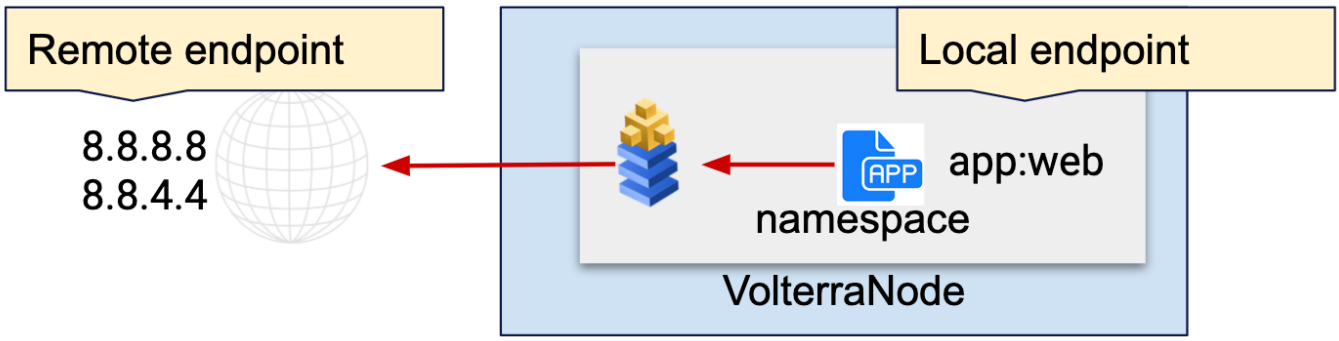
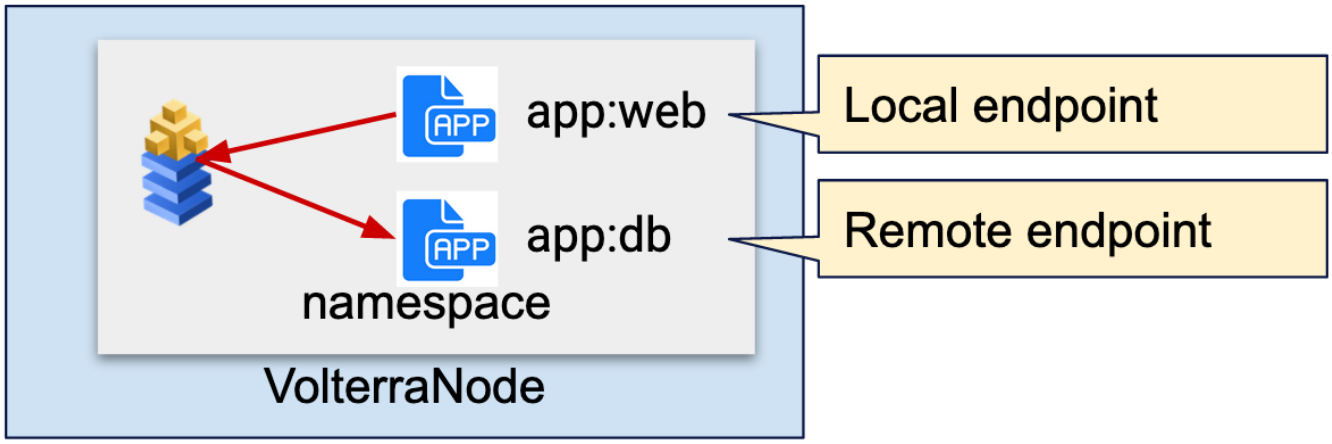
Network policy

Network PolicyはL3-L4のIngress/Egressのセキュリティを提供します。 Remote EndpointからLocal Endpointに入ってくるトラフィックをIngress、Local EndpintからRemote Endpointに出ていくトラフィックをEgressとなります。 例えば以下の場合、Remote Endpointは(8.8.8.8/32, 8.8.4.4/32)となり、Local Endpointは app:webが設定されたPodとなります。

