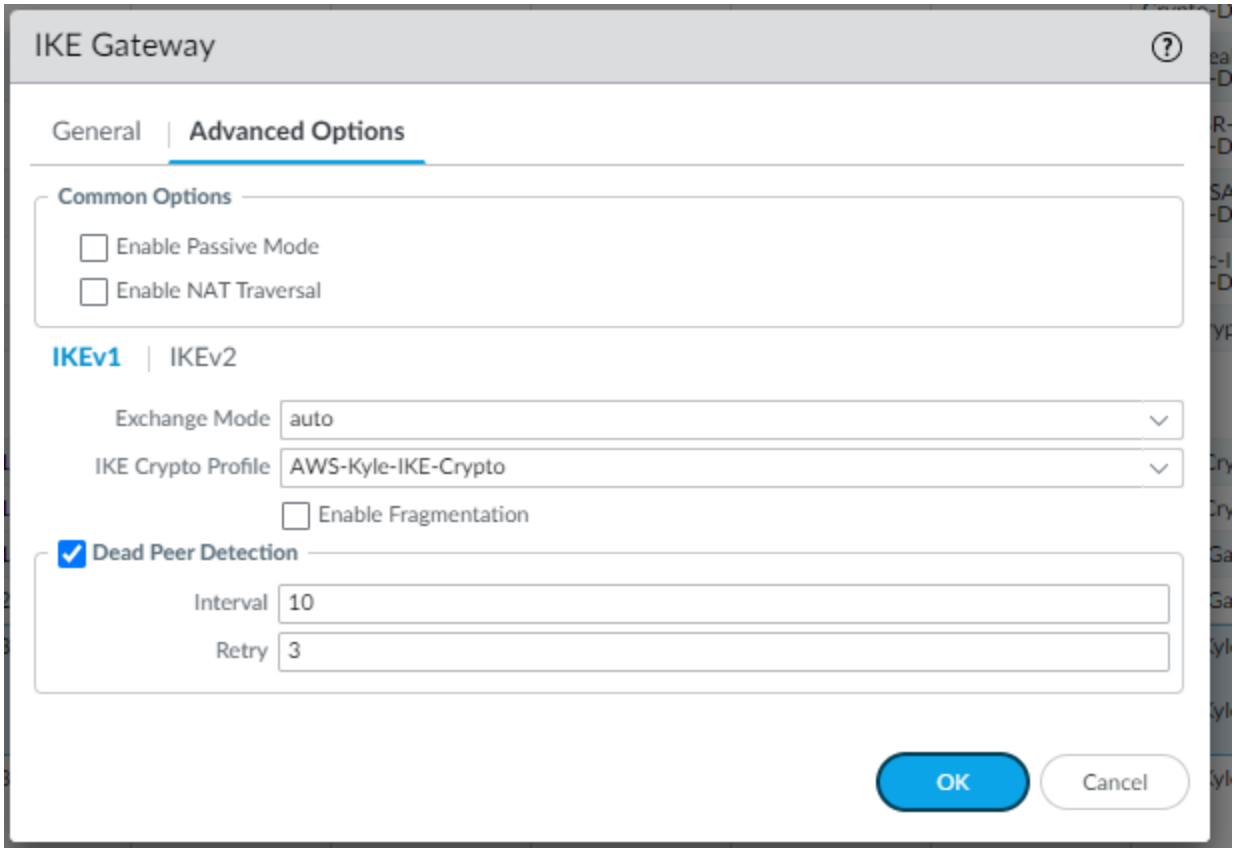
Configure your monitor profile, IKE Gateways, etc, to look like this. We are essentially updating everything to use the above IPs we just got from AWS, and set up monitoring, etc to match AWS' defaults



