SD-EWAN Scenarios

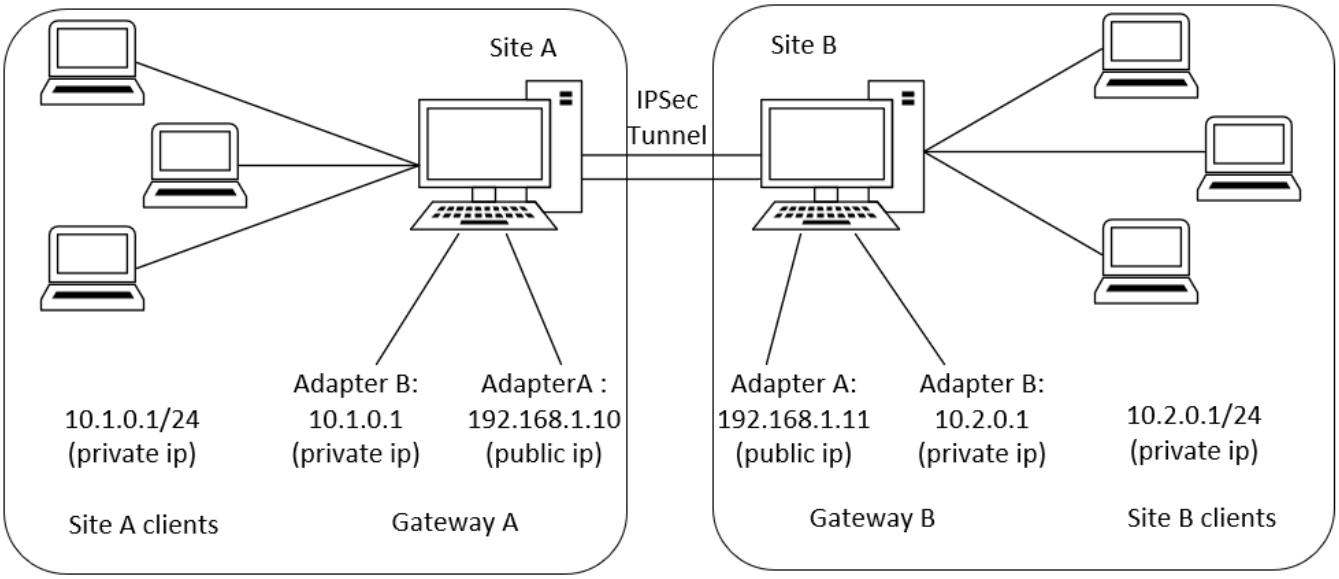
This page intend to list bunch of scenarios for our SD-EWAN case, including the decomposed scenarios and the overall integrated scenario.

- Decomposed scenarios
  - Decomposed Scenario A: Site-to-Site tunnel with static public IP address
    - CRs for the scenario
      - Rest calls
  - Decomposed Scenario B: Host-to-Site tunnel when the initiator requests an overlay IP
    - CRs for the scenario
    - Rest calls
  - Decomposed Scenario C: Host to host tunnel
    - CRs for the scenario
    - Rest calls
- Targeted Scenarios
  - Scenario A: Edge to traffic hub tunnel where inter micro-service communication across edges that attached to same traffic hub.
  - Scenario B: Edge to Edge tunnels when micro-service communication happens across edges without involving hubs
  - Scenario C: Hub to hub tunnel when inter micro-service communication across edges that attached to different traffic hubs
- Overall scenarios

Decomposed scenarios

Decomposed Scenario A: Site-to-Site tunnel with static public IP address

In this scenario, both sites have static public IP address and setup a tunnel between sites. After the tunnel is established, the clients within the site should be able to ping the clients on the other side through the tunnel. The tunnel is authenticated through pre-shared key.



Scenario Description:

Tunnel between site A and site B
Suppose there are two sites A and B. A comes with the subnet 10.1.0.1/24, B comes with the subnet 10.2.0.1/24
Gateway for A is 192.168.1.10
Gateway for B is 192.168.1.11
A and B would like to establish a tunnel
10.1.0.1/24 == 10.2.0.1/24

CRs for the scenario

Proposal CR

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecProposal
metadata:
  name: test_proposal_1
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  encryption_algorithm: aes128
  hash_algorithm: sha256
  dh_group: modp3072
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Sample CR for gateway A

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: siteA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 192.168.1.11
  authentication_method: psk
  pre_shared_key: test123
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_subnet: 10.1.0.1/24
      remote_subnet: 10.2.0.1/24
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Sample CR for gateway B

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: siteB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 192.168.1.10
  authentication_method: psk
  pre_shared_key: test123
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_B
      type: tunnel
      mode: start
      local_subnet: 10.2.0.1/24
      remote_subnet: 10.1.0.1/24
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Rest calls