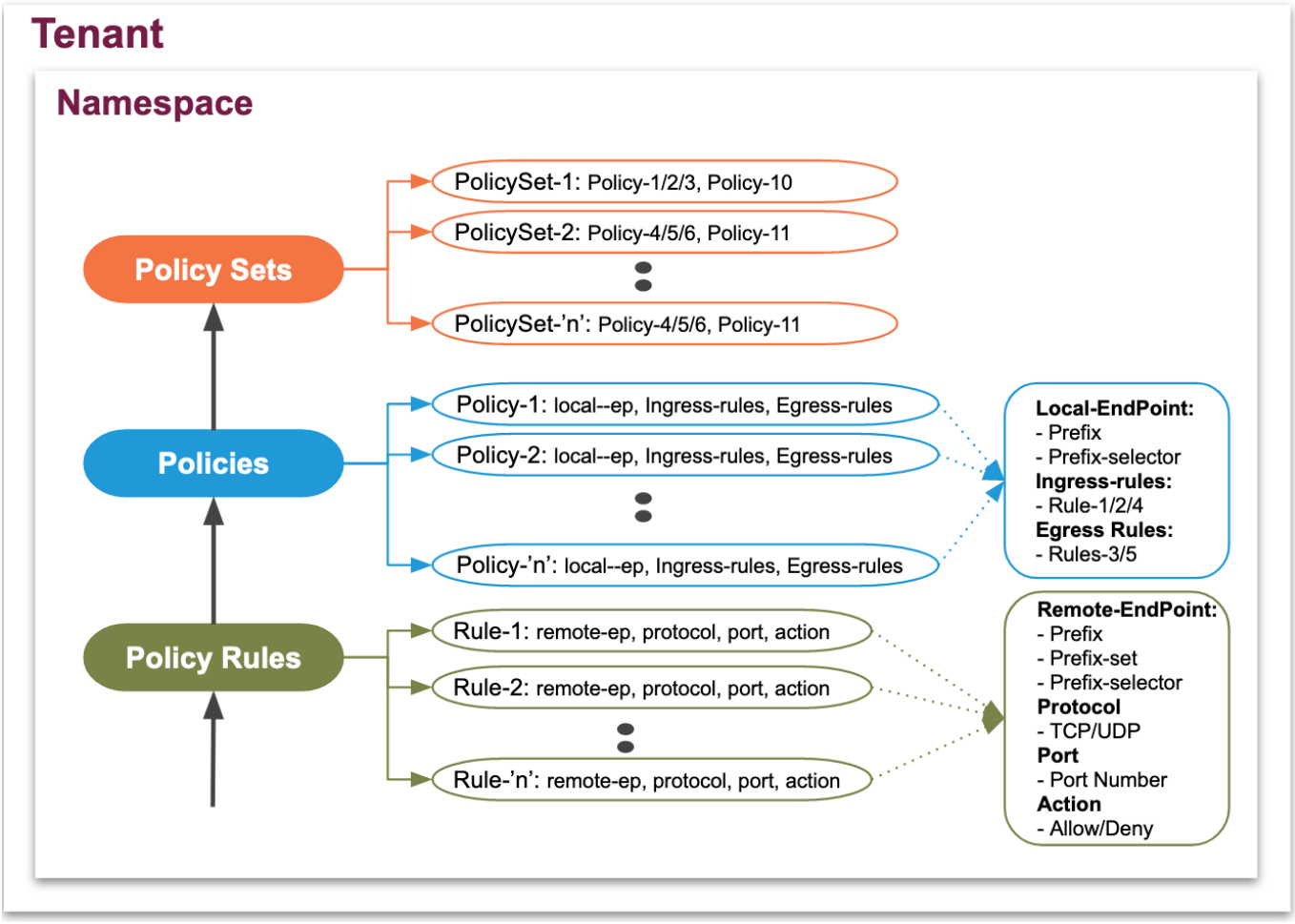


以下の場合、Remote Endpointはapp:dbが設定されたPodとなり、Local Endpointは app:webが設定されたPodとなります。



Network policyの構造

コンフィグNetrowk Policy RuleでRemote endpointの条件を作成し、Network PolicyでLocal Endpointに対してNetwork Policy Ruleを適用します。Network Policy SetでNetwork Policy RuleをNamespaceに対して適用します。



Network Policy

インターネットへの通信制御

namespace:`security`を作成し、vk8sにVirutal siteを設定します。 Name: `pref-tokyo` Site type: `CE` Site Selecter Expression: `pref:tokyo`