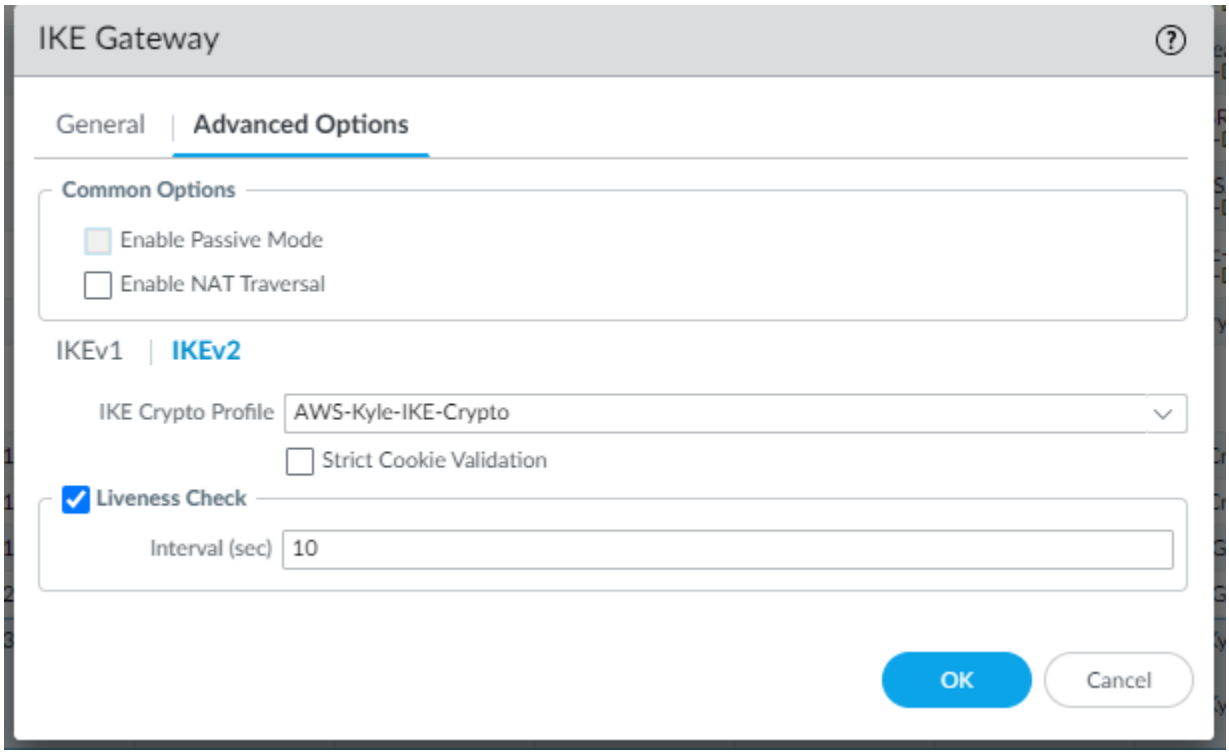




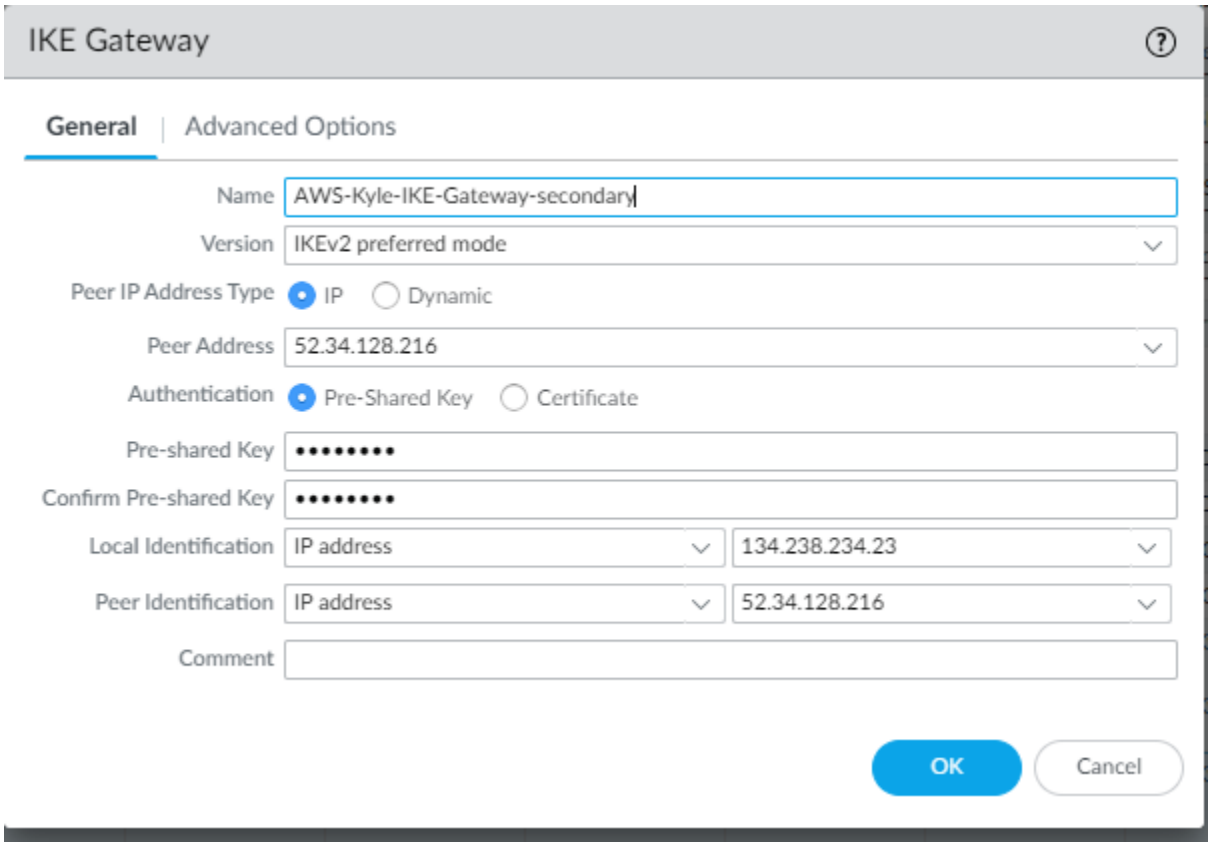
134.238.234.23 is the public IP of the Prisma Access Service Connection node
44.236.179.88 is the public IP of the AWS gateway that Prisma Access is building a Service connection to. This is the “Outside IP” of the primary tunnel in AWS. See tunnel state screenshot in the AWS config workflow



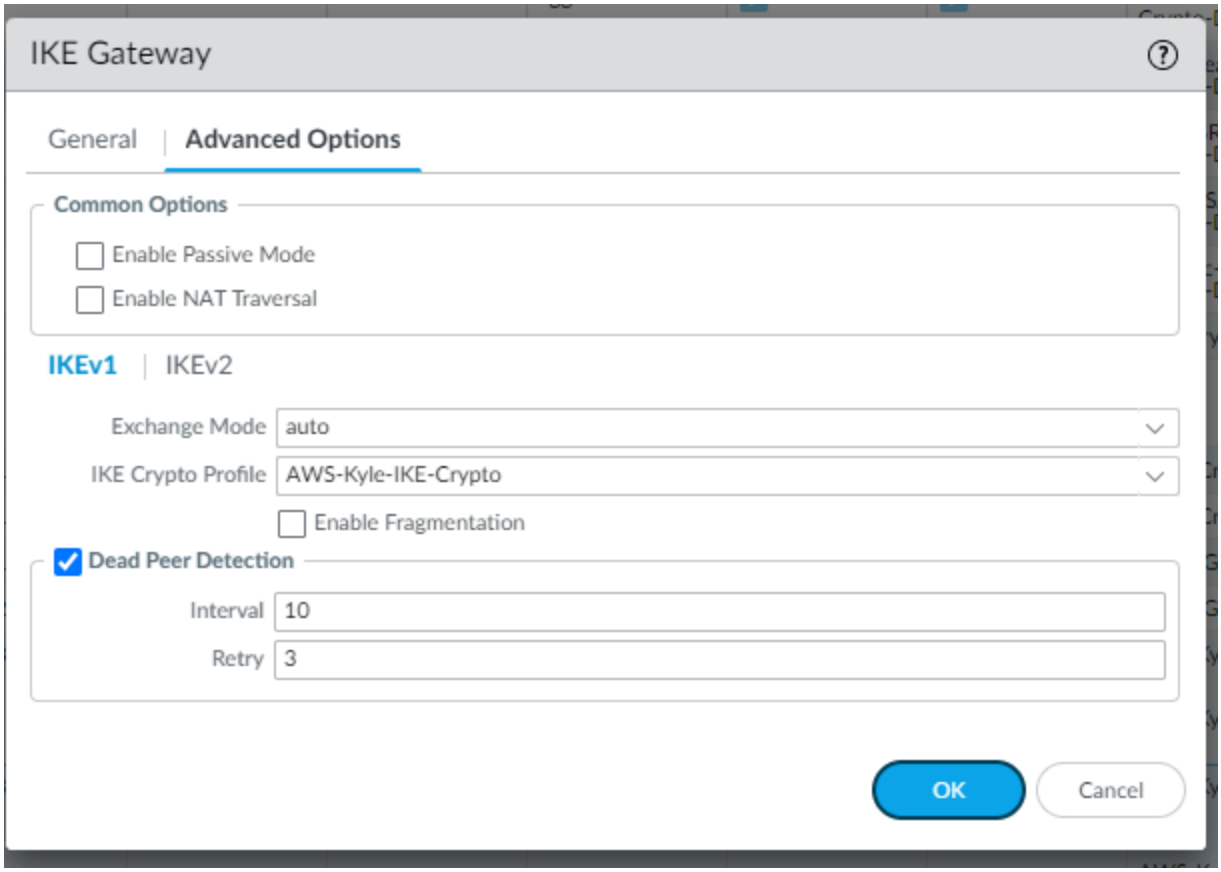
These settings match the default AWS settings and work (as of December 2022)

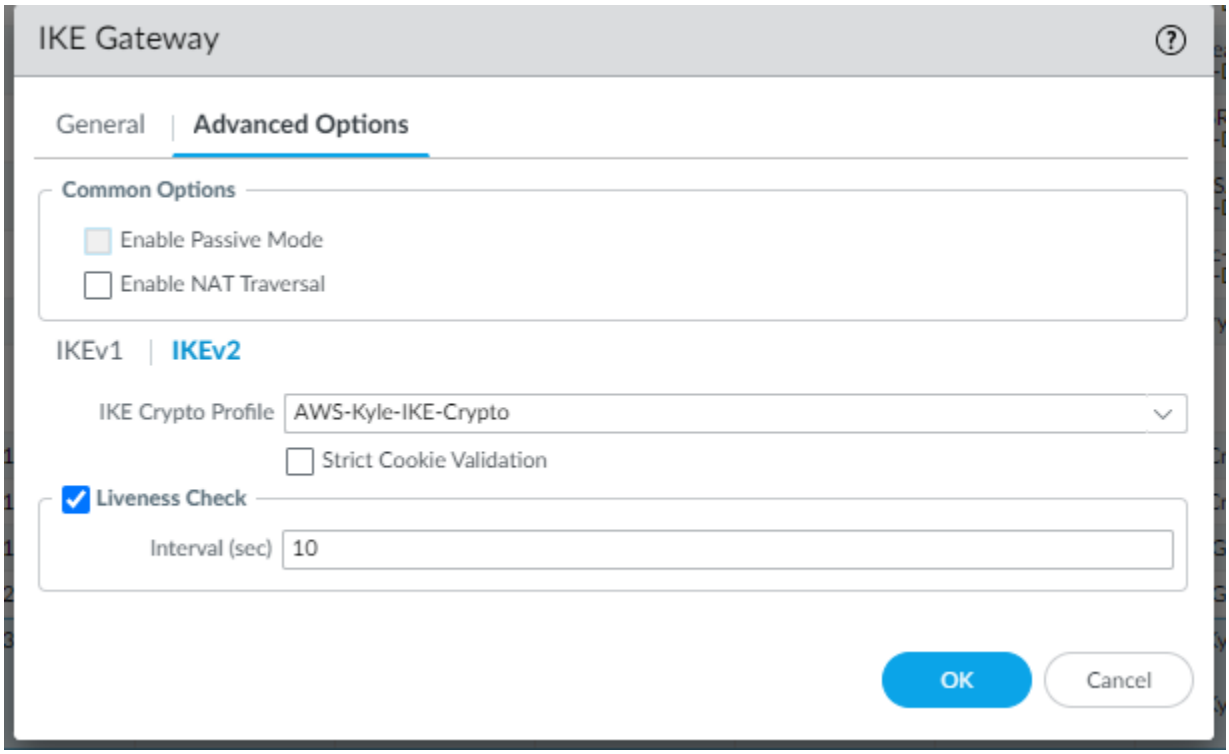


Now we have switched over to the IKEv2 section and just setting the crypto profile and liveness check interval

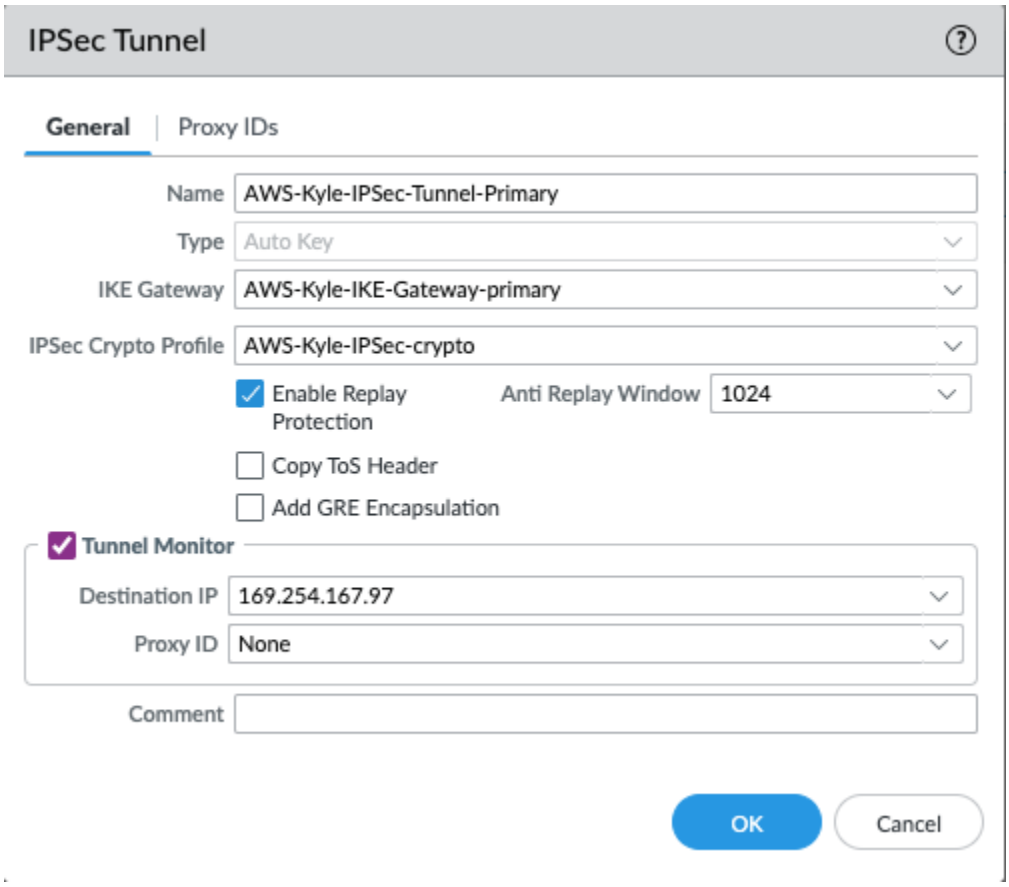


Okay now we repeat everything for the secondary WAN connection / secondary tunnel in Prisma Access. The only thing that changes in the peer Address and Peer ID IP both become the “Outside IP” of the secondary tunnel from AWS (again see the previous screenshot of the ‘tunnel state’ from AWS)


IKE Gateway
General
Advanced Options
Common Options
Enable Passive Mode
Enable NAT Traversal
IKEv1
IKEv2
Exchange Mode
auto
IKE Crypto Profile
AWS-Kyle-IKE-Crypto
Enable Fragmentation
Dead Peer Detection
Interval
10
Retry
3
OK
Cancel


IKE Gateway
General
Advanced Options
Common Options
Enable Passive Mode
Enable NAT Traversal
IKEv1
IKEv2
IKE Crypto Profile
AWS-Kyle-IKE-Crypto
Strict Cookie Validation
Liveness Check
Interval (sec)
10
OK
Cancel

Now we will set the IPSec Tunnel configs:



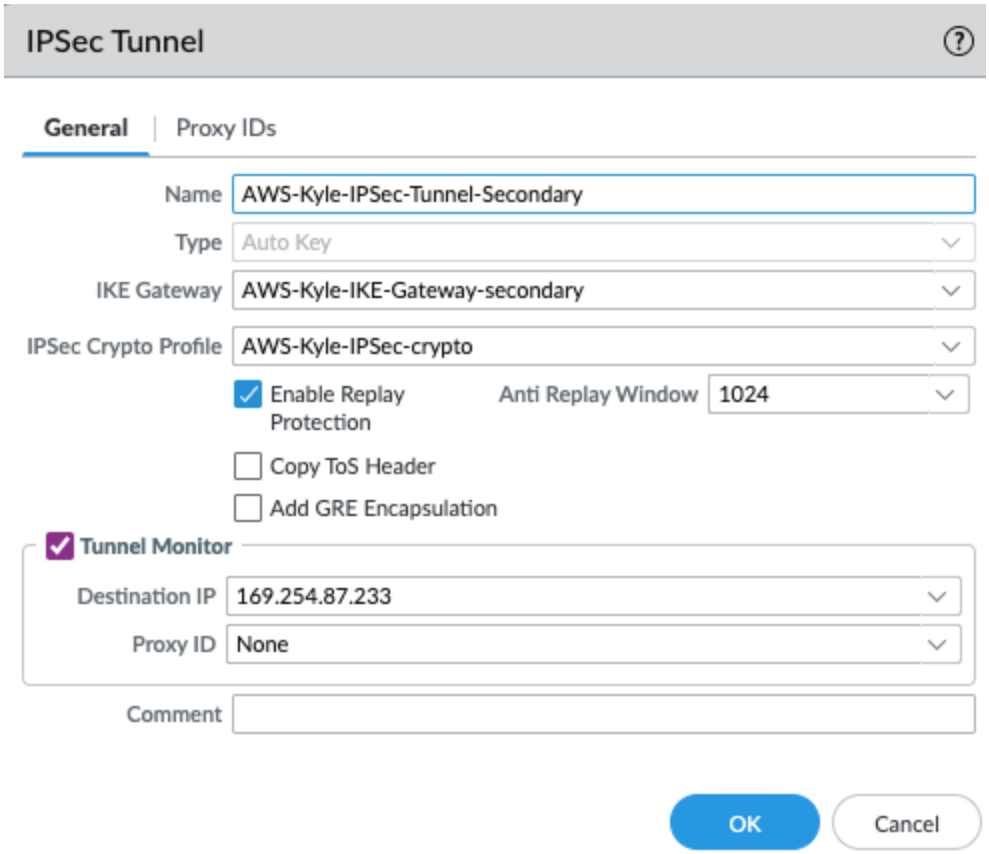
Destination IP for the Tunnel Monitor is the IP subnet of the inside interface of the AWS side of the tunnel, with the last octet incremented by 1:

*we do not set any proxy IDs

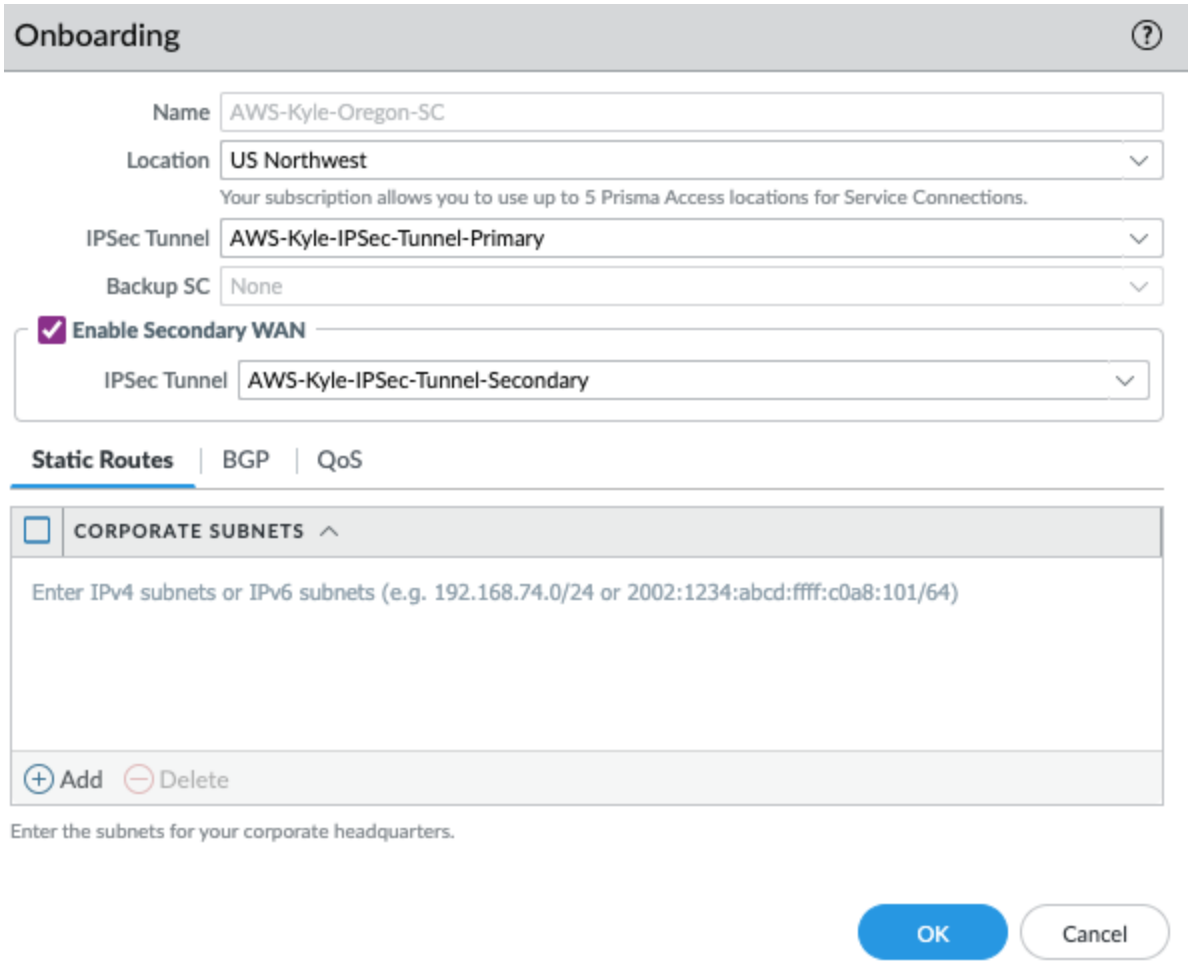
Tunnel state

Tunnel number	Outside IP address	Inside IPv4 CIDR
Tunnel 1	44.236.179.88	169.254.167.96/30
Tunnel 2	52.34.128.216	169.254.87.232/30

Repeat for the secondary tunnel:

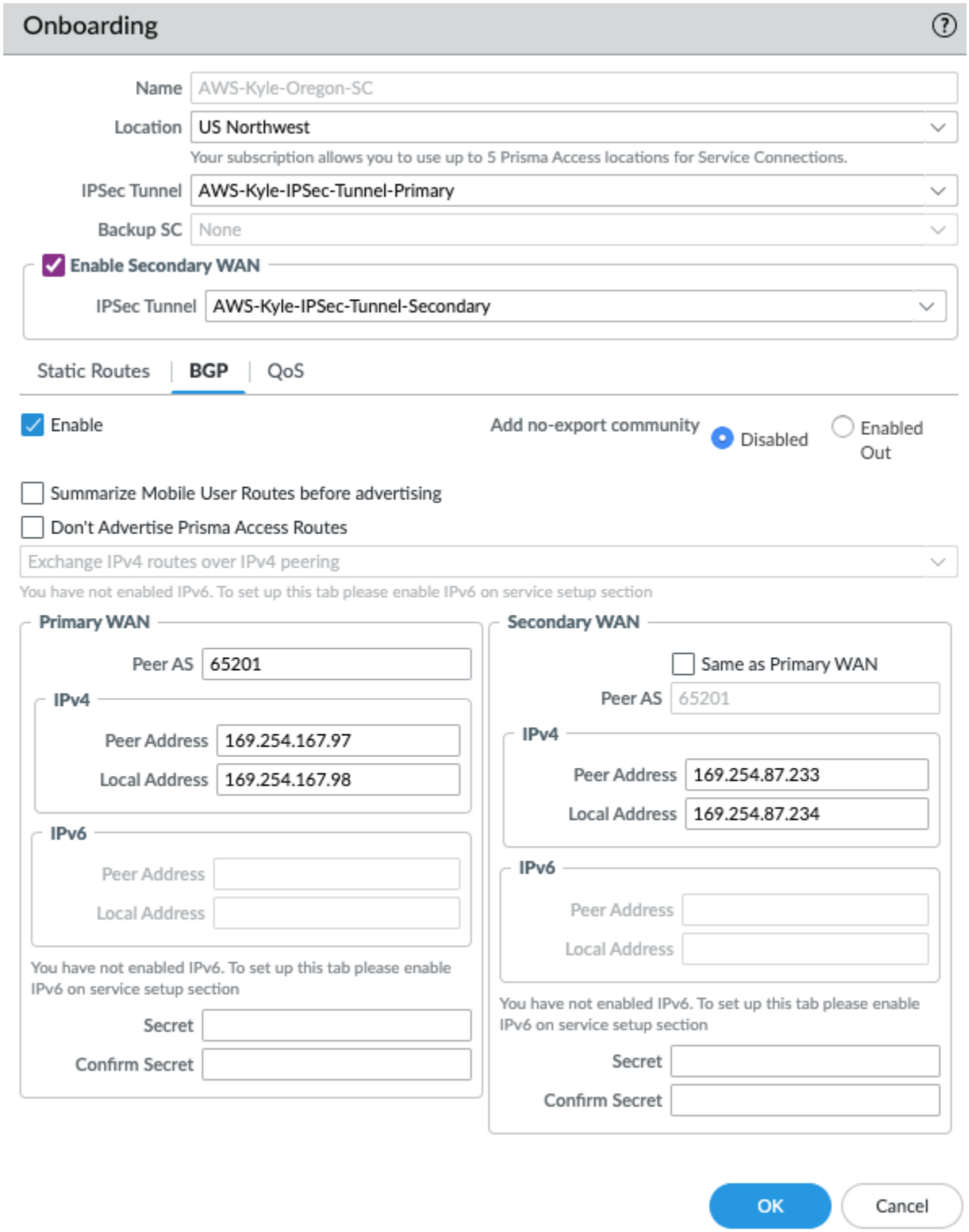


Now return to Panorama > Cloud Services > Configuration > Service Connection



Ensure this looks the same as yours

Finally it is time to set up BGP:



Peer AS is the AS set in AWS
Peer and local addresses are the first and second IPs from the inside subnet from the AWS tunnel (respectively for the primary and secondary tunnel in AWS - see below for this info again)

Tunnel state

Tunnel number	Outside IP address	Inside IPv4 CIDR
Tunnel 1	44.236.179.88	169.254.167.96/30
Tunnel 2	52.34.128.216	169.254.87.232/30

***you can also get the correct IP values by exporting the AWS tunnel configs (in palo alto v 7.0+ format) and reading through them to find these IPs