Sites settings

GET /cgi-bin/luci/sdewan/ipsec/v1/sites

```
{
   "sites": [
    {
          "name": “siteA",
          "remote":"192.168.1.11",
          "crypto_proposal": "test_proposal_1",
          "pre_shared_key": "test123",
          "authentication_method": "psk",
          "connections": [
           { "name": "connection_A",
             "type": "tunnel",
             "mode": "start",
             "local_subnet": "10.1.0.1/24",
             "remote_subnet": "10.2.0.1/24",
             "crypto_proposal": "test_proposal_1"
           }
         ]
      },
     {   "name": "siteB",
          "gateway":"192.168.1.10",
          "crypto_proposal": "test_proposal_1",
          "pre_shared_key": "test123",
          "authentication_method": "psk",
          "remote_identifier": "@moon.strongswan.org",
          "local_identifier": "@sun.strongswan.org",
          "connections": [
           { "name": "connection_B",
             "type": "tunnel",
             "mode": "start",
             "local_subnet": "10.2.0.1/24",
             "remote_subnet": "10.1.0.1/24",
             "crypto_proposal": "test_proposal_1"
           }
      ]
     }
  ]
}
```

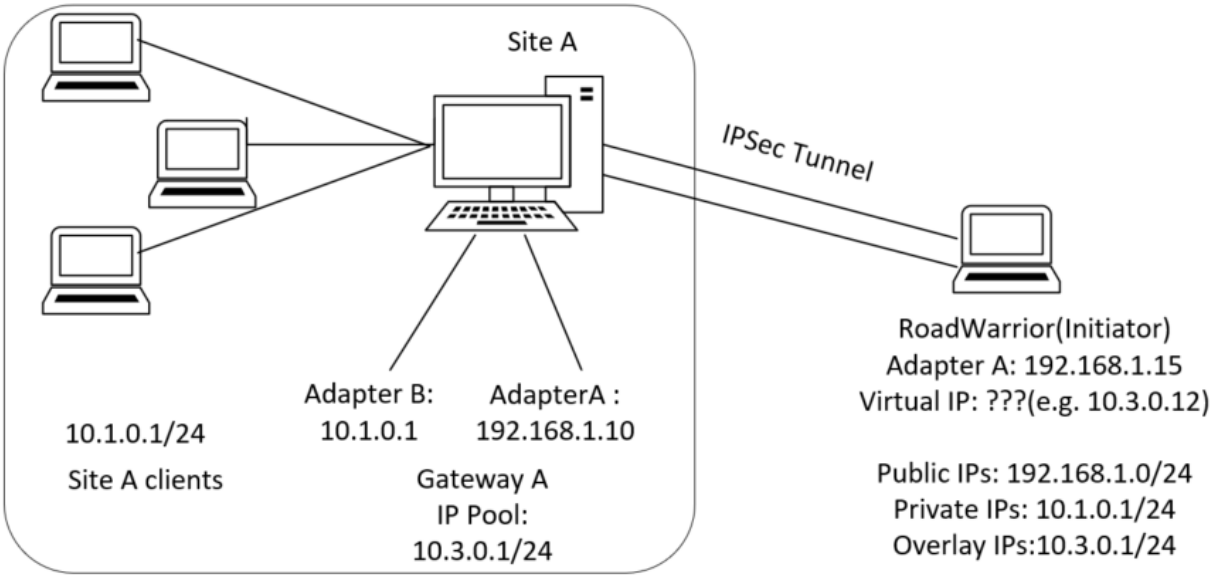

Proposal settings

GET /cgi-bin/luci/sdewan/ipsec/v1/proposals

```
{ "proposals": [
        {
          "name": "proposal1",
          "crypto_algorithm": "aes128",
          "hash_algorithm": "sha256",
          "dh_group": "modp3072"
        }
    ]
}
```

Decomposed Scenario B: Host-to-Site tunnel when the initiator requests an overlay IP

In this scenario, the initiator sends out a request to the responder(either a site gateway/remote host) which has a static public ip address(or dynamic pubic IP with static domain name) in order to setup a tunnel between. However, this time, the roadwarrior is also going to ask for a virtual IP that assigned by the responder. After the tunnel is established, the roadwarrior should be able to get an overlay IP and ping the clients on the other side through the tunnel. The tunnel is authenticated through pre-shared key.



Scenario Description:

Tunnel between site A and host B(Responder and Initiator)
Suppose there is one site A and one host B. A comes with the subnet 10.1.0.1/24.
Gateway for A is 192.168.1.10 which is a public ip address
Host B has no public address and want to request one from the peer(suppose the vip assigned is 10.3.0.12)
A and B would like to establish a tunnel
10.1.0.1/24 == 10.3.0.12/32

CRs for the scenario

Proposal CR

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecProposal
metadata:
  name: test_proposal_1
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  encryption_algorithm: aes128
  hash_algorithm: sha256
  dh_group: modp3072
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Sample CR for gateway A

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: siteA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: %any
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_subnet: 10.1.0.1/24
      remote_sourceip: 10.3.0.1/24
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Sample CR for host B

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: hostB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 192.168.1.10
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_sourceip: %config
      remote_subnet: 0.0.0.0/0
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Rest calls

Sites settings

GET /cgi-bin/luci/sdewan/ipsec/v1/sites

```
{
  "sites": [
    {   "name": "siteA",
        "remote":"%any",
        "crypto_proposal": "test_proposal_1",
        "pre_shared_key": "test123",
        "authentication_method": "psk",
        "local_identifier": "@moon.strongswan.org",
        "remote_identifier": "@sun.strongswan.org",
        "connections": [
          { "name": "connA",
            "type": "tunnel",
            "mode": "start",
            "local_subnet": "10.1.0.1/24",
            "remote_sourceip": "10.3.0.1/24",
            "crypto_proposal": "test_proposal_1"
          }
        ]
    },
    {   "name": "hostB",
        "remote":"192.168.1.10",
        "crypto_proposal": "test_proposal_1",
        "pre_shared_key": "test123",
        "authentication_method": "psk",
        "remote_identifier": "@moon.strongswan.org",
        "local_identifier": "@sun.strongswan.org",
        "connections": [
          { "name": "connA",
            "type": "tunnel",
            "mode": "start",
            "local_sourceip": "%config",
            "remote_subnet": "10.1.0.1/24",
            "crypto_proposal": "test_proposal_1"
          }
    ]
  }
  ]
}
```

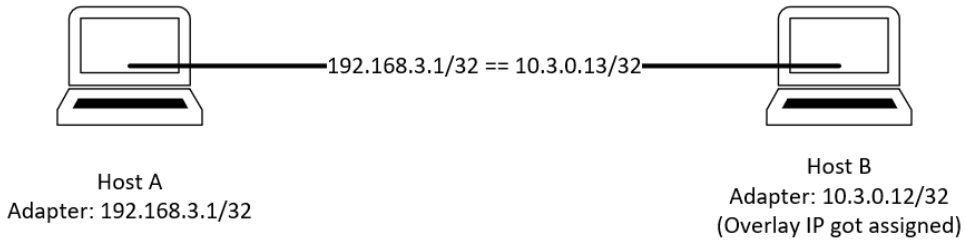

Proposal settings

GET /cgi-bin/luci/sdewan/ipsec/v1/proposals

```
{ "proposals": [
    {
        "crypto_algorithm": "aes128",
        "hash_algorithm": "sha256",
        "dh_group": "modp3072"
    }
  ]
}
```

Decomposed Scenario C: Host to host tunnel

Setup a tunnel between the host who got assigned the virtual IP and another host with PIP.



Scenario Description:

Tunnel between host A and host B
Suppose there are two hosts A and B.
A has a public ip which is 192.168.3.1
B is a host which already get a vip 10.3.0.12
A and B would like to establish a tunnel
192.168.3.1/32 == 10.3.0.12/32

CRs for the scenario

Proposal CR

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecProposal
metadata:
  name: test_proposal_1
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  encryption_algorithm: aes128
  hash_algorithm: sha256
  dh_group: modp3072
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Sample CR for host A

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: hostA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: %any
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      remote_sourceip: 10.3.0.12
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Sample CR for host B

```
apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: hostB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 192.168.3.1
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_sourceip: 10.3.0.13
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

Rest calls

Sites settings

GET /cgi-bin/luci/sdewan/ipsec/v1/sites

```
{
   "sites": [
     {   "name": "hostA",
         "remote":"%any",
         "crypto_proposal": "test_proposal_1",
         "pre_shared_key": "test123",
         "authentication_method": "psk",
         "local_identifier": "@moon.strongswan.org",
         "remote_identifier": "@sun.strongswan.org",
         "connections": [
          { "name": "connA",
            "type": "tunnel",
            "mode": "start",
            "remote_sourceip": "10.3.0.12",
            "crypto_proposal": "test_proposal_1"
          }
         ]
      },
     {   "name": "hostB",
         "remote":"192.168.3.1",
         "crypto_proposal": "test_proposal_1",
         "pre_shared_key": "test123",
         "authentication_method": "psk",
         "remote_identifier": "@moon.strongswan.org",
         "local_identifier": "@sun.strongswan.org",
         "connections": [
          { "name": "connA",
            "type": "tunnel",
            "mode": "start",
            "local_sourceip": "10.3.0.12",
            "crypto_proposal": "test_proposal_1"
          }
      ]
     }
  ]
}
```

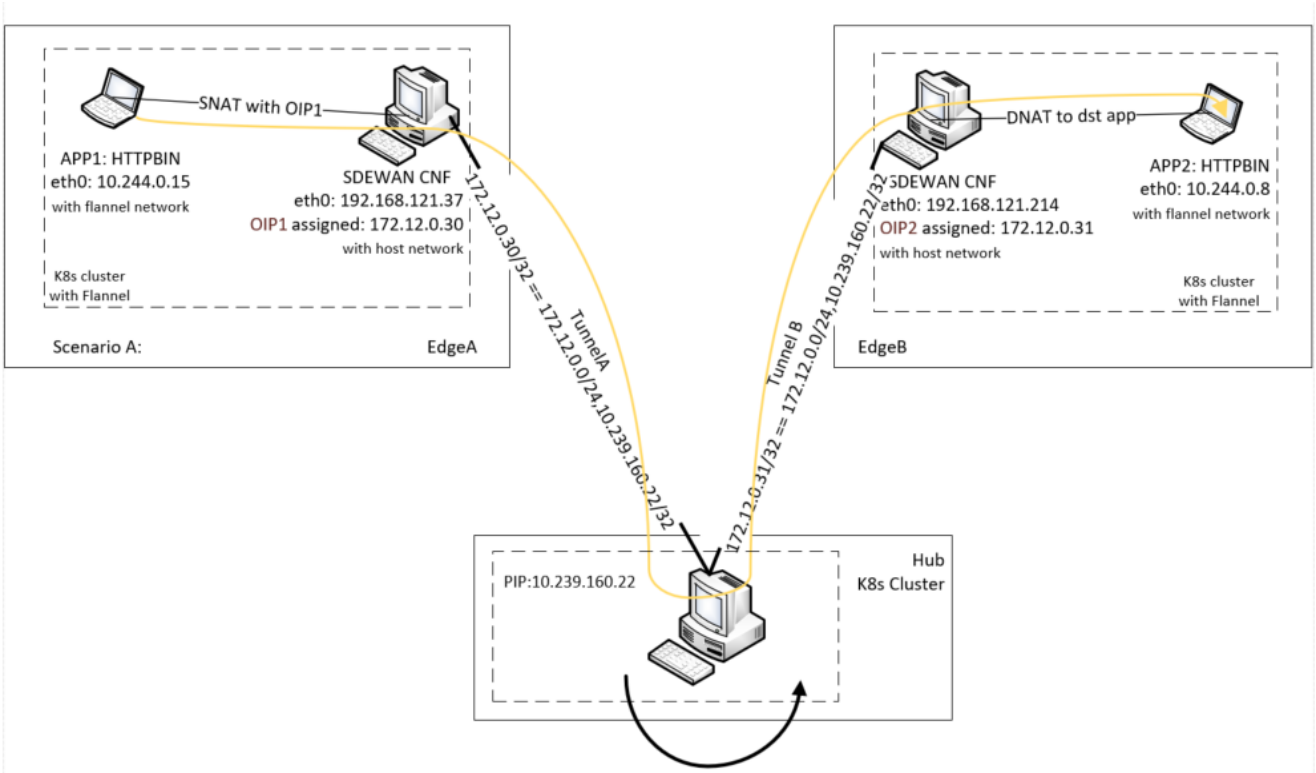

Proposal settings

GET /cgi-bin/luci/sdewan/ipsec/v1/proposals

```
{ "proposals": [
    {
        "crypto_algorithm": "aes128",
        "hash_algorithm": "sha256",
        "dh_group": "modp3072"
    }
  ]
}
```

Targeted Scenarios

Scenario A: Edge to traffic hub tunnel where inter micro-service communication across edges that attached to same traffic hub.



Sample CR for edgeA

```
CR for sdewan cnf on edgeA:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: edgeA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.160.22
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_sourceip: %config
      remote_subnet: 0.0.0.0/0,
10.239.160.22/32
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for edgeA

```
Strongswan configs for sdewan cnf on edgeA:

conn siteA-connA
  left=%any
  right=10.239.160.22
  leftsourceip=%config
  rightsubnet=0.0.0.0/0,10.239.160.22/32
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  esp=aes192-sha1-modp3072
  ike=aes192-sha1-modp3072
  type=tunnel
```

Sample CR for edgeB

```
CR on sdewan cnf on edgeB:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: edgeB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.160.22
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_sourceip: %config
      remote_subnet: 0.0.0.0/0,
10.239.160.22/32
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for edgeB

```
Strongswan configs for sdewan cnf on edgeB:

conn edgeB-connA
  left=%any
  right=10.239.160.22
  leftsourceip=%config
  rightsubnet=0.0.0.0/0,10.239.160.22/32
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  esp=aes192-sha1-modp3072
  ike=aes192-sha1-modp3072
  type=tunnel
```

Sample CR for Hub

```
CR on sdewan on hub:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: Hub
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: %any
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_subnet: 172.12.0.1/24,
10.239.160.22/32
      remote_sourceip: 172.12.0.1/24
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

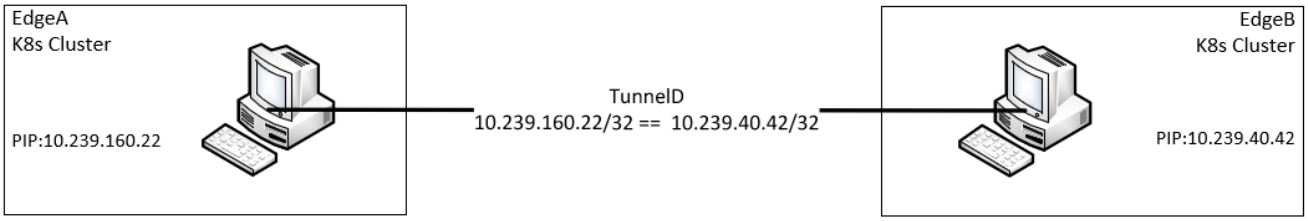
The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for hub

```
Strongswan configs for sdewan cnf on hub:

conn tunnel
  left=10.239.160.22
  leftsubnet=172.12.0.1/24,10.239.160.22/32
  rightsourceip=172.12.0.30-172.12.0.45
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
```

Scenario B: Edge to Edge tunnels when micro-service communication happens across edges without involving hubs



Sample CR for EdgeA

```
CR on sdewan cnf on edgeA:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: edgeA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.40.42
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for edgeA

```
Strongswan configs for sdewan cnf on edgeA:

conn edgeA-connection_A
  left=%any
  right=10.239.40.42
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
```

Sample CR for EdgeB

```
CR on sdewan cnf on edgeB:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: edgeB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.160.22
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

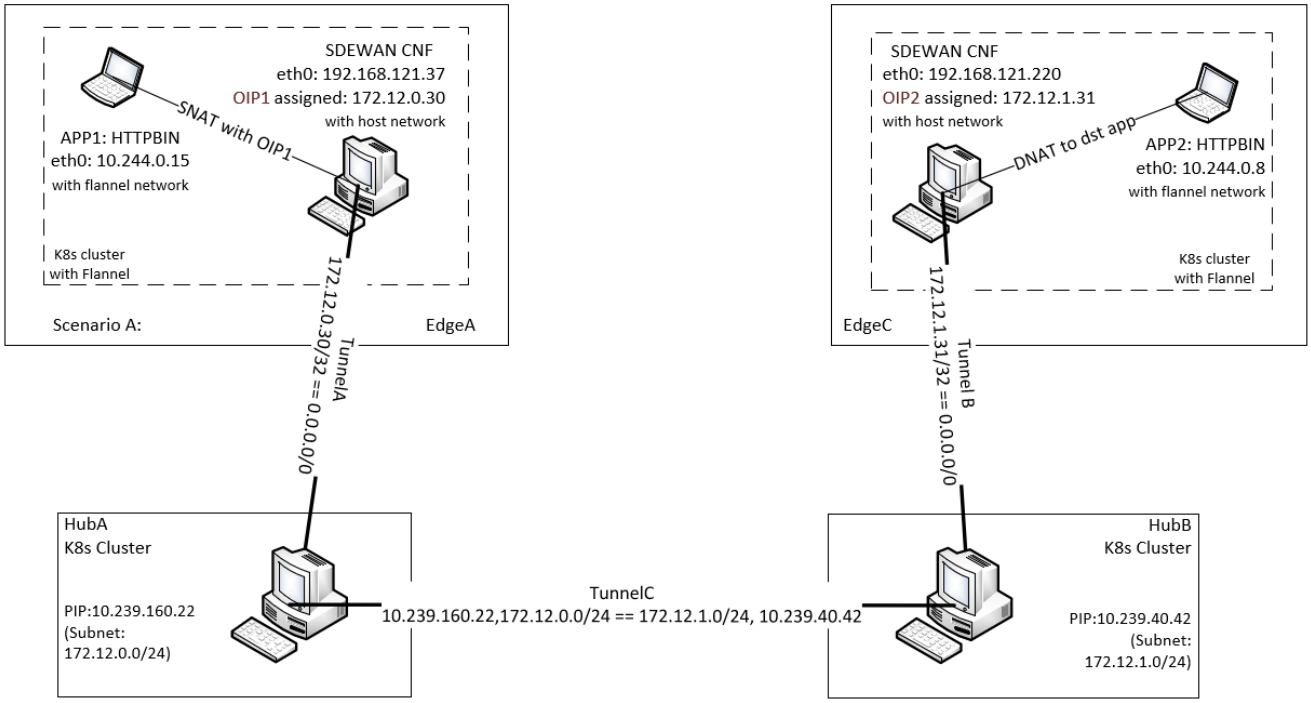
The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for edgeB

```
Strongswan configs for sdewan cnf on edgeB:

conn edgeB-connection_A
  left=%any
  right=10.239.160.22
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
```

Scenario C: Hub to hub tunnel when inter micro-service communication across edges that attached to different traffic hubs



Sample CR for edgeA

```
CR for sdewan cnf on edgeA:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: edgeA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.160.22
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_sourceip: %config
      remote_subnet: 0.0.0.0/0,
10.239.160.22/32
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for edgeA

```
Strongswan configs for sdewan cnf on edgeA:

conn edgeB-connection_A
  left=%any
  right=10.239.160.22
  localsourceip=%config
  rightsubnet=0.0.0.0/0,10.239.160.22/32
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
```

Sample CR for edgeB

```
CR for sdewan cnf on edgeB:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecHost
metadata:
  name: edgeB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.40.42
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_sourceip: %config
      remote_subnet: 0.0.0.0/0,
10.239.40.42/32
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for edgeB

```
Strongswan configs for sdewan cnf on edgeB:

conn edgeB-connection_A
  left=%any
  right=10.239.40.42
  localsourceip=%config
  rightsubnet=0.0.0.0/0,10.239.40.42/32
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
```

Sample CR for HubA

```
CR on sdewan on hubA:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: HubA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: %any
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_subnet: 172.12.0.1/24,
10.239.160.22/32
      remote_sourceip: 172.12.0.30-
172.12.0.45
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True

---

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: HubA
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.40.42
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_B
      type: tunnel
      mode: start
      local_subnet: 172.12.0.1/24,
10.239.160.22/32
      remote_subnet: 172.12.1.1/24,
10.239.40.42/32
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for HubA

```
Strongswan configs for sdewan cnf on hub:

conn HubA-connection_A
  left=%any
  leftsubnet=172.12.0.1/24,10.239.160.22/32
  rightsourceip=172.12.0.30-172.12.0.45
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
conn HubA-connection_B
  left=%any
  leftsubnet=172.12.0.1/24,10.239.160.22/32
  rightsubnet=172.12.1.1/24,10.239.40.42/32
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
```

Sample CR for HubB

```
CR on sdewan on hubB:

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: HubB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: %any
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_A
      type: tunnel
      mode: start
      local_subnet: 172.12.1.1/24,
10.239.40.42/32
      remote_sourceip: 172.12.1.31-
172.12.1.35
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True

---

apiVersion: sdewan.akraino.org/v1alpha1
kind: IpsecSite
metadata:
  name: HubB
  namespace: default
  labels:
    sdewanPurpose: cnf-1
spec:
  remote: 10.239.160.22
  authentication_method: psk
  pre_shared_key: test
  crypto_proposal:
    - test_proposal_1
  connections:
    - connection_name: connection_B
      type: tunnel
      mode: start
      remote_subnet: 172.12.0.1/24,
10.239.160.22/32
      local_subnet: 172.12.1.1/24,
10.239.40.42/32
      crypto_proposal:
        - test_proposal_1
status:
  appliedVersion: "1"
  appliedTime: "2020-04-12T09:28:38Z"
  inSync: True
```

The CRs defined will then be interpreted into some IPSec configuration that could be recognized by Openwrt and then translate to Strongswan configs

Strongswan configs for HubB

```
Strongswan configs for sdewan cnf on hub:

conn HubB-connection_A
  left=%any
  leftsubnet=172.12.1.1/24,10.239.40.42/32
  rightsourceip=172.12.1.31-172.12.1.35
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
conn HubA-connection_B
  left=%any
  leftsubnet=172.12.1.1/24,10.239.40.42/32
  rightsubnet=172.12.0.1/24,10.239.160.42
/32
  leftauth=psk
  rightauth=psk
  auto=start
  keyexchange=ikev2
  ike=aes192-sha1-modp3072
  esp=aes192-sha1-modp3072
  type=tunnel
```

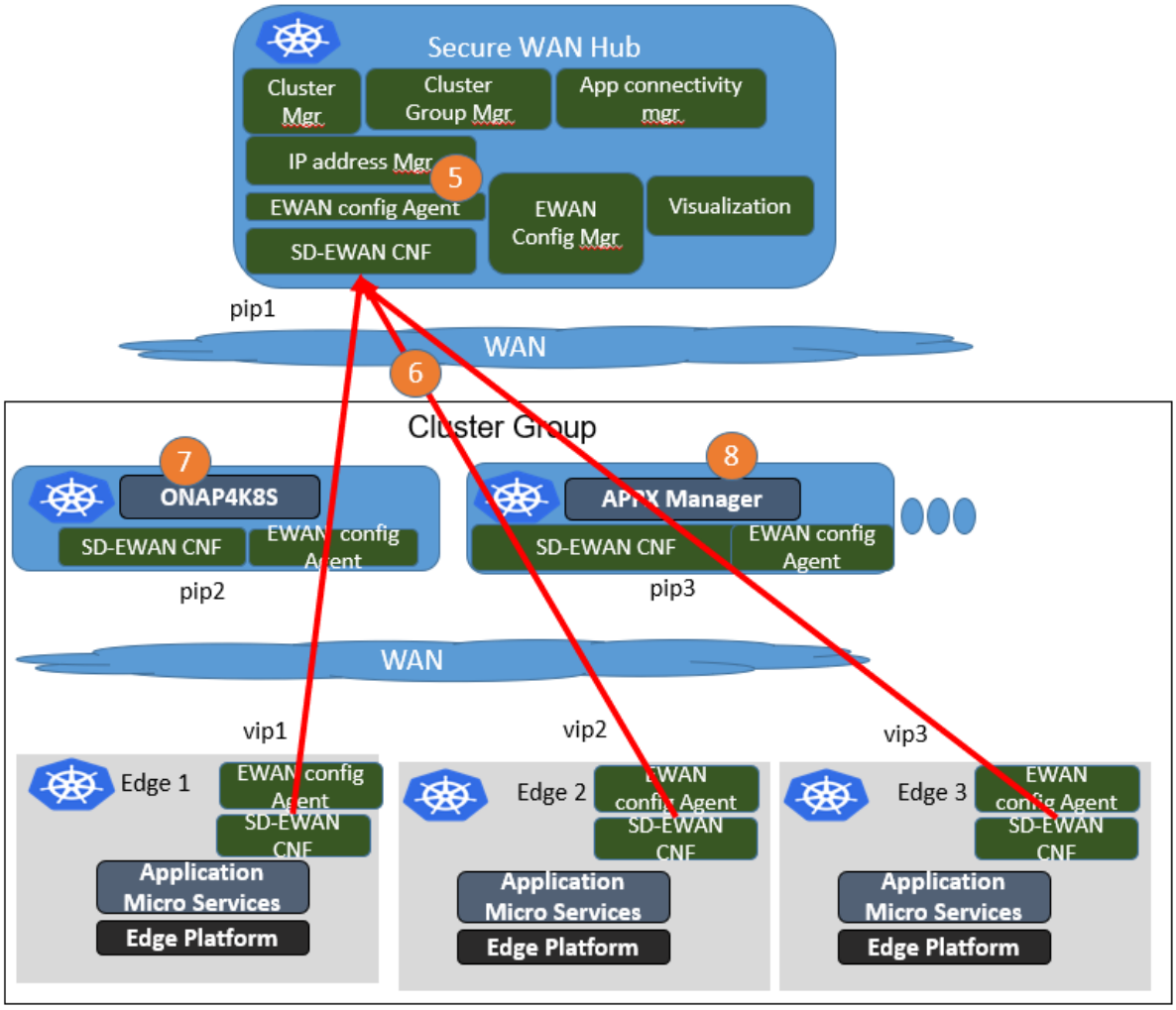
Overall scenarios

Here shows the overall scenario we want to achieve in the ICN SDEWAN case.

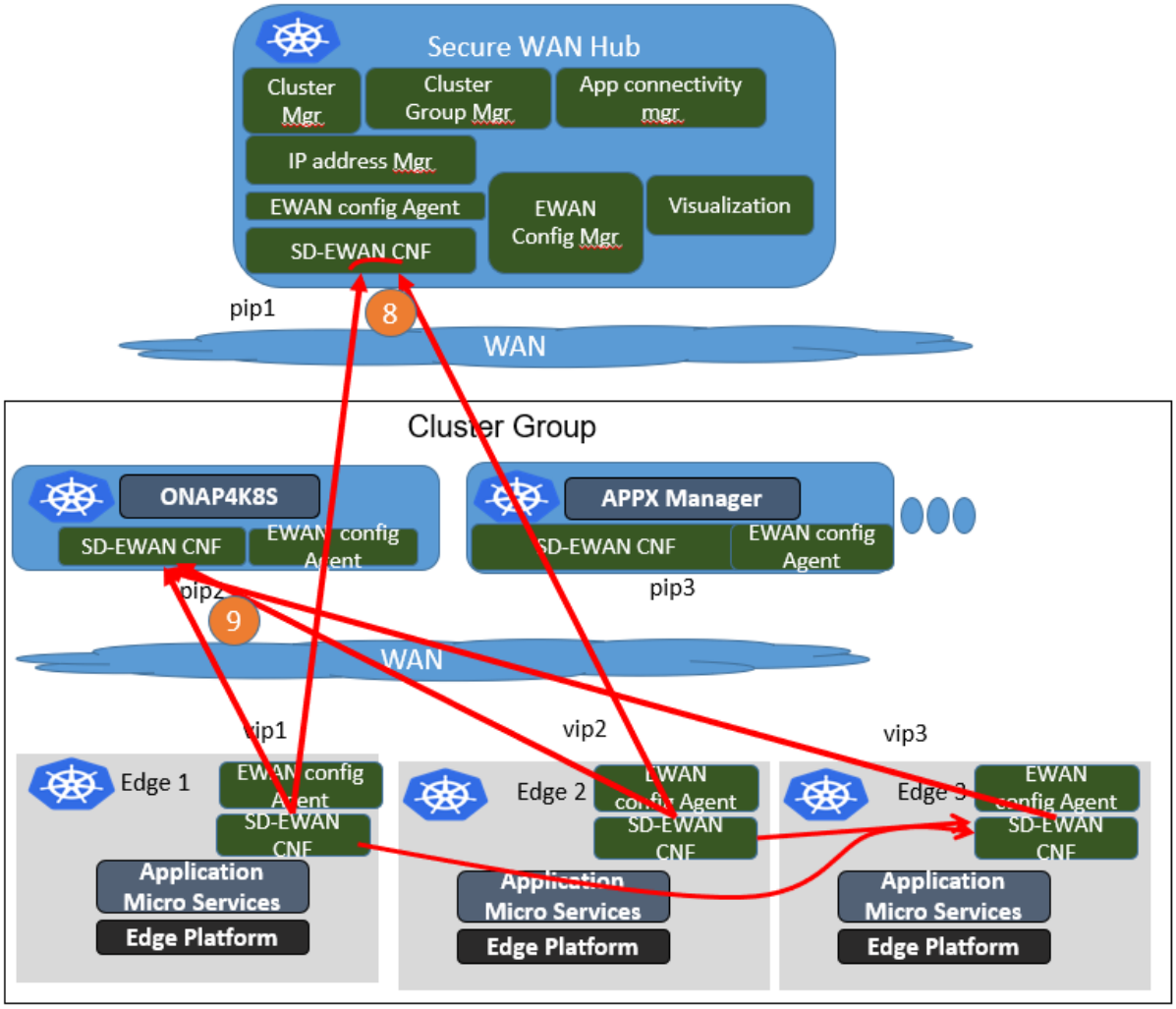
The first step would be the edge initialization. The edges will try to connect to the central Secure WAN hub through the IPsec tunnel. There could be different scenarios containing the decomposed ones listed above:

(a) Initiator to Responder tunnels where there is edge one side with public IP address(or dynamic pubic IP with static domain name). Later, using DNAT to deliver the information to pods inside the cluster.

(b) Initiator to Responder tunnels to get overlay IP address, where the edge initiator don't have public IP address. Later, using DNAT to deliver the information to pods inside the cluster.



Next, the edges would use the virtual IPs/public IPs to setup IPSec tunnels with other clusters. In some cases, they need to go through the SD-EWAN CNF inside the Secure WAN Hub as Spoke and Hub to communicate with each other.


Secure WAN Hub
Cluster Mgr
Cluster Group Mgr
App connectivity mgr
IP address Mgr
EWAN config Agent
SD-EWAN CNF
EWAN Config Mgr
Visualization
pip1
8
WAN
Cluster Group
ONAP4K8S
SD-EWAN CNF
EWAN config Agent
APPX Manager
SD-EWAN CNF
EWAN config Agent
pip3
9
WAN
vip1
vip2
vip3
Edge 1
EWAN config Agent
SD-EWAN CNF
Application Micro Services
Edge Platform
Edge 2
EWAN config Agent
SD-EWAN CNF
Application Micro Services
Edge Platform
Edge 3
EWAN config Agent
SD-EWAN CNF
Application Micro Services
Edge Platform