

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



The screenshot shows a table with the following columns: NAME, SERVICE IP ADDRESS, USER-ID AGENT ADDRESS, LOCAL IP ADDRESS, STATIC SUBNET, EBGP ROUTER, and BRANCH AS AND ROUTER. There are two rows of data.

NAME	SERVICE IP ADDRESS	USER-ID AGENT ADDRESS	LOCAL IP ADDRESS	STATIC SUBNET	EBGP ROUTER	BRANCH AS AND ROUTER
LasVegasDC	130.41.195.210	10.114.0.1	3.226.186.46	0.0.0.0/0	169.254.117.230	65201 169.254.117.229 BGP Status
AWS-Kyle-Oregon-SC	134.238.234.23	10.114.0.2	44.236.179.88		169.254.167.98 169.254.87.234	65201 169.254.167.97 BGP Status 65201 169.254.87.233 BGP Status

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

Service Setup | Mobile Users - GlobalProtect | Mobile Users - Explicit Proxy | Remote Networks | **Service Connection** | Traffic Steering | [_ \(help@prismacloud.com\)](#)

Onboarding 2 items → X

CONNECTION NAME	LOCATION	Primary IPsec Tunnel		Secondary IPsec Tunnel		CORPORATE SUBNETS	Remote Network	
		IPSEC TUNNEL	PEER IP ADDRESS	IPSEC TUNNEL	PEER IP ADDRESS		BGP PRIMARY	BGP SECONDARY
<input type="checkbox"/> LasVegasDC	US West	AWSEastVPN	3.226.186.46	(disabled)	(disabled)	0.0.0.0/0	Enabled Peer AS: 65201 Peer IP: 169.254.117.229 Local IP: 169.254.117.230	(disabled)
<input type="checkbox"/> AWS-Kyle-Oregon-SC	US Northwest	AWS-Kyle-IPsec-Tunnel-Primary	44.236.179.88	AWS-Kyle-IPsec-Tunnel-Secondary	52.34.128.216		Enabled Peer AS: 65201 Peer IP: 169.254.167.97 Local IP: 169.254.167.98	Enabled Peer AS: 65201 Peer IP: 169.254.87.233 Local IP: 169.254.87.234

Monitor Profile ?

Name

Action Wait Recover Fail Over

Interval (sec)

Threshold

IKE Gateway ?

General | Advanced Options

Name

Version

Peer IP Address Type IP Dynamic

Peer Address

Authentication Pre-Shared Key Certificate

Pre-shared Key

Confirm Pre-shared Key

Local Identification

Peer Identification

Comment

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

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

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

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

The screenshot shows the 'IKE Gateway' configuration window. The 'General' tab is active, displaying the following settings:

- Name: AWS-Kyle-IKE-Gateway-secondary
- Version: IKEv2 preferred mode
- Peer IP Address Type: IP, Dynamic
- Peer Address: 52.34.128.216
- Authentication: Pre-Shared Key, Certificate
- Pre-shared Key: [Masked]
- Confirm Pre-shared Key: [Masked]
- Local Identification: IP address (selected), 134.238.234.23
- Peer Identification: IP address (selected), 52.34.128.216
- Comment: [Empty]

Buttons: OK, Cancel

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:

IKE Crypto Profile:

Enable Fragmentation

Dead Peer Detection

Interval:

Retry:

IKE Gateway ?

General | **Advanced Options**

Common Options

- Enable Passive Mode
- Enable NAT Traversal

IKEv1 | **IKEv2**

IKE Crypto Profile:

Strict Cookie Validation

Liveness Check

Interval (sec):

Now we will set the IPSec Tunnel configs:

IPSec Tunnel ⓘ

General | Proxy IDs

Name:

Type:

IKE Gateway:

IPSec Crypto Profile:

Enable Replay Protection Anti Replay Window:

Copy ToS Header

Add GRE Encapsulation

Tunnel Monitor

Destination IP:

Proxy ID:

Comment:

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:

IPSec Tunnel



General | Proxy IDs

Name

Type

IKE Gateway

IPSec Crypto Profile

Enable Replay Protection Anti Replay Window

Copy ToS Header

Add GRE Encapsulation

Tunnel Monitor

Destination IP

Proxy ID

Comment

OK

Cancel

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

Onboarding



Name

Location

Your subscription allows you to use up to 5 Prisma Access locations for Service Connections.

IPSec Tunnel

Backup SC

Enable Secondary WAN

IPSec Tunnel

Static Routes | BGP | QoS

CORPORATE SUBNETS ^

Enter IPv4 subnets or IPv6 subnets (e.g. 192.168.74.0/24 or 2002:1234:abcd:ffff:c0a8:101/64)

Enter the subnets for your corporate headquarters.

OK

Cancel

Ensure this looks the same as yours

Finally it is time to set up BGP:

Onboarding ?

Name

Location

Your subscription allows you to use up to 5 Prisma Access locations for Service Connections.

IPSec Tunnel

Backup SC

Enable Secondary WAN

IPSec Tunnel

Static Routes | **BGP** | QoS

Enable Add no-export community Disabled Enabled Out

Summarize Mobile User Routes before advertising

Don't Advertise Prisma Access Routes

Exchange IPv4 routes over IPv4 peering

You have not enabled IPv6. To set up this tab please enable IPv6 on service setup section

Primary WAN

Peer AS

IPv4

Peer Address

Local Address

IPv6

Peer Address

Local Address

You have not enabled IPv6. To set up this tab please enable IPv6 on service setup section

Secret

Confirm Secret

Secondary WAN

Same as Primary WAN

Peer AS

IPv4

Peer Address

Local Address

IPv6

Peer Address

Local Address

You have not enabled IPv6. To set up this tab please enable IPv6 on service setup section

Secret

Confirm Secret

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

rtb-06718988f6be39084 Actions

ⓘ You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer ✕

Details [Info](#)

Route table ID rtb-06718988f6be39084	Main Yes	Explicit subnet associations -	Edge associations -
VPC vpc-007647b97e24c2aa4 project-vpc	Owner ID 257071854835		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (5) Edit routes

Both < 1 > ⊗

Destination	Target	Status	Propagated
1.2.3.0/24	vgw-0157699a4981f2781	Active	Yes
10.0.0.0/16	local	Active	No
10.114.0.0/24	vgw-0157699a4981f2781	Active	Yes
10.115.0.0/24	vgw-0157699a4981f2781	Active	Yes
10.115.1.0/24	vgw-0157699a4981f2781	Active	Yes

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

AWS-Kyle-Oregon-SC 134.238.234.23 10.114.0.2 44.236.179.88

BGP Status ?

Refresh BGP Status
Manual Peer **Local RIB** RIB Out

Refresh

1 item → ✕

PREFIX	FLAG	NEXT HOP	WEIGHT	LOCAL PREFER...	AS PATH	ORIGIN	MED	FLAP COUNT
10.0.0.0/16	*	169.25...	0	100	65201	IGP	100	0

Export to CSV

Redundancy Status

LOCATION

All

Close

*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