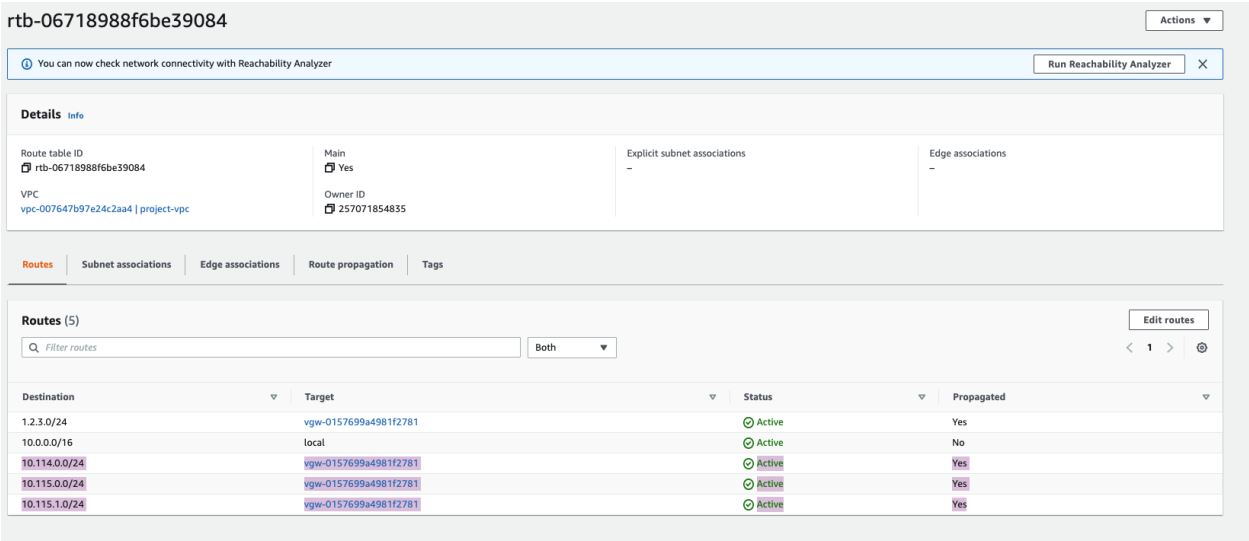
```
edit network virtual-router default protocol bgp
 set router-id 134.238.234.23
 set install-route yes
 set enable yes
 set local-as 65000
  edit peer-group AmazonBGP
   edit peer amazon-vpn-05faa21cefde38a54-0
    set peer-as 65201
    set connection-options keep-alive-interval 10
    set connection-options hold-time 30
    set enable yes
    set local-address ip 169.254.167.98/30
    set local-address interface tunnel.1
    set peer-address ip 169.254.167.97
    top
```

Time to finish:

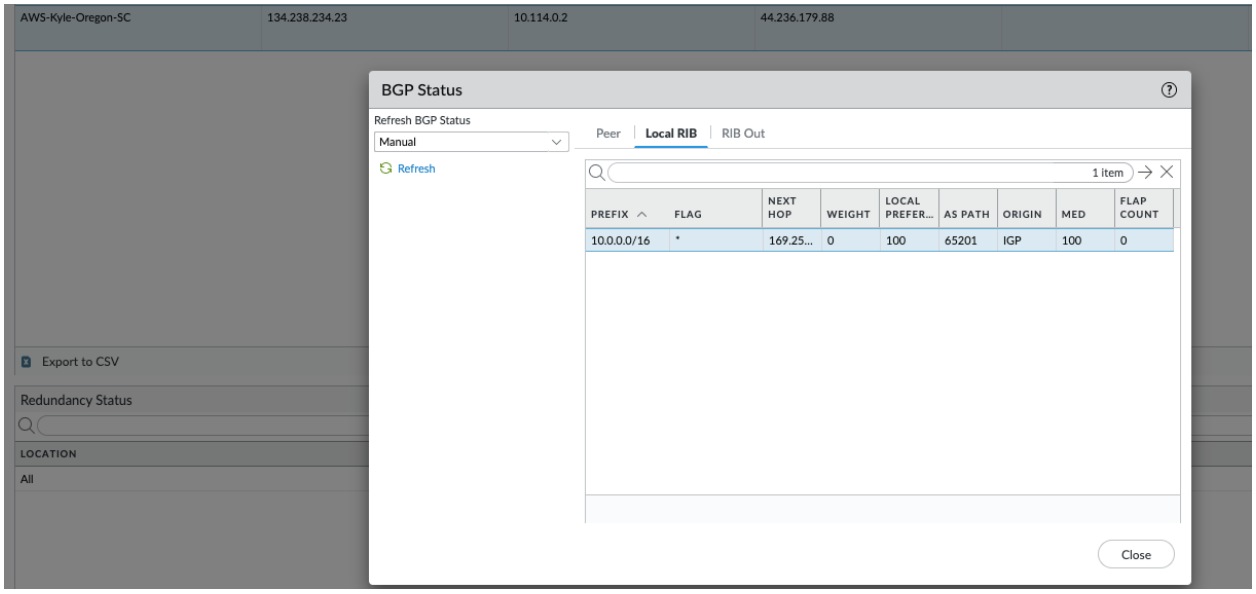
Commit and push
Wait about 15 minutes
Check the tunnel and BGP status in Prisma Access and AWS

Results:

AWS Route table for this VPC has learned the Prisma Access routes via BGP



Prisma Access has learned the /16 in this AWS VPC via BGP a well:



*in my case, 10.114 is the prisma access infra subnet, 10.115 is the global mobile user IP pool, and 1.2.3.0/24 was a dummy static route on a Remote Network connection attached to the same Prisma Access tenant