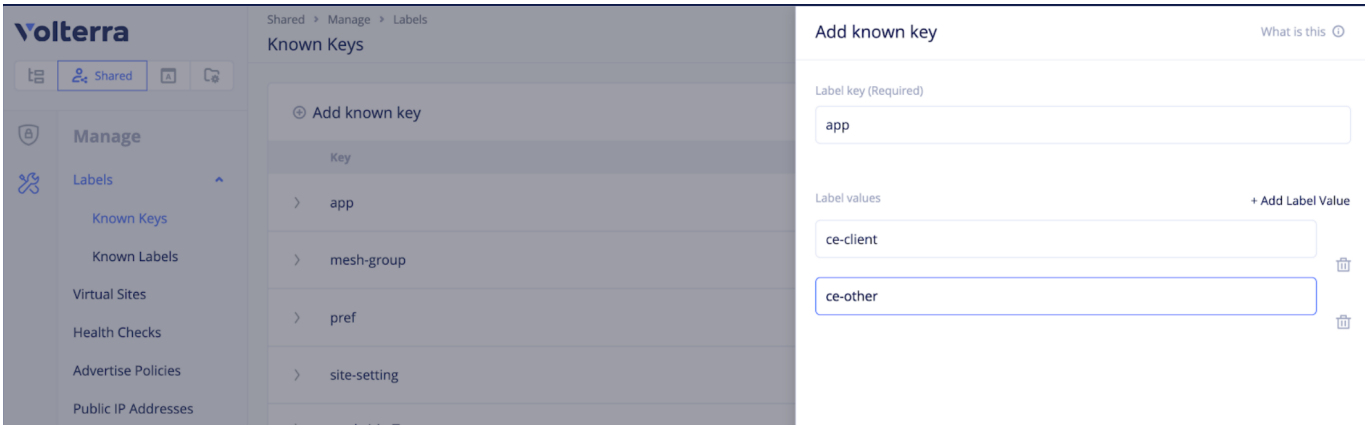
- Freeユーザーの場合は既存のNamespaceを先に削除してから作成してください。

shared namespaceで known keyを作成します。

Label key: `app`

label value:

- `ce-client`
- `ce-other`



Network policyで使用するラベルは、2020/8/24時点でShared namespaceのknown labelsとknown keysに設定されているか、ves.io/app ラベルを使用する必要があります。

ラベルが異なる2つのPod, app:ce-clientとapp:ce-otherを作成します。

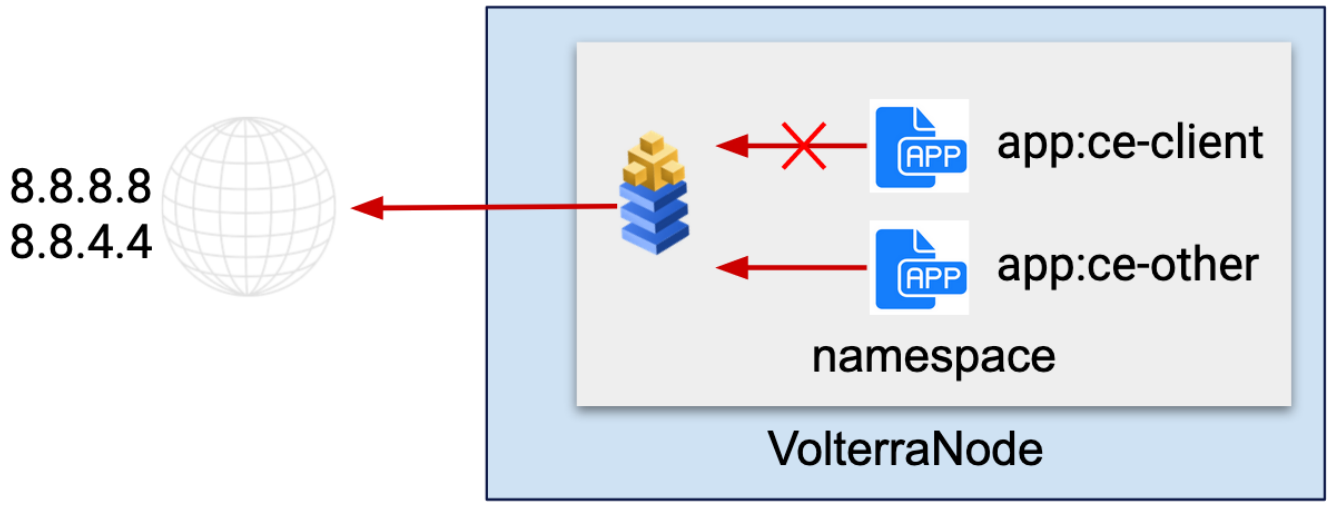
ce-client

```
apiVersion: apps/v1
metadata:
  name: ce-client
  namespace: security
  annotations:
    ves.io/virtual-sites: security/vsite-adc
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ce-client
  template:
    metadata:
      labels:
        app: ce-client
    spec:
      containers:
        - name: ce-client
          image: dnakajima/netutils:1.3
```

```
apiVersion: apps/v1
metadata:
  name: ce-other
  namespace: security
  annotations:
    ves.io/virtual-sites: security/vsite-adc
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ce-other
```

```
    template:
      metadata:
        labels:
          app: ce-other
      spec:
        containers:
          - name: ce-other
            image: dnakajima/netutils:1.3
```

作成したPod, app:ce-clientのにGoogle-DNