Contents

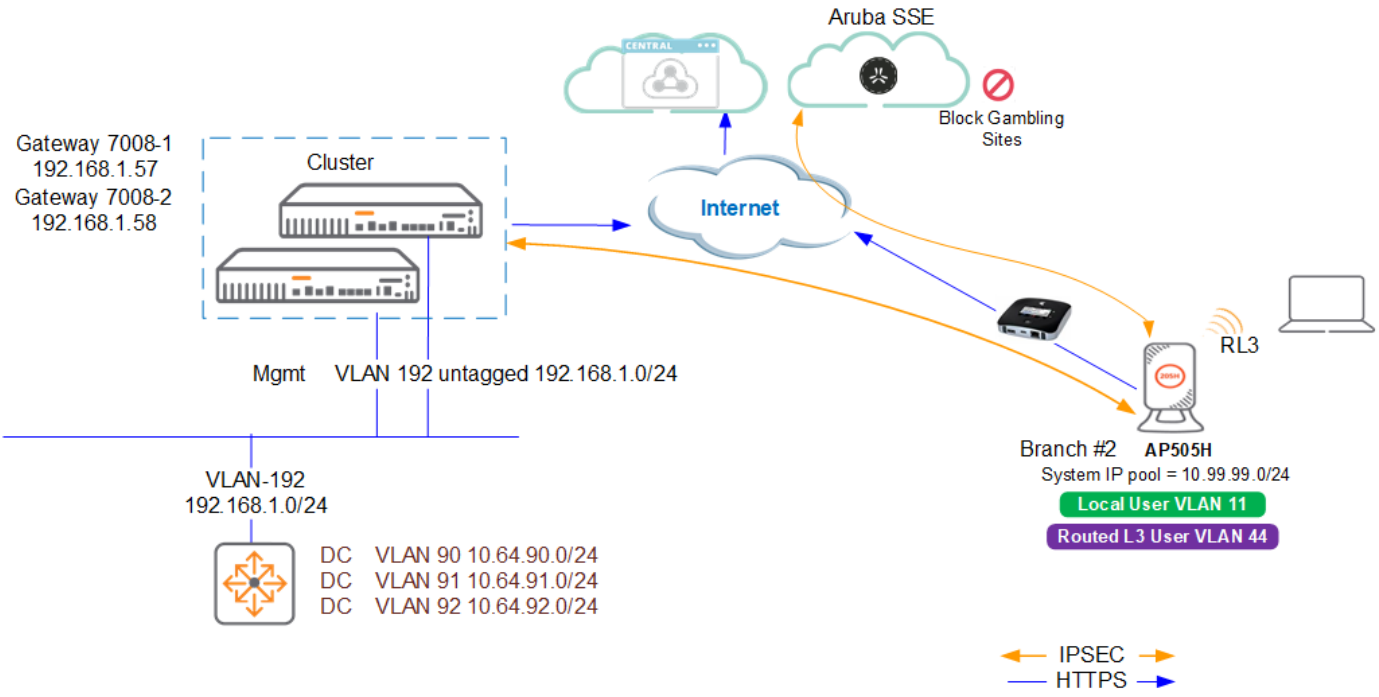
1.1 Revision History

DATE	VERSION	EDITOR	CHANGES
11 Apr 2024	0.1	Ariya Parsamanesh	Initial creation
06 May 2024	0.2	Ariya Parsamanesh	Added User Testing section

2 Microbranch and Aruba SSE Integration

The Microbranch (MB) solution can be seamlessly integrated with leading cloud security providers such as Aruba SSE (Secure Service Edge), Netskope, Zscaler, and Prisma through Aruba Central's "Cloud Connect" service. This enables the establishment of a secure connection between the MB AP and one or multiple cloud-hosted enforcement control points.

Aruba SSE serves as an Internet onramp, acting as the next hop for Internet-bound traffic from MB. Through Aruba Central's cloud connect service, IPSEC tunnels are automatically orchestrated to connect MB to Aruba Public SSE. These IPSEC tunnels utilize the Internet Key Exchange (IKE) protocol, enabling traversal of NAT boundaries and leveraging IKEv2 for authentication, while minimizing overhead. This setup ensures a secure and efficient connection between the Microbranch solution and Aruba SSE, enhancing network security and compliance capabilities.



2.1 Before You Start

I am assuming you have a working Microbranch setup which means the MB access point is

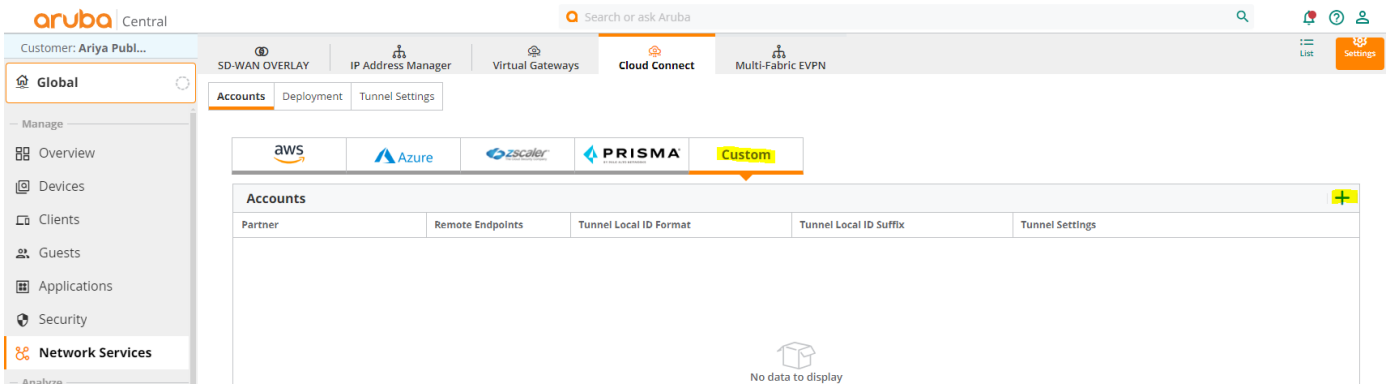
- added and subscribed with Advance AP foundation license in Aruba Central. MB APs require Advanced AP license for any integration through Cloud Connect service
- part of the AOS10 microbranch group and is configured with firmware 10.5.1.1, 10.6.0.0 or later.

2.2 Aruba Central Configuration

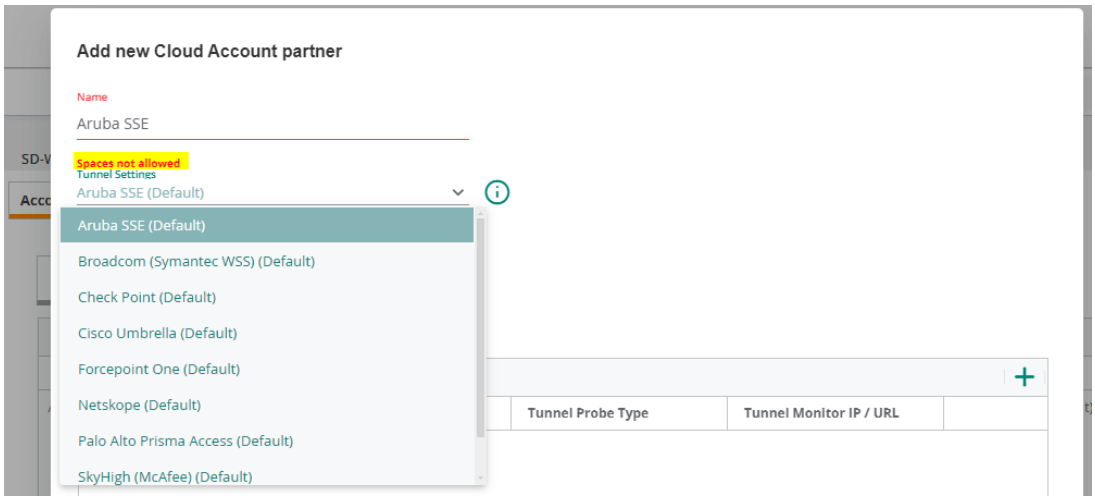
The main tasks are:

1. Setting up IPSEC Tunnels between microbranch APs and Aruba SSE cloud primary and secondary nodes
2. Enabling Orchestrating Aruba SSE tunnels to select groups.
3. Configuring policy based routing to redirect user's web traffic for inspection by Aruba SSE
4. Configuring Aruba SSE for IPSEC tunnels from the microbranch APs
5. Creating a policy for web traffic from microbranch APs.

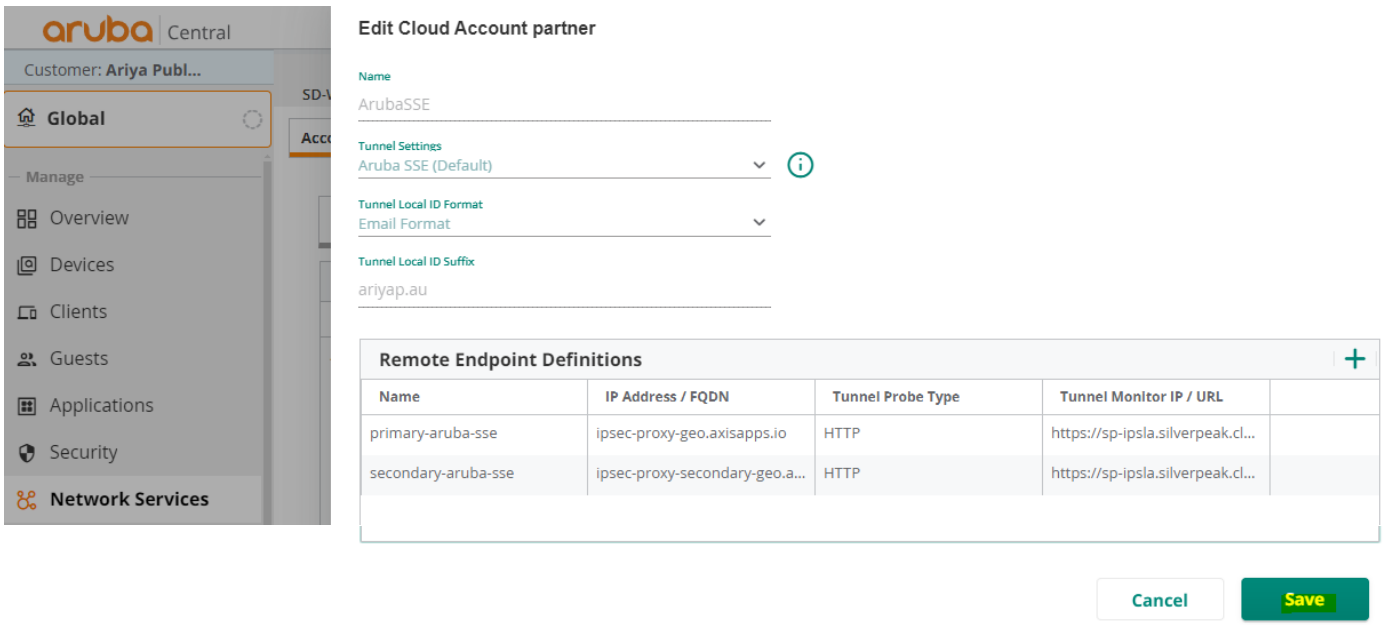
Here we'll start with by going to cloud connect selecting the "Settings" and using the customer tab.



Here we'll add an entry and note that as indicated, there should not be any spaces in the name. Also note that there are other custom SSE partners as listed.

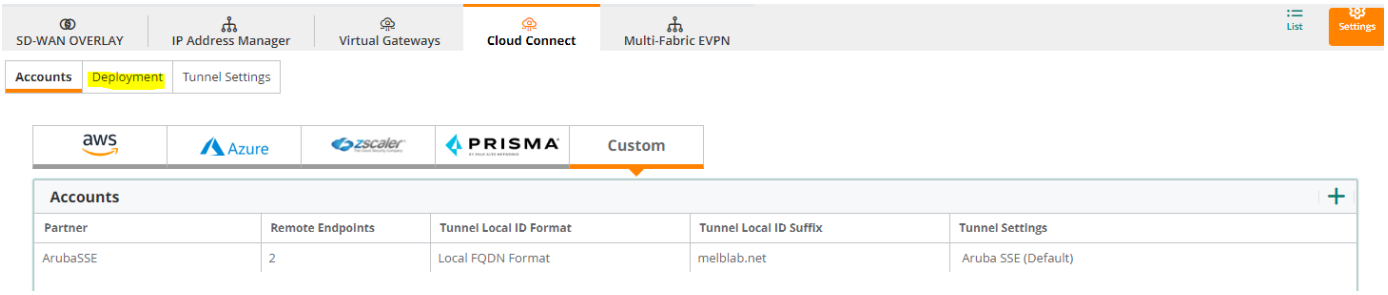


We'll use "email format" as the Tunnel Local ID Format, as shown below. You can use any local suffix which should match on the Aruba SSE tunnel configuration. Here I am using "ariyap.au"

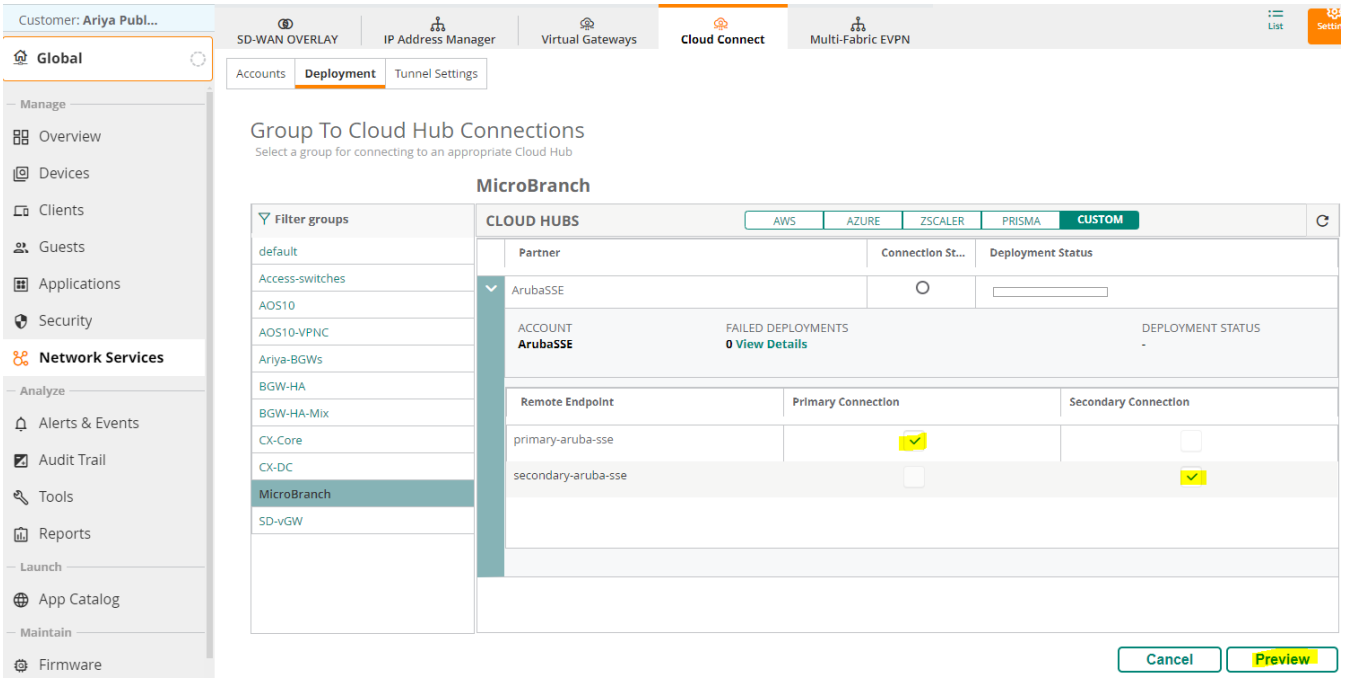


Primary-Aruba-SSE	Secondary-Aruba-SSE
FQDN: ipsec-proxy-geo.axisapps.io Tunnel Probe Type: HTTP Tunnel Monitor IP/URL: https://sp-ipsla.silverpeak.cloud	FQDN: ipsec-proxy-secondary-geo.axisapps.io Tunnel Probe Type: HTTP Tunnel Monitor IP/URL: https://sp-ipsla.silverpeak.cloud

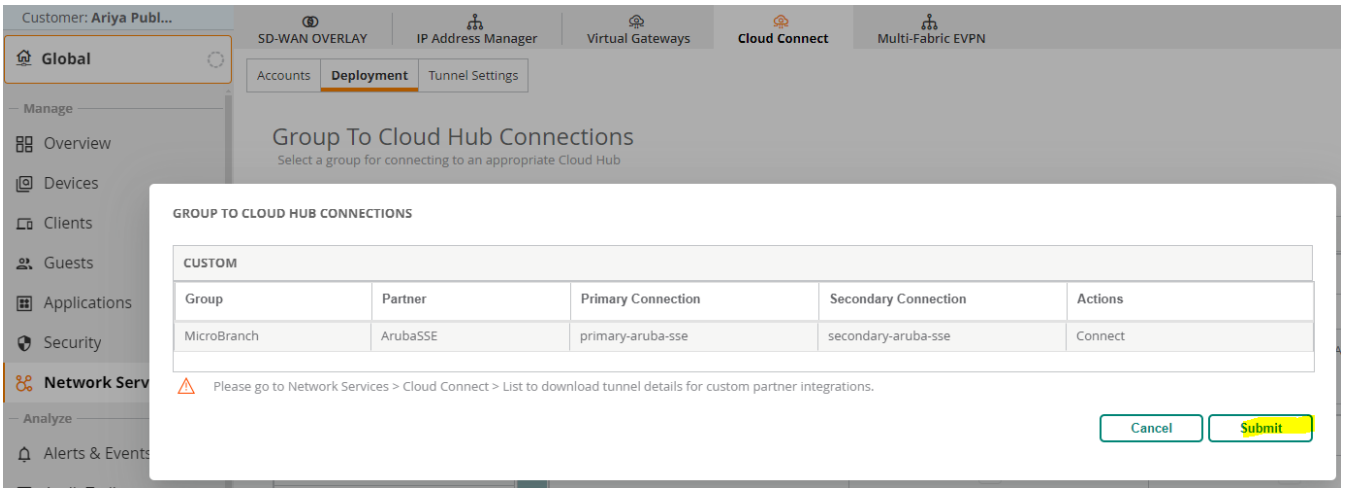
Click on the Save button.



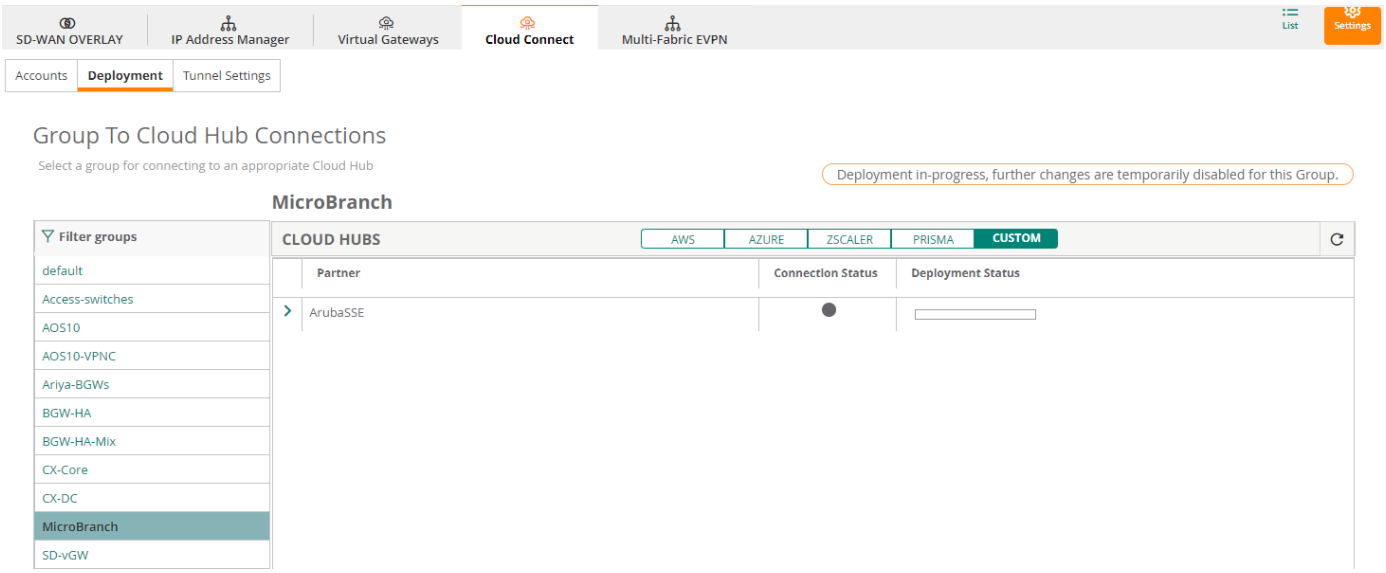
Wait for a minute or two. Then go to the deployment tab and choose your group, in my case its “MicroBranch” group and select the primary and secondary connections.

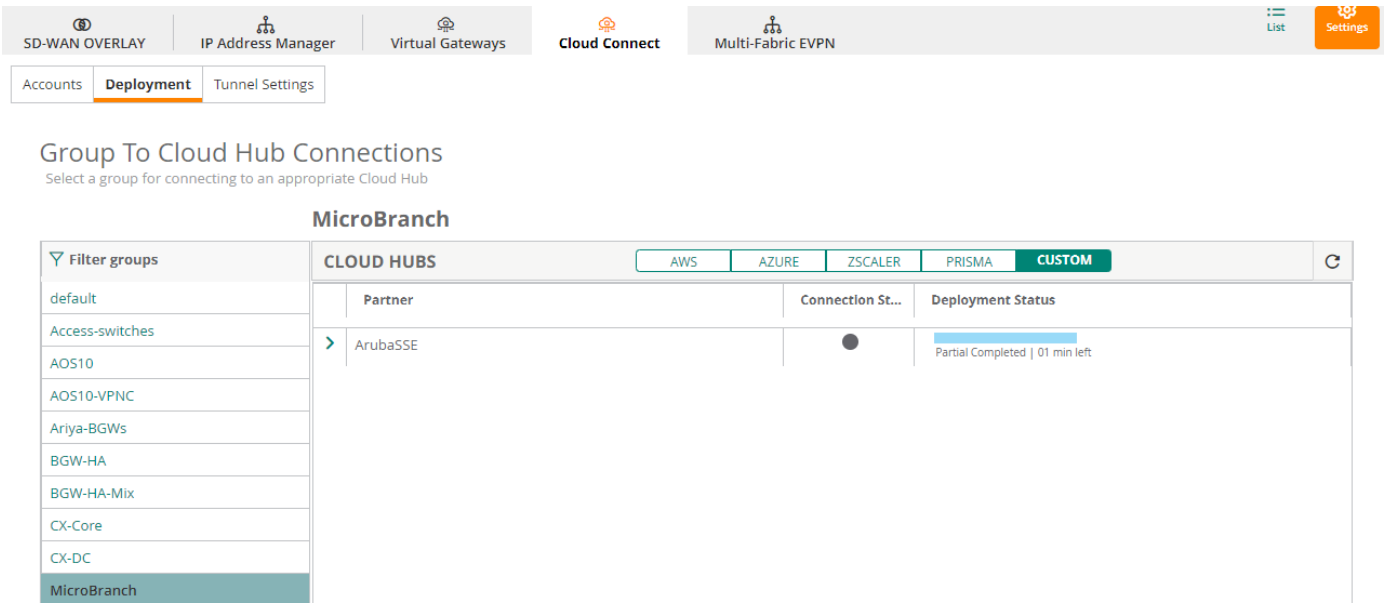


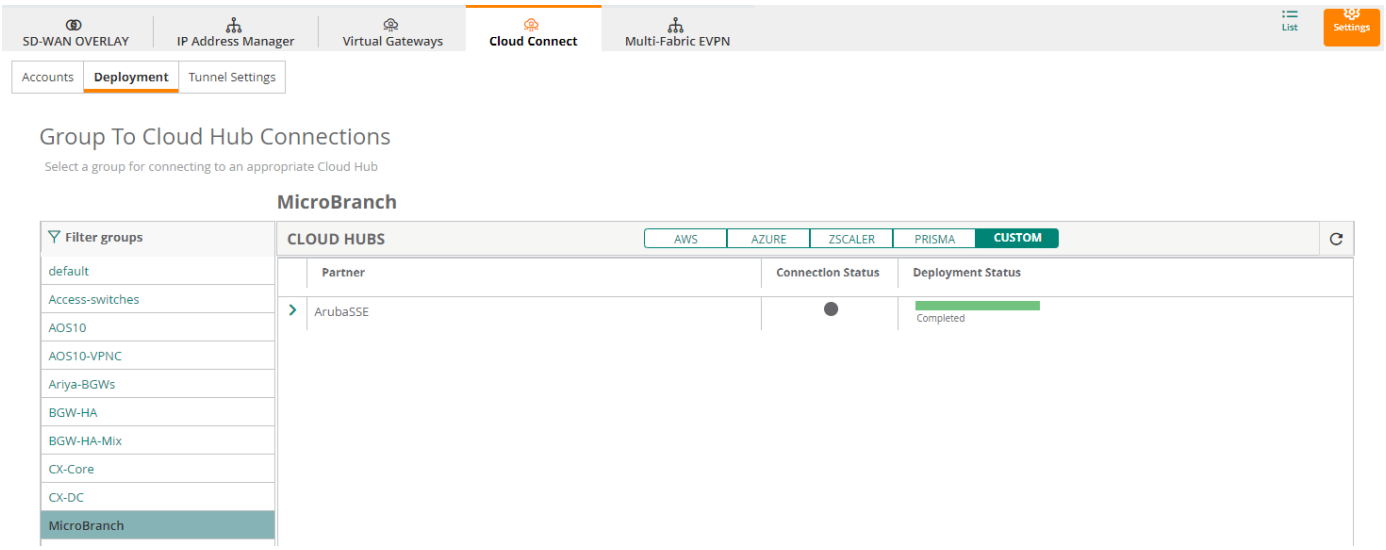
Click on Preview and submit it.



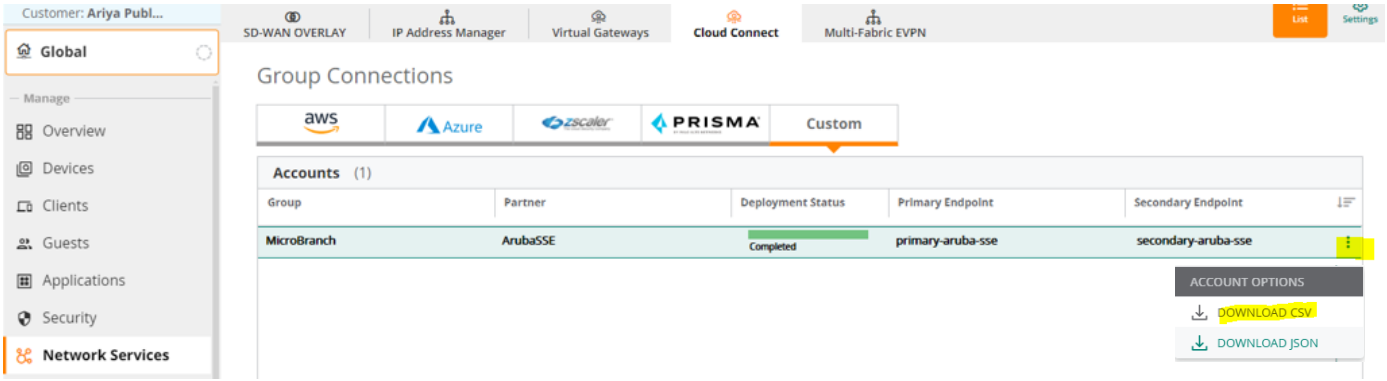
Again wait a few minutes.







Once it is completed as shown above and while still in “cloud connect” click on the “list” icon, go to the “Custom” tab and download the account options. I have used CSV format.



You need to open the CSV file and the information that we need are “Source Identity” and PSKs that corresponds to your microbranch APs. You need to copy it. Take particular note of the uplink name as the Identity and PSKs are specific to the uplinks. We’ll use these when we configure Aruba SSE to allow these IPSEC tunnels.

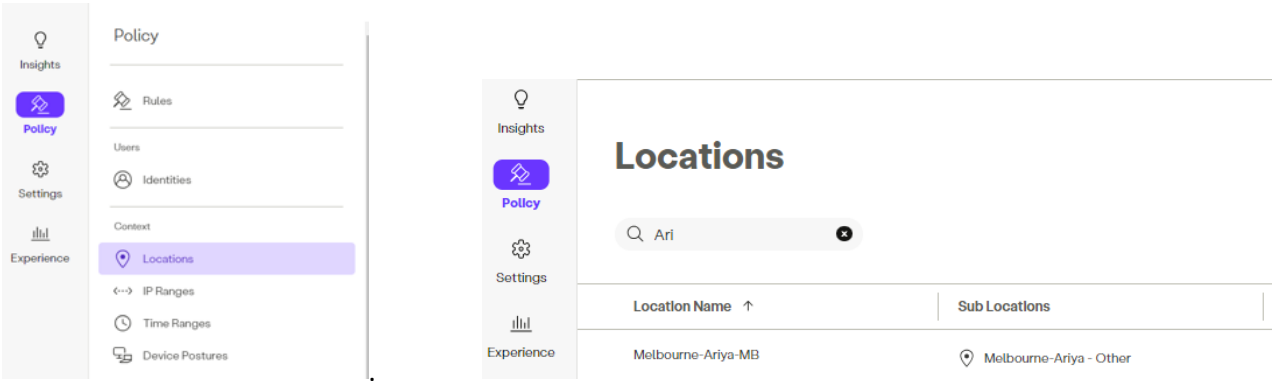
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Aruba Devic	Source Identity	PSK	Source IP Ad	Uplink Name	Encryptio	Maximum	Enabled	Tunnel Na	Primary P	Primary P	Failover P	Failover P	Site Name	
2					E0-Uplink	AES256	100	TRUE	MicroBran	primary-a	ipsec-pro	secondary	ipsec-pro	MicroBranch1	
3					LTE-Uplink	AES256	100	TRUE	MicroBran	primary-a	ipsec-pro	secondary	ipsec-pro	MicroBranch1	
4					E0-Uplink	AES256	100	TRUE	MicroBran	primary-a	ipsec-pro	secondary	ipsec-pro	MicroBranch2	
5					LTE-Uplink	AES256	100	TRUE	MicroBran	primary-a	ipsec-pro	secondary	ipsec-pro	MicroBranch2	

That’s all you need to configure on the Aruba Central side for integrating it with Aruba SSE.

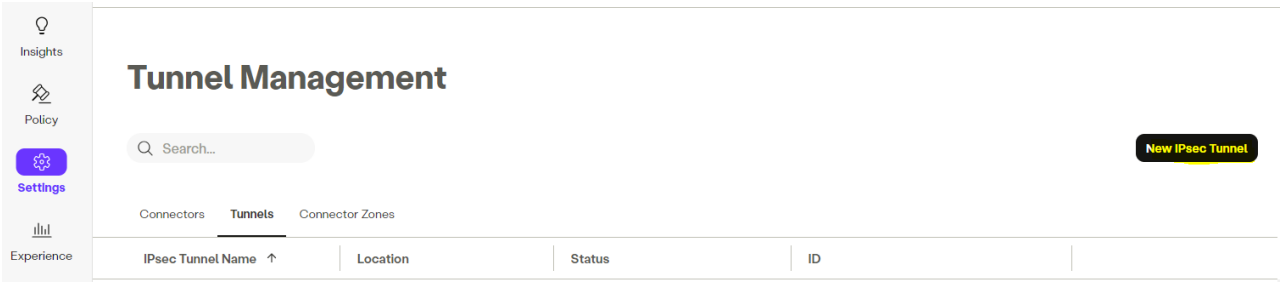
2.3 Aruba SSE Configuration

Here we’ll cover the bare minimum configuration that is needed for this solution to work. You need to login to the Aruba SSE portal https://manage.axissecurity.com/

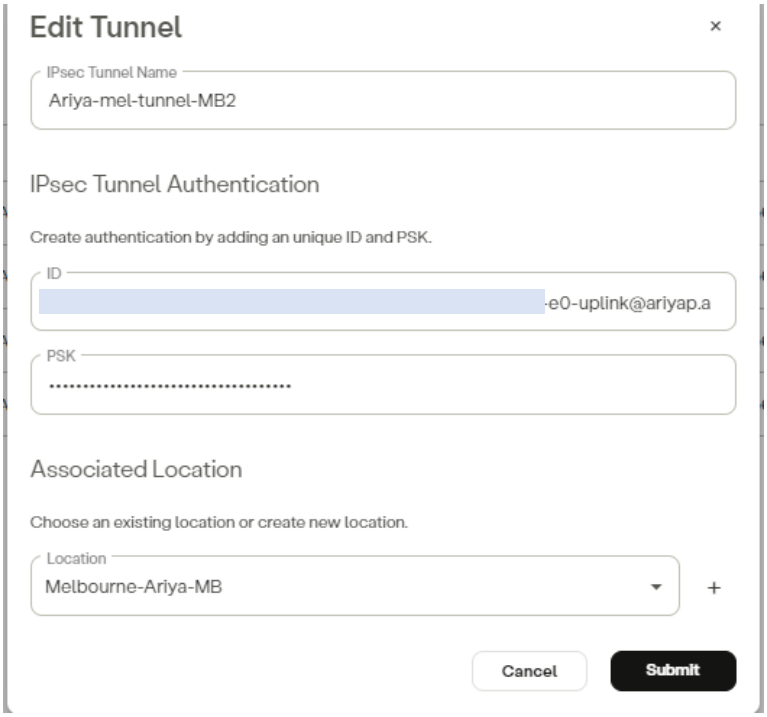
Then under Policy > Locations , add a new location, in my case its Melbourne-Ariya-MB



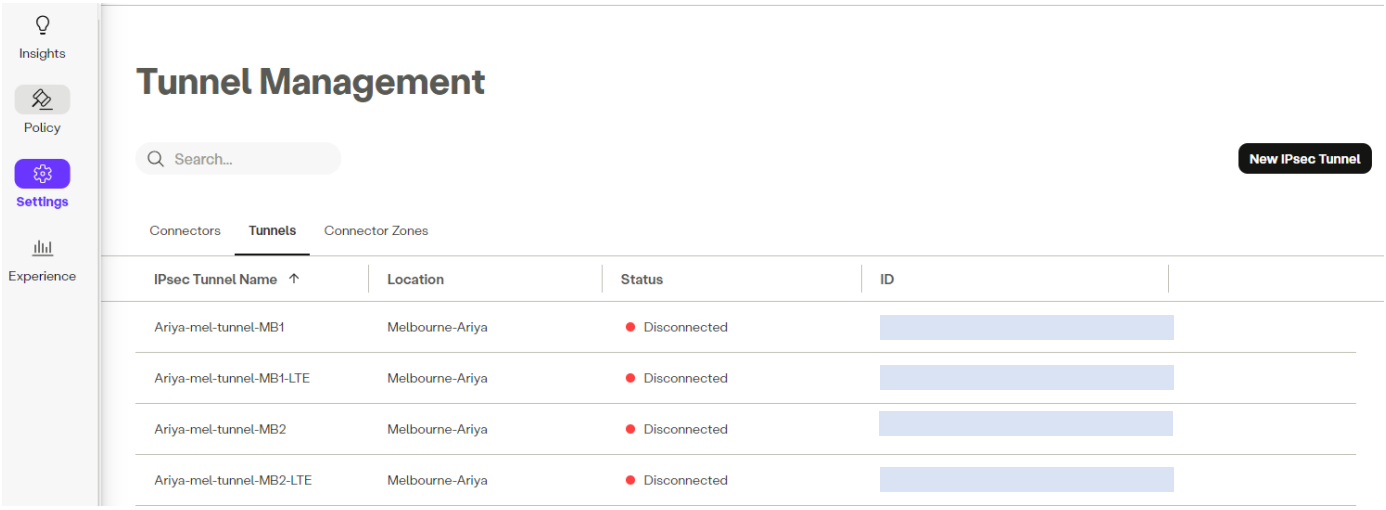
Then you should go to Settings > Connectors > Tunnels and click “New IPsec Tunnel”.



I have started with the IPSEC tunnel for MicroBranch2. Note that I have copied the “Source Identity” and PSKs for my microbranch2 for E0 uplink (from the CSV file) and pasted it in the ID and PSK section here, and finally associate it with the location that we created earlier which in my case is “Melbourne-Ariya”.



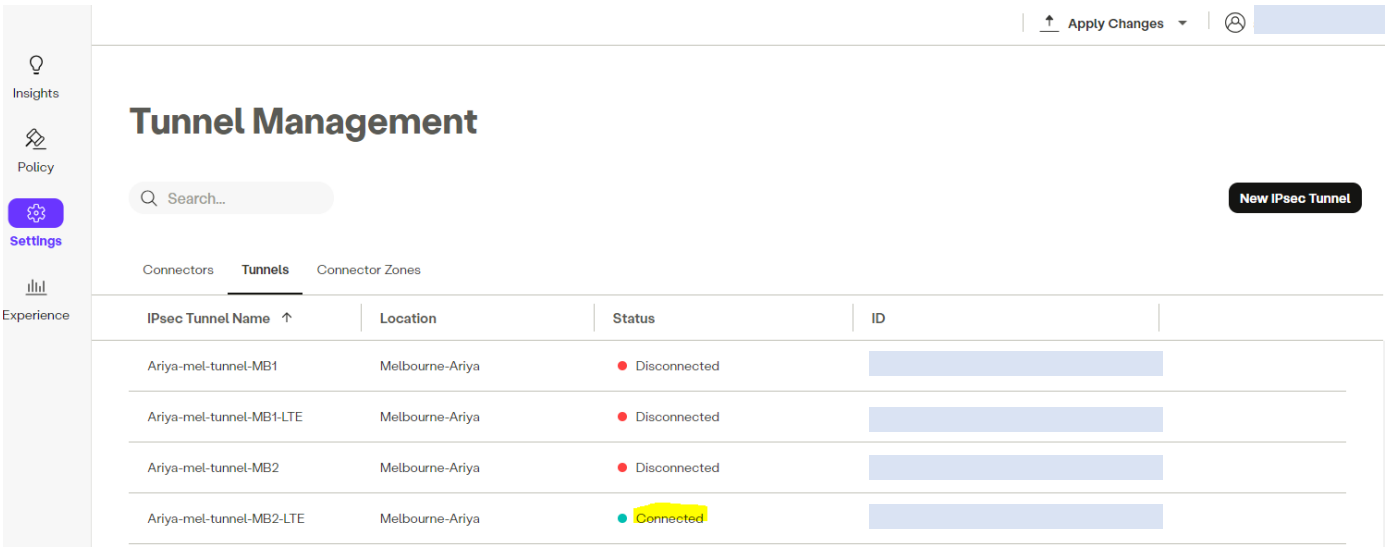
I also added the new Tunnels for LTE uplink for my microbranch APs. Here is my list of tunnels.



I have 2x MB APs and each have 2 uplinks (E0 and LTE), that is why I need 4x tunnels to be configured in Aruba SSE dashboard. Once you configure the tunnels here, then its matter of time for the IPSEC tunnels to be established. I have connected my microbranch2 AP. Note that all of them are associated with location = “Melbourne-Ariya-MB”.

2.4 Integration Verification

At this point the devices (in my case APs) in the microbranch group should have by now established the IPSEC tunnels. First checking the Aruba SSE dashboard for tunnels.



Next we'll check the microbranch AP2's CLI and use a few show commands.

We'll start with checking the uplink of that AP.

```
MicroBranch2# sh uplink status

Uplink preemption              :enable
Uplink preemption interval     :300
Uplink health check            :enable
Uplink health check host       :pqm.arubanetworks.com
Uplink health check IP         :13.239.61.151
AP1X type:NONE
Certification type:NONE
Validate server:NONE

Dns server Table
------------------
Type     IP
----     --
static   1.0.0.1
static   1.1.1.1

Uplink Table
--------------
Type       VLAN  Backup  Uplink-id   State  Reach State  Wan Type  Prio  In Use  Interface  IP
Mask           GW           Public IP
----       ----  ------  ---------   -----  -----------  --------  ----  ------  ---------  --
Ethernet   4092  No      E0-Uplink   UP     UP           INET      0     Yes     br0.4092
192.168.2.49  255.255.255.0  192.168.2.1  1.152.107.214
Cellular   4095  Yes     LTE-Uplink  DOWN   DOWN         LTE       7     No      ppp0
0.0.0.0       0.0.0.0        0.0.0.0      0.0.0.0
Wifi-sta   4097  Yes     wifi-sta    INIT   INIT         WIFI      6     No      wuplink0
0.0.0.0       0.0.0.0        0.0.0.0      0.0.0.0

Wired Port Table
------------------
Port   State   Type   Bonding(Admin/Oper/Active)
----   -----   ----   --------------------------
eth0   UP      WAN    Yes/Yes/Yes
eth1   DOWN    LAN    No/No/No
eth2   DOWN    LAN    No/No/No
eth3   DOWN    LAN    No/No/No
eth4   DOWN    LAN    No/No/No
```

```
MicroBranch2#
```

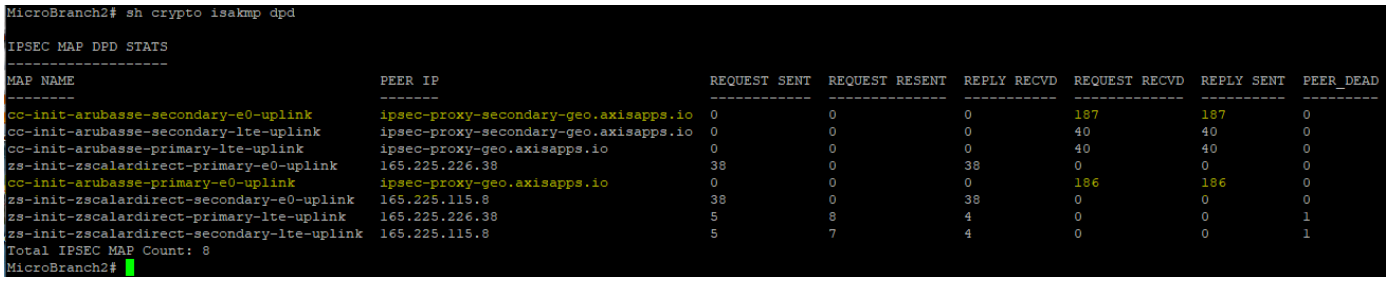
So MicroBranch2 is using E0 as uplink. Next, we'll check the IPSEC stats.

```
MicroBranch2# sh crypto ipsec stats

IPSEC STATS
-----------
MAP NAME                                    IP ADDR         DEVNAME  TX/RX PACKETS  TX/RX BYTES
TX/RX DROPS  TX/RX ERRORS
--------                                    -------         -------  -------------  -----------
-----------  ------------
gw-ipsecmap-20:4c:03:0a:b9:e0-e0-uplink     203.214.83.128  tun0     157/156        16576/17214
0/0          0/0
gw-ipsecmap-20:4c:03:0a:b9:e0-lte-uplink    203.214.83.128           0/0            0/0
0/0          0/0
zs-init-zscalardirect-primary-e0-uplink     165.225.226.38  tun1     0/0            0/0
0/0          0/0
cc-init-arubasse-secondary-e0-uplink        54.253.209.61   tun2     0/0            0/0
0/0          0/0
cc-init-arubasse-secondary-lte-uplink       54.253.209.61            0/0            0/0
0/0          0/0
cc-init-arubasse-primary-lte-uplink         3.25.3.36                0/0            0/0
0/0          0/0
cc-init-arubasse-primary-e0-uplink          3.25.3.36       tun4     0/0            0/0
0/0          0/0
zs-init-zscalardirect-secondary-e0-uplink   165.225.115.8   tun3     0/0            0/0
0/0          0/0
zs-init-zscalardirect-primary-lte-uplink    165.225.226.38           0/0            0/0
0/0          0/0
zs-init-zscalardirect-secondary-lte-uplink  165.225.115.8            0/0            0/0
0/0          0/0
Total IPSEC Count: 10

MicroBranch2#
```

So the IPSEC tunnels for Tun2 and Tun4 are being formed. You need to check the ATA table to ensure they are in CONNECTED state.

```
MicroBranch2# sh ata endpoint status

ATA Endpoint Status
-------------------
UUID                                    IP ADDR         STATE                TUN DEV  TUN
SPI(OUT/IN)    LINK TAG    VALID TIME(s)  TUNNEL TYPE  GRE VLANs      HBT(Jiff/Missed/Sent/Rcv)
INNER IP    UP TIME(s)
----                                    -------         -----                -------  ------------
---    --------    -------------  -----------  ---------      -------------------------  ------
<changed>  203.214.83.128  SM_STATE_CONNECTED  tun0     af5b0900/c5f9e100  E0-Uplink   129314
GREoIPSec    1,22,192,4094  808/0/336/305              10.99.99.4  2024-04-13 12:28:56
<changed>  203.214.83.128  SM_STATE_INIT                96bd8500/2e4ad00   LTE-Uplink  128957
GREoIPSec    1,22,192,4094  427/0/258/258              10.99.99.4  2024-04-13 12:22:58
<changed>  165.225.226.38  SM_STATE_CONNECTED  tun1     3765c59f/e9806100  E0-Uplink   28464
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:28:03
<changed>  54.253.209.61   SM_STATE_CONNECTED  tun2     c72d2e6b/dea98d00  E0-Uplink   28464
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:28:03
<changed>  54.253.209.61   SM_STATE_INIT                c7dd2ee8/66eb5b00  LTE-Uplink  28215
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:23:53
<changed>  3.25.3.36       SM_STATE_INIT                c08c124a/1a38d300  LTE-Uplink  28215
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:23:54
<changed>  3.25.3.36       SM_STATE_CONNECTED  tun4     cdedc259/cfff6f00  E0-Uplink   28464
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:28:03
<changed>  165.225.115.8   SM_STATE_CONNECTED  tun3     2fdf861d/7ea80000  E0-Uplink   28464
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:28:03
<changed>  165.225.226.38  SM_STATE_INIT                49358d6e/4cf0d200  LTE-Uplink  28322
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:25:40
<changed>  165.225.115.8   SM_STATE_INIT                25b8dc6f/b763d800  LTE-Uplink  28342
IPSEC        NULL           0/0/0/0                    10.99.99.4  2024-04-13 12:26:01
Total Endpoints Count: 10

MicroBranch2#
```

Note that I have replaced the actual UUID with <changed>. So now we have tun2 and tun4 CONNECTED. We'll check the IPSEC dead peer detection (DPD) which is enabled by default.



```
MicroBranch2# sh crypto isakmp dpd

IPSEC MAP DPD STATS
-------------------
MAP NAME                                PEER IP                                 REQUEST SENT  REQUEST RESENT  REPLY RECVD  REQUEST RECVD  REPLY SENT  PEER_DEAD
--------                                -------                                 ------------  --------------  -----------  -------------  ----------  ---------
cc-init-arubasse-secondary-e0-uplink    ipsec-proxy-secondary-geo.axisapps.io   0             0               0            187            187         0
cc-init-arubasse-secondary-lte-uplink   ipsec-proxy-secondary-geo.axisapps.io   0             0               0            40             40          0
cc-init-arubasse-primary-lte-uplink     ipsec-proxy-geo.axisapps.io             0             0               0            40             40          0
zs-init-zscalardirect-primary-e0-uplink 165.225.226.38                          38            0               38           0              0           0
cc-init-arubasse-primary-e0-uplink      ipsec-proxy-geo.axisapps.io             0             0               0            186            186         0
zs-init-zscalardirect-secondary-e0-uplink 165.225.115.8                         38            0               38           0              0           0
zs-init-zscalardirect-primary-lte-uplink 165.225.226.38                         5             8               4            0              0           1
zs-init-zscalardirect-secondary-lte-uplink 165.225.115.8                        5             7               4            0              0           1
Total IPSEC MAP Count: 8
MicroBranch2#
```

Another easy way of checking to see if the tunnels are up, is by using this command.

```
MicroBranch2# sh l3d oap tunnels

L3D OAP Tunnel Table
--------------------
Peer MAC           Map Name                                Map Id   State  GenId   Pair UUID
--------           --------                                ------   -----  -----   ---------
                   cc-init-arubasse-primary-e0-uplink      0x50007  Up     232842  <changed>
                   cc-init-arubasse-secondary-e0-uplink    0x50004  Up     232840  <changed>
20:4c:03:0a:b9:e0  gw-ipsecmap-20:4c:03:0a:b9:e0-e0-uplink 0x50001  Up     232844  <changed>
                   zs-init-zscalardirect-primary-e0-uplink 0x50003  Up     232841  <changed>
                   zs-init-zscalardirect-secondary-e0-uplink 0x50008 Up     232839  <changed>
MicroBranch2#
```

Note that microbranch APs use Overlay Agent Protocol (OAP) to connect to Aruba Central Route/Tunnel orchestrator to get the all the information about routing and tunnels that needs to be established.

```
MicroBranch2# sh l3d oap

OAP Status
Admin State:                UP
Oper State:                 UP
Master:                     127.0.0.1:50050
Channel state:              CONNECTED
Serial:                     SDSDSDSDS
MAC:                        20:4c:03:b2:75:97
Site ID (Channel):          20:4c:03:b2:75:97 (20:4c:03:b2:75:97)
Tenant ID (Channel):         ()
Tenant ID:
Tunnel Interface:           tsgw
Channel UP since:           2024-04-27 10:32:59.109
Channel Down count:         0
Learnt Routes:              3
 IPv4 checksum:             0xc2b03bb08e02d098/3, Inst_1.2.POD_TABLE.1.1.0
Advertised Routes:          2
 IPv4 checksum:             0x1010bbe165ae8585/2
Tunnels:                    5
RTB Gen ID:                 1710680410299140
PCM Gen ID IPv4 routes:     1710339637195768
Graceful Restart timer:     86400 seconds
Num of Create Channel:           1
Num of SyncReq Sent:             22
Num of SyncRep Received:         20
Num of SyncReq since last SyncRep Received:         0
Last SyncReq Sent:          2024-04-27 10:54:13.111
Last SyncRep Received:      2024-04-27 10:54:13.638
Num of FullSyncReq Sent:         0
Num of FullSyncReq Received:     4
Last RX: 2024-04-27 10:54:13.637, RX process: 2024-04-27 10:54:13.638 QSz: 0 MaxQSz: 1
Last TX: 2024-04-27 10:54:13.111, queue: 0
```

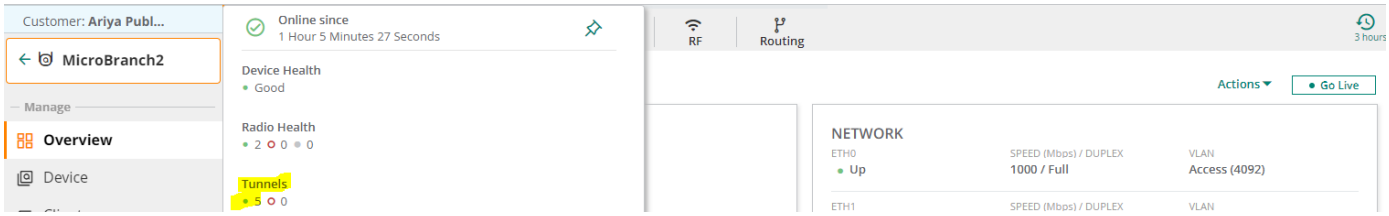
```
Peak Routes IPv4:              3 at 2024-04-27 10:34:01.486
Peak Tunnels:                  9 at 2024-04-27 10:38:27.945

MicroBranch2#
```

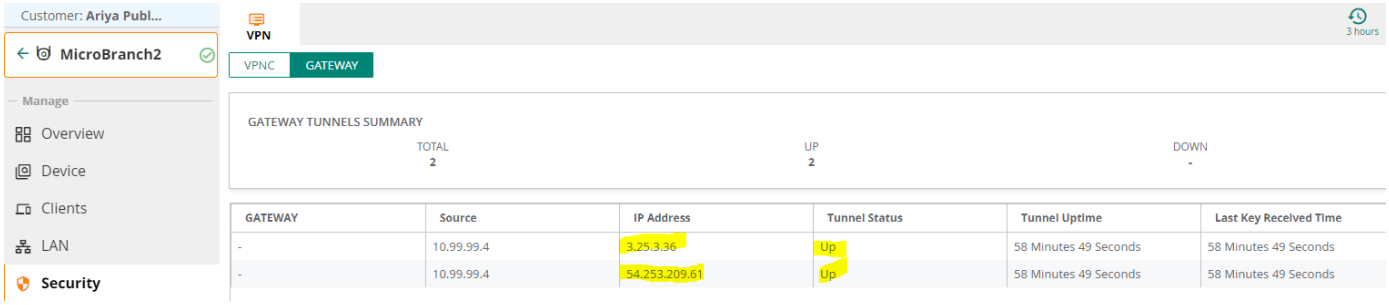
And finally I'll check the datapath session table and look for udp 4500 which is used for NAT traversal traffic. The highlighted lines are for the primary and secondary Aruba SSE nodes.

```
MicroBranch2# sh datapath session | incl 4500

13.239.61.151      192.168.2.49     17   4500  4500  0    0    48   0    dev13       4708 38e     21c50 F
203.214.83.128     192.168.2.49     17   4500  4500  0    0    0    0    local       4410 911     66404 Api
192.168.2.49       203.214.83.128   17   4500  4500  0    0    48   0    local       4410 912     661c8 pi
165.225.226.38     192.168.2.49     17   4500  4500  0    0    0    0    local       37   1       70    F
192.168.2.49       54.253.209.61    17   4500  4500  0    0    0    0    local       47b5 170     a2e0  FC
192.168.2.49       165.225.226.38   17   4500  4500  0    0    0    1    local       37   1       70    FC
54.253.209.61      192.168.2.49     17   4500  4500  0    0    0    1    local       47b5 170     a220  F
192.168.2.49       165.225.115.8    17   4500  4500  0    0    0    1    local       134  2       e0    FC
192.168.2.49       3.25.3.36        17   4500  4500  0    0    0    1    local       47b4 16d     a190  FC
192.168.2.49       13.239.61.151    17   4500  4500  0    0    0    1    dev13       4708 0       0     FYC
165.225.115.8      192.168.2.49     17   4500  4500  0    0    16   0    local       134  2       e0    F
3.25.3.36          192.168.2.49     17   4500  4500  0    0    0    0    local       47b4 16d     a0d0  F

MicroBranch2#
```

After some time you should be able to the Aruba SSE tunnels in corresponding Site topology view

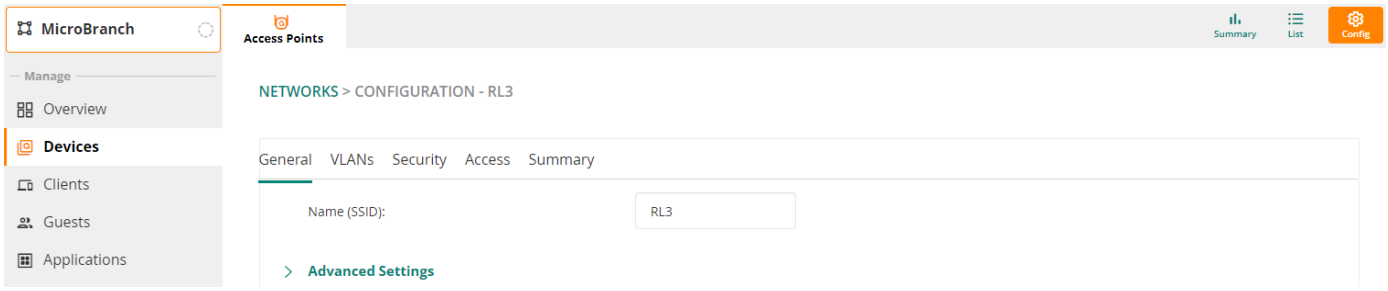
You should also be able to see the tunnels as well.

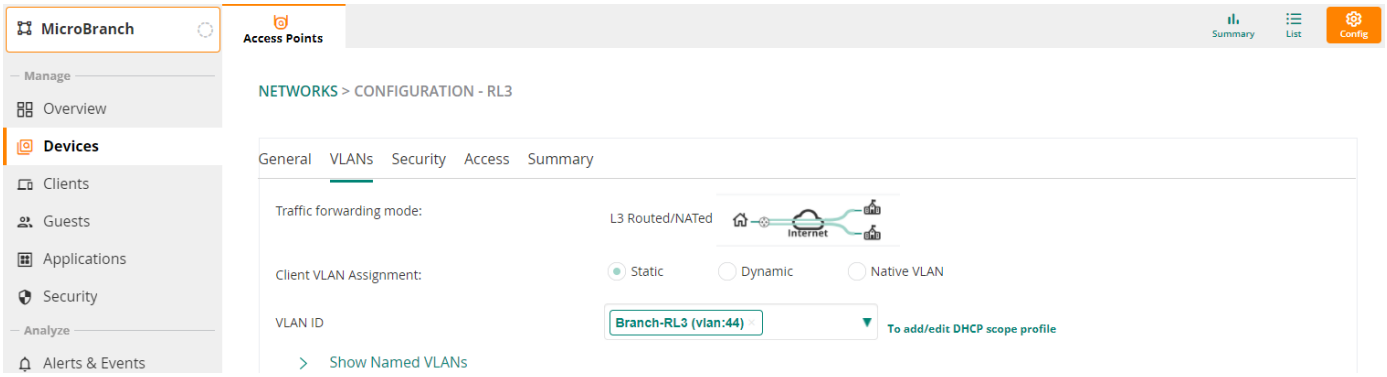


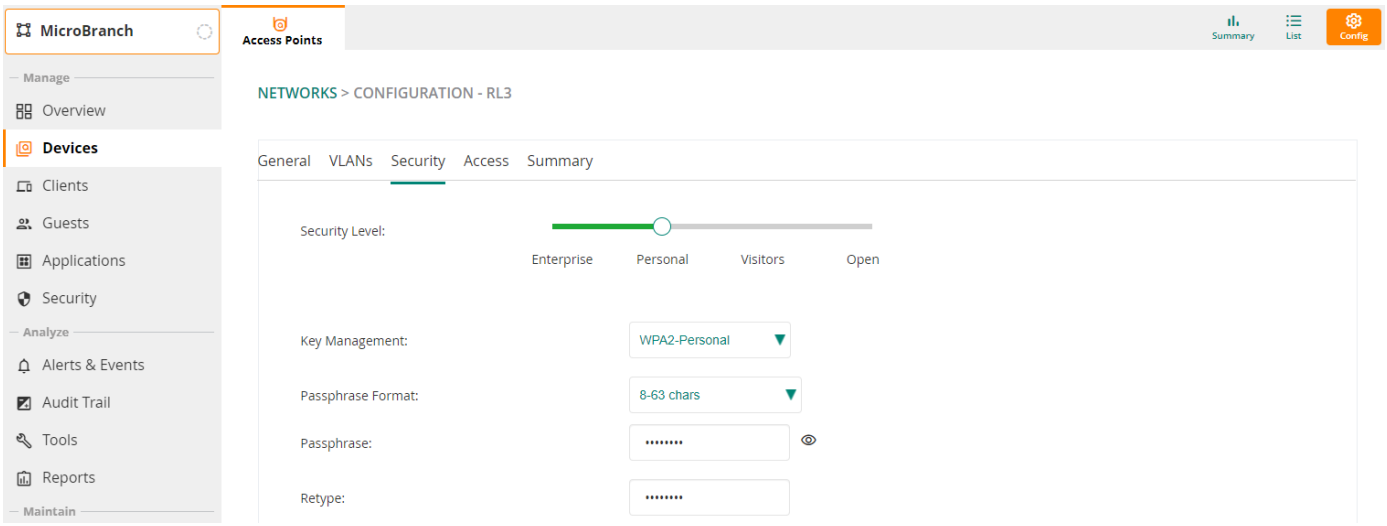


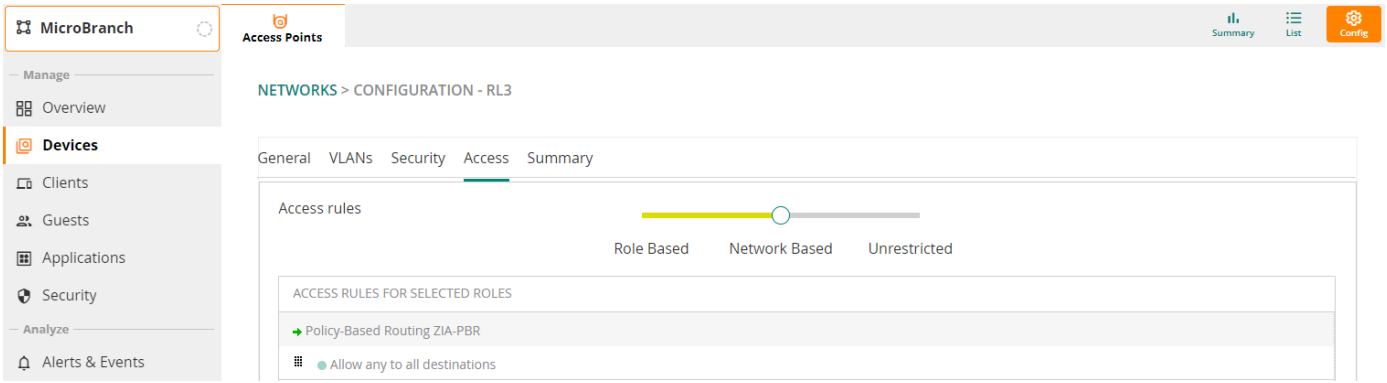
2.5 WLAN Configuration

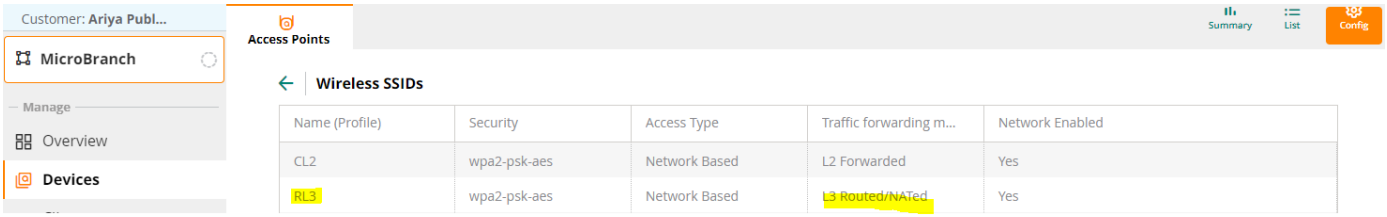
Here we have configured a RL3 SSID which has a default user role of RL3 which we can use in our testing.









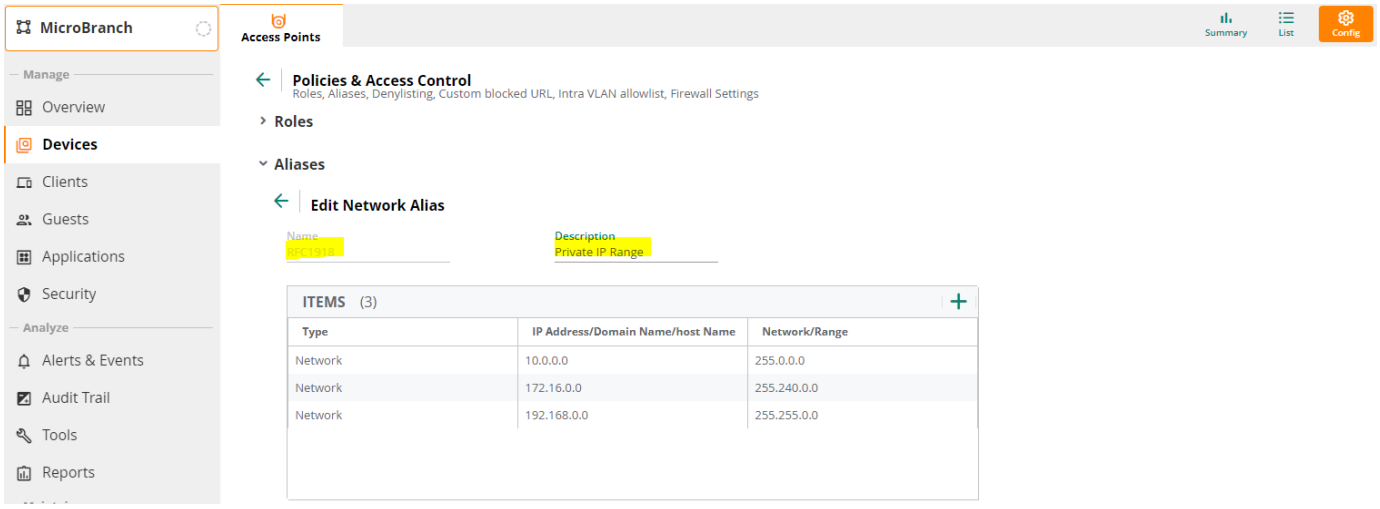


Now we need a mechanism to selectively redirect user's HTTP/HTTPS traffic to Aruba SSE, for inspection and policy compliance. These are the users that will be connecting to RL3 wlan. We'll use Policy Base Routing (PBR) for it.

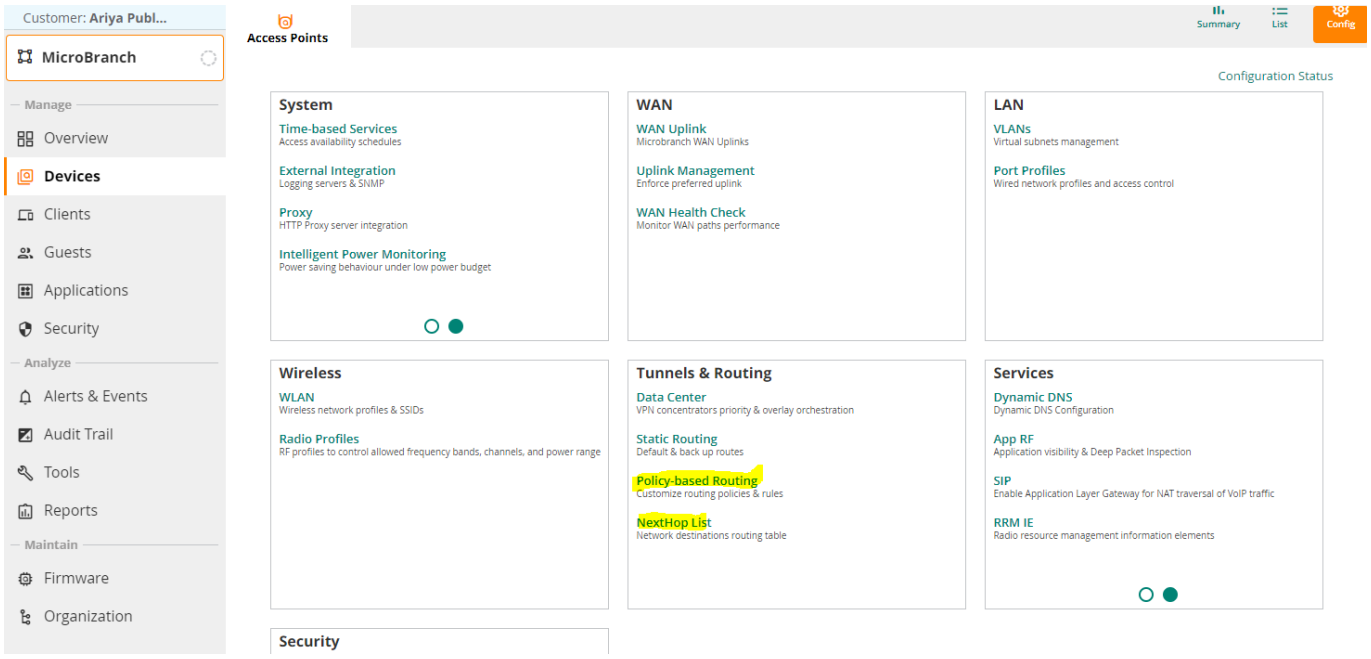
2.6 Policy Based Routing Configuration

Here we'll configure PBR and associate it with a user role. Here we want the Internet traffic to be policy routed through the Aruba SSE tunnels.

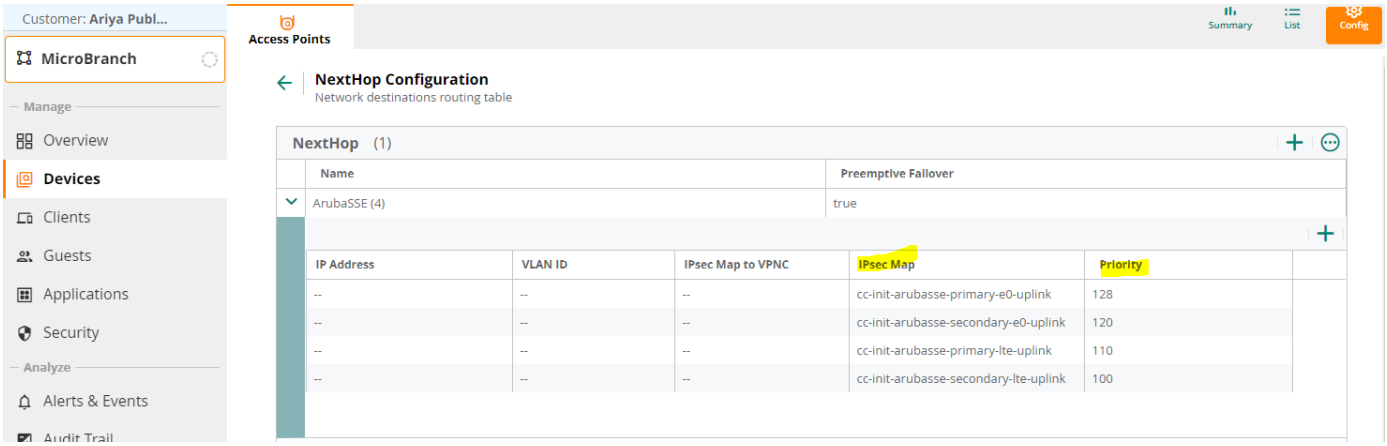
1. Create a RFC1918 alias to group all the private IP subnets.



2. Create a Next Hop List.

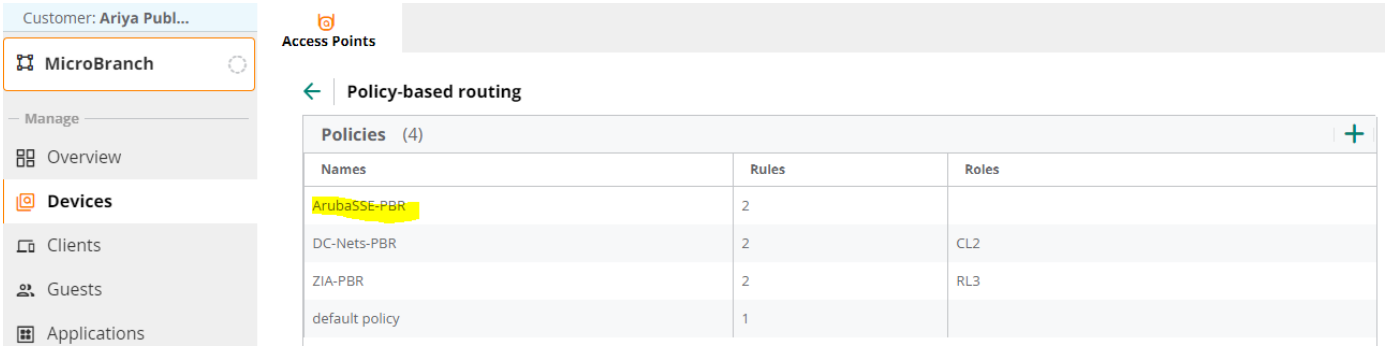


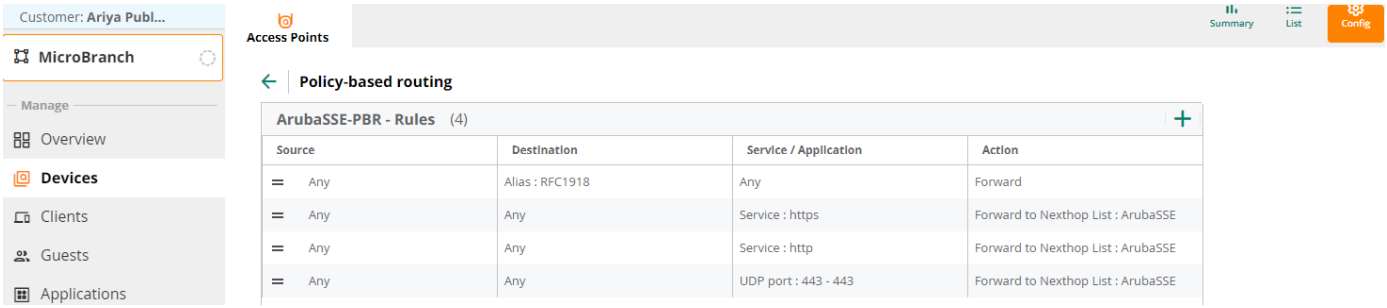
First we'll add all the IPSEC tunnels to the NextHop list so we can then use it in our PBR. NextHop list makes it easier for cases where you have 2 or more IPSEC tunnels that we can forward traffic. In our case we have 2x IPSEC tunnels for each of the uplinks (eth0 and LTE). Note that highest priority is given to Aruba SSE primary IPSEC tunnel through Eth0 uplink followed by Aruba SSE secondary IPSEC tunnel and so on.



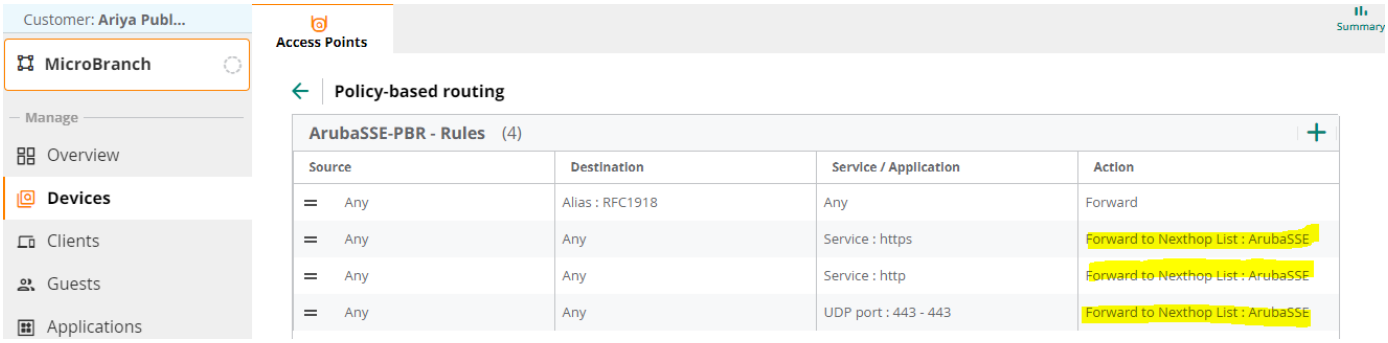
3. Create a PBR policy.

Now we'll make use of the NextHop list in our PBR as shown below.

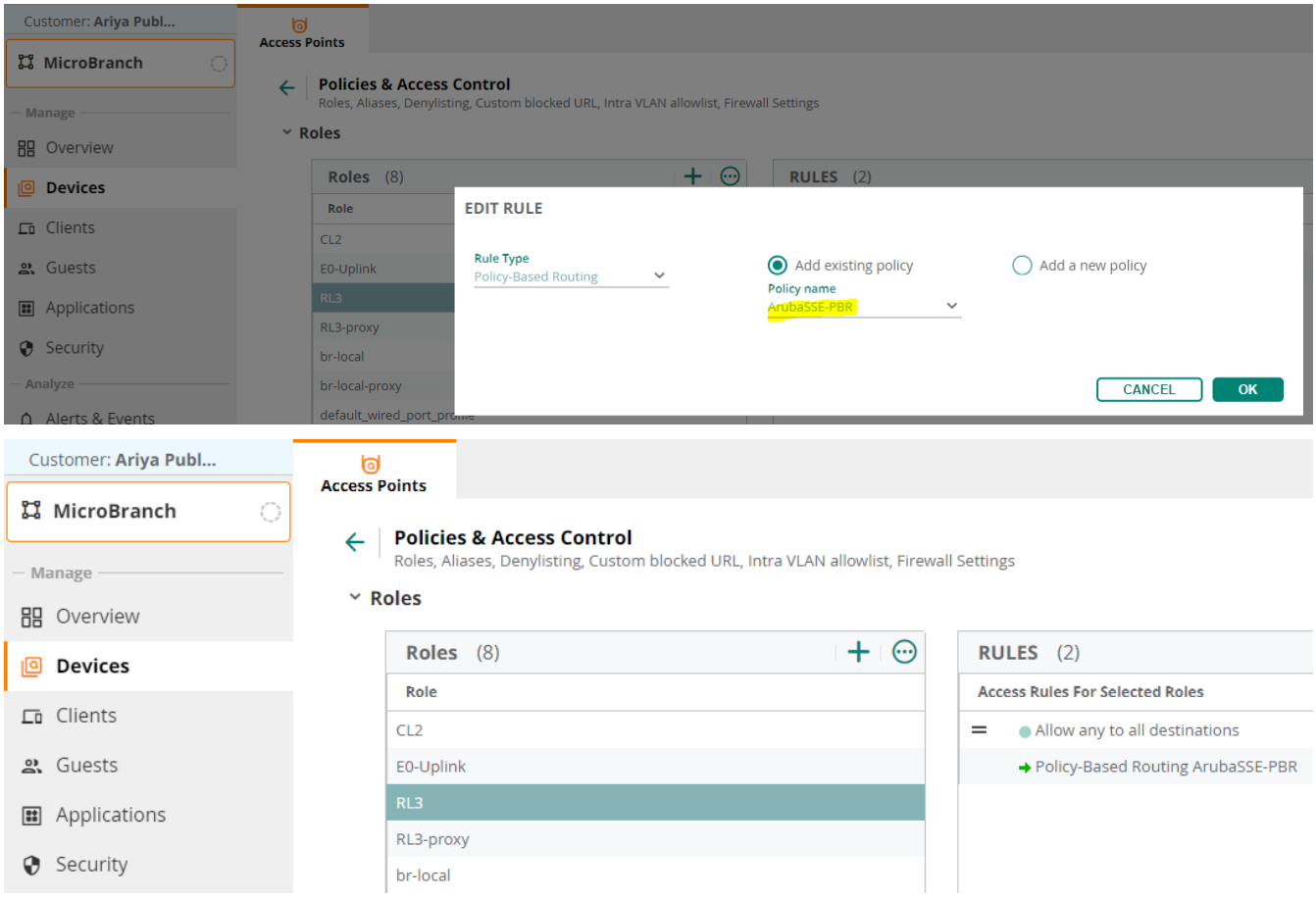




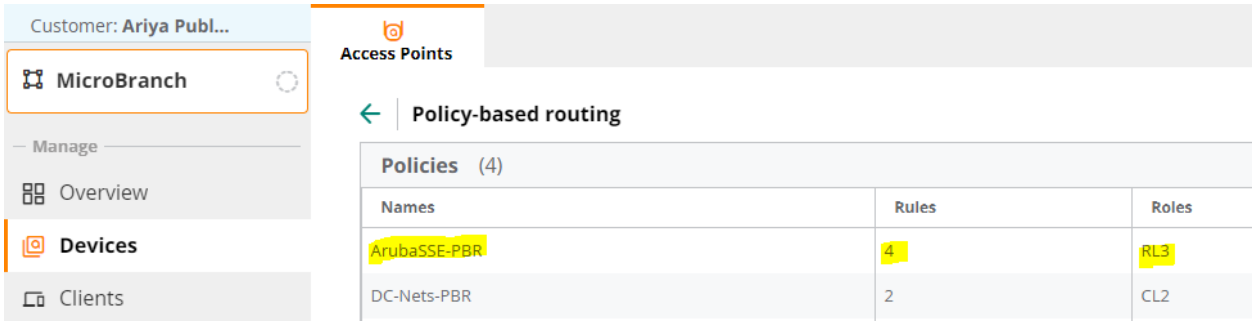
Here is my basic redirection policy for web traffic.



4. Associate the PBR policy with a user role, by going to "Policies & Access Control". Here we are associating it to RL3 user role.

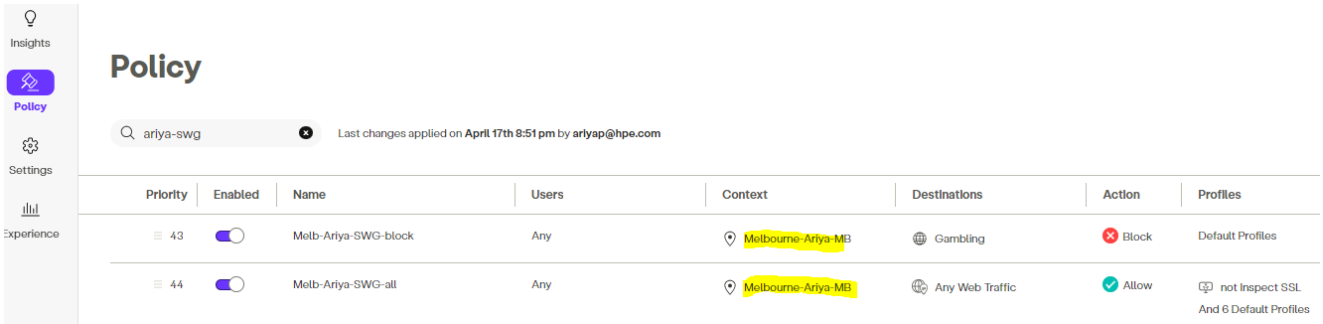


And here is the final PBR.



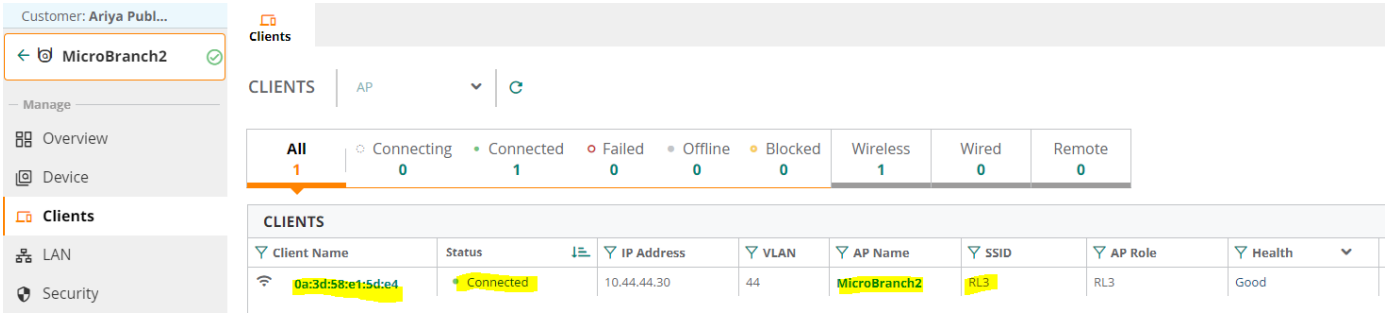
2.7 Aruba SSE configuration

We need to add some policies for the incoming redirected HTTP/HTTPS traffic from the MB. As shown below we have these simple rules for blocking gambling sites and the other allowing the rest of the traffic coming specifically from my microbranch APs.



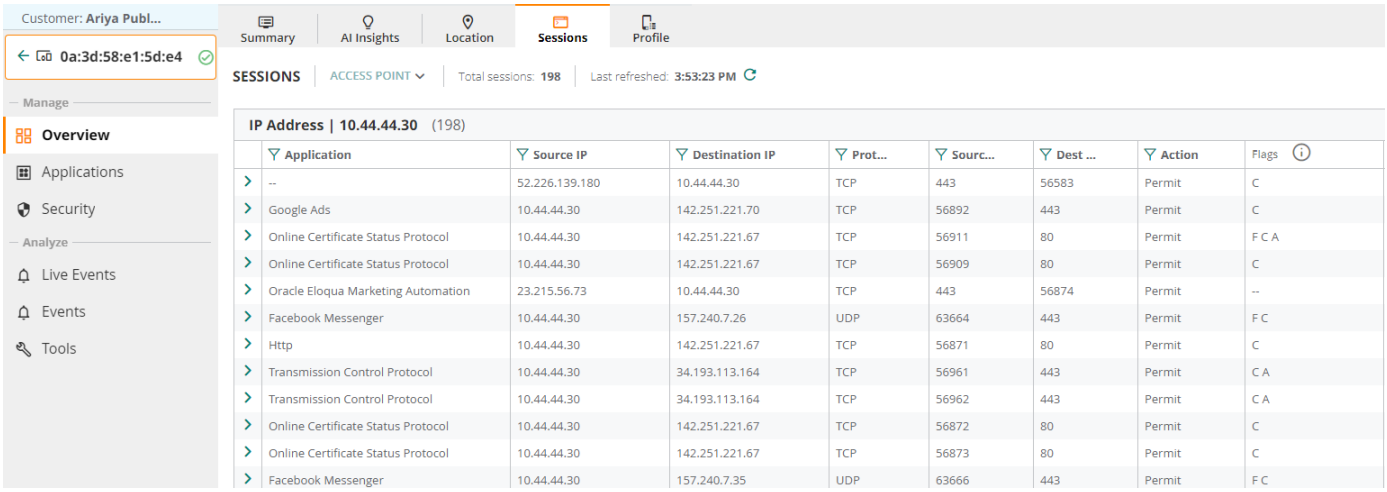
2.8 User Testing

We are ready to test now by connecting the user to RL3 SSID.

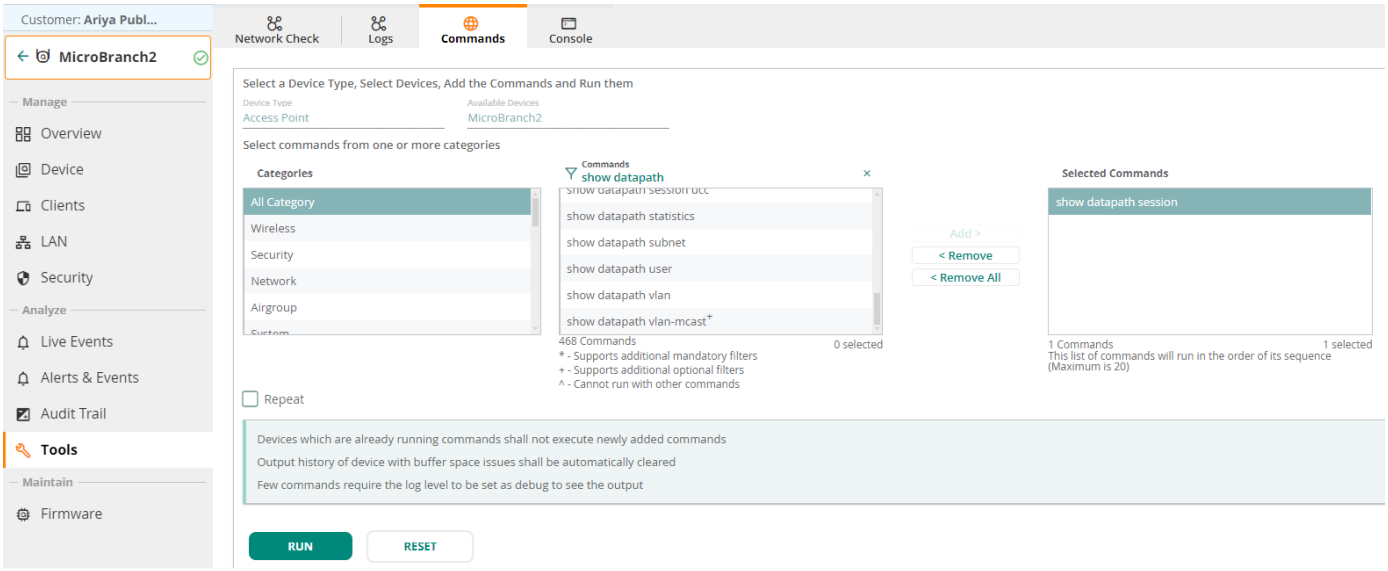


Now that the user is connected we'll open the browser and generate traffic by going to various sites and also gambling related sites to check our compliance rule tat should block gambling sites.

Here is the client session table from Aruba Central.



You can use the Tools->Command to run "show datapath session" command on the microbranch AP.



DEVICE OUTPUT

DEVICE

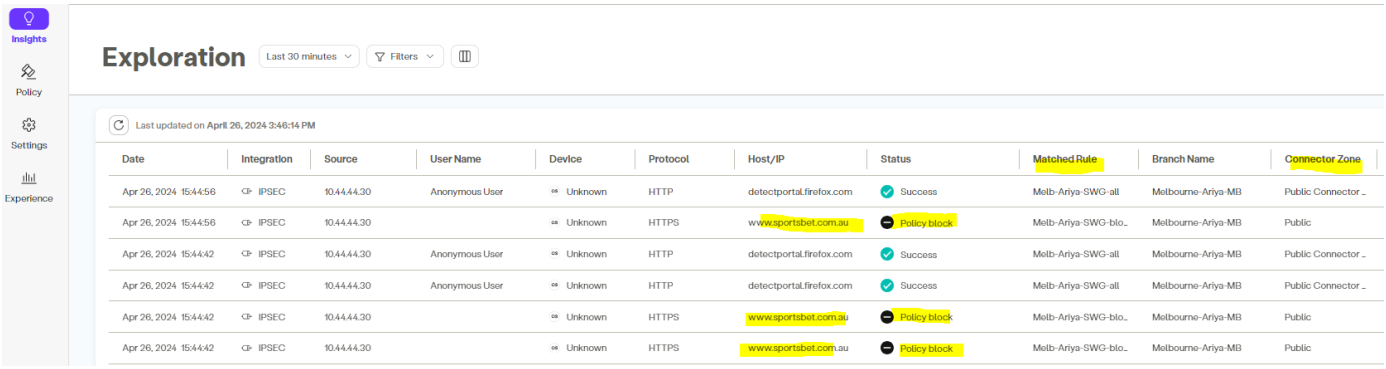
MicroBranch2

Output for the device: MicroBranch2

```
Flow Offload Denylist Flags: O - Openflow, E - Default, U - User os unknown, T - Tunnel
                             R - L3 route

Source IP         Destination IP  Prot SPort Dport Cntr Prio ToS Age Destination TAge Packets Bytes Flags  Offload flags
----------------- --------------- ---- ----- ----- ---- ---- --- --- ----------- ---- ------- ----- ------ -------------
34.111.194.12     10.44.44.30     6    443   57088 0    0    0   0   dev13       206  5       ef    Ci
192.168.2.1       192.168.2.49    1    3983  0     0    0    16  0   dev13       10   1       5c    FA
10.44.44.30       142.251.221.78  6    57077 443   0    0    0   0   dev13       1cc  4       c7    i
192.168.2.1       192.168.2.49    1    3982  0     0    0    16  0   dev13       11   1       5c    FA
192.168.2.1       192.168.2.49    1    3981  0     0    0    16  0   dev13       12   1       5c    FA
192.168.2.1       192.168.2.49    1    3985  0     0    0    16  0   dev13       e    1       5c    FA
192.168.2.1       192.168.2.49    1    3984  0     0    0    16  0   dev13       f    1       5c    FA
10.44.44.30       52.143.87.66    6    57126 443   0    0    0   0   pbr-nhl 1   219  d       709   Ci
139.180.160.82    192.168.2.49    17   123   123   0    0    0   0   local       10   1       4c    F
10.44.44.30       142.251.221.67  6    57076 80    0    0    0   0   pbr-nhl 1   1c9  5       cd    CAd
10.44.44.30       142.251.221.67  6    57081 80    0    0    0   0   pbr-nhl 1   1c8  5       cd    CAd
52.226.139.180    10.44.44.30     6    443   56473 0    0    0   0   dev13       11   1       28    CA
10.44.44.30       34.111.208.231  6    57116 443   0    0    0   0   pbr-nhl 1   257  11      b4d   Ci
172.217.167.115   10.44.44.30     6    443   57092 0    0    0   0   dev13       1f5  5       ef    Ci
172.217.167.115   10.44.44.30     6    443   57091 0    0    0   0   dev13       202  5       ef    Ci
10.44.44.30       44.207.238.46   6    57123 443   0    0    0   0   pbr-nhl 1   22f  e       cf5   Ci
10.44.44.30       52.84.150.36    6    57128 443   0    0    0   0   pbr-nhl 1   210  d       6f9   Ci
192.28.144.124    10.44.44.30     6    443   57104 0    0    0   0   pbr-nhl 1   225  6       138   i
10.44.44.30       35.174.210.7    6    57124 443   0    0    0   0   pbr-nhl 1   21e  d       c57   Ci
```

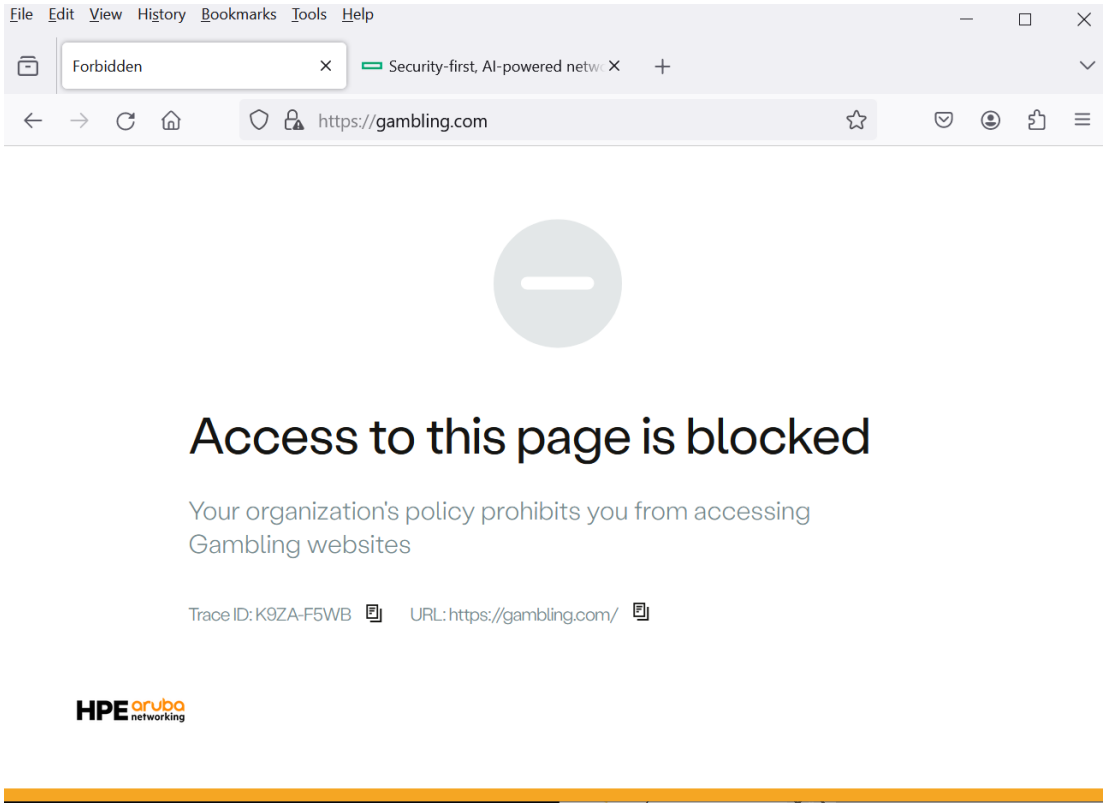
Look for the destination column and you should see the next hop list that we configured.

Now let's check the Aruba SSE dashboard.



As shown above we can see some permitted and blocked traffic. Looking closely you'll see the block sites are displayed and all are gambling related as per our configured policy.

This is what the user will see when trying to browse to gambling sites.

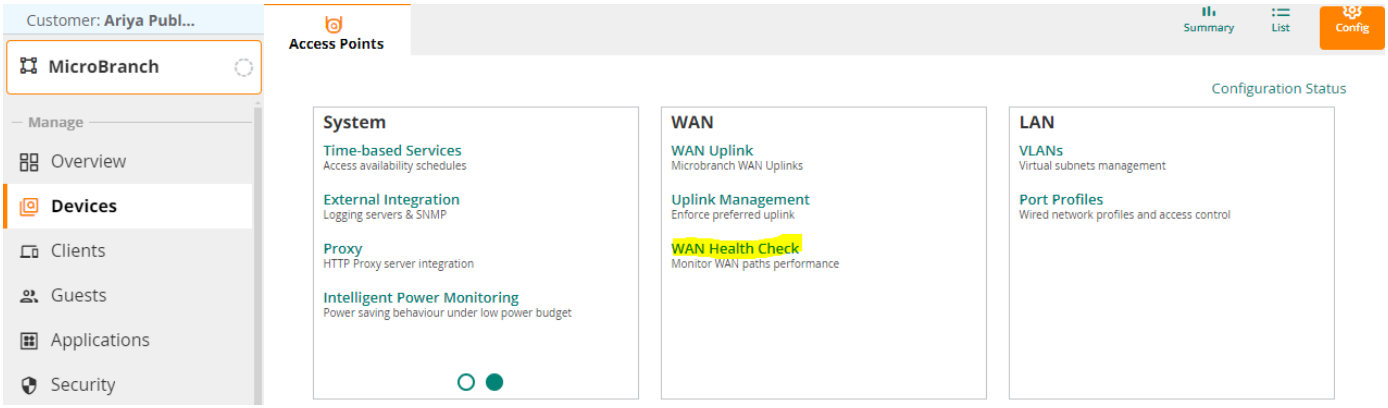


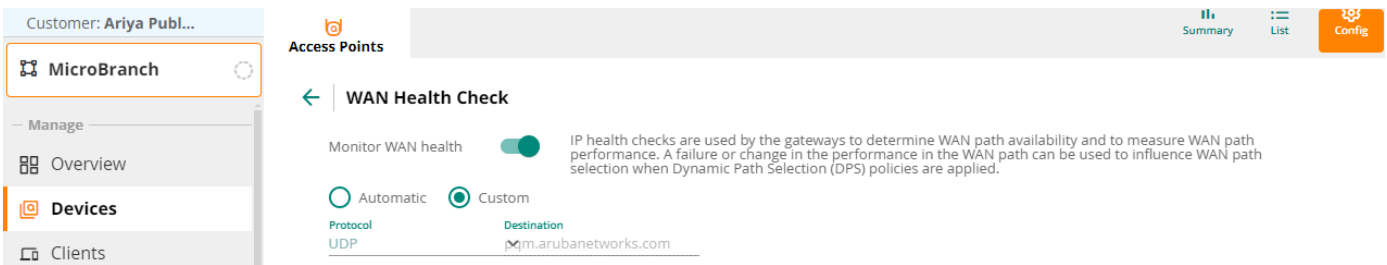
Nowe we can also check the user session table on the MB to see if the next hop list are being used. Note that the client IP address is 10.44.44.30

```
MicroBranch2# sh datapath session
Datapath Session Table Entries
------------------------------
Flags: A - Application Firewall Inspect
       C - client, D - deny, E - Media Deep Inspect
       F - fast age, G - media signal, H - high prio
       I - Deep inspect, L - ALG session, M - mirror, N - dest NAT
       O - Session is programmed through SDN/Openflow controller
       P - set prio, R - redirect, S - src NAT,
       T - set ToS, U - Locally destined, V - VOIP
       X - Http/https redirect for dpi denied session
       Y - no syn
       a - rtp analysis, h - Https redirect error page
       i - in offload flow, m - media mon
       p - Session is marked as permanent
       s - media signal
       d - DPI cache hit
       f - FIB init pending in session
RAP Flags: 0 - Q0, 1 - Q1, 2 - Q2, r - redirect to conductor
           t - time based, i - in flow, l - local redirect
Flow Offload Denylist Flags: O - Openflow, E - Default, U - User os unknown, T - Tunnel, R - L3 route

Source IP         Destination IP  Prot SPort Dport Cntr Prio ToS Age Destination TAge Packets Bytes Flags  Offload flags
---------------   --------------  ---- ----- ----- ---- ---- --- --- ----------- ---- ------- ----- ------ -------------

MicroBranch2# sh datapath session | incl 10.44.44.30
52.226.139.180    10.44.44.30     6    443   56473 0    0    0   0   dev13       5aa  a       190   Ci
203.134.85.113    10.44.44.30     6    443   57032 0    0    0   0   pbr-nhl 1   58f  20      131d  i
10.44.44.30       44.207.238.46   6    57123 443   0    0    0   0   dev13       55b  e       391   i
10.44.44.30       52.84.150.36    6    57128 443   0    0    0   1   pbr-nhl 1   559  e       2bd   FCi
10.44.44.30       35.174.210.7    6    57124 443   0    0    0   1   pbr-nhl 1   555  15      dbe   Ci
10.44.44.30       34.107.243.93   6    56472 443   0    0    0   0   dev13       511  8       140   i
44.207.238.46     10.44.44.30     6    443   57123 0    0    0   0   dev13       55b  10      454   Ci

MicroBranch2#
```

2.9 Health Check

Microbranch also supports the health check probes to measure WAN availability and latency on uplinks like branch gateways. You can enable it from here.





Once you enable “Monitor WAN Health”, the probes are sent through the underlay and based on the probe response the uplinks are marked as available or not. The probes are sent 5x UDP or ICMP every 10 sec. When a probe is lost then the frequency of sending probes increases to every 2 sec. The details are mentioned here.

Here are some useful commands to check the health probes. This command displays built-in profile information and the one we are interested in is health-check probe.

```
MicroBranch2# show hcm probe-profile

Build-in probe profile
----------------------
Name          Probe Mode  Jitter  Frequency(in sec)  Retries  Burst size
----          ----------  ------  -----------------  -------  ----------
default       icmp        No      10                 3        5
health-check  udp         Yes     10                 3        5
data-vpnc     udp         Yes     10                 3        5

MicroBranch2#
```

This is to view the details of the health check probe, copy the Token ID for the uplink health check.

```
MicroBranch2# show hcm probe-node

IP Health-check Entries
-----------------------
Token                                              Token Index  Action ID  Destination    Vlan
Request From     Profile Type  Mode           Method  Status  Probe State       Reason
-----                                              -----------  ---------  -----------
DefaultGW-control-plane-192.168.2.1-4092-2         2            0          192.168.2.1    4092
default-gateway  default       control-plane  icmp    Up      probe in process  hcm probe run
init success

ProbeNodeEvent-control-plane-13.239.61.151-4092-1  1            0          13.239.61.151  4092
Uplink           health-check  control-plane  udp     Up      probe in process  hcm probe run
init success

VPNC-control-plane-192.168.1.57-36860-7            7            0          192.168.1.57   36860
VPNC             data-vpnc     control-plane  udp     Up      probe in process  hcm probe run
init success

MicroBranch2#
```

Now you can use the token ID in this command to see the details of the probe result like jitter, etc.

```
MicroBranch2# show hcm probe-node token ProbeNodeEvent-control-plane-13.239.61.151-4092-1

Token: ProbeNodeEvent-control-plane-13.239.61.151-4092-1, Token index: 1
Running test case id 0, running process pid 9424
Request Node ID: 0
        Request Node Name: ProbeNodeEvent-control-plane-13.239.61.151-4092-1_0
        Destination: 13.239.61.151
        Vlan: 4092
        Action: probe start
        Report Type: 1
        Probe Mode: control-plane
        Probe Method: udp
        Probe From: Uplink
        Probe Count(0 means always): 0
        Probe Profile: health-check
                How Often Start One Test: 10
                The Number Of Packets For Each Test: 5
                The Times AP doesn't get response Will change the status to Down: 3
        Probe Result:
                State: Up
                Latency(ms): 21.760
                Jitter(ms): 0.472098
                MOS: 4.4
                Packet Loss: 0%
        Probe Running State: probe in process
        Probe Running Note: hcm probe run init success

MicroBranch2#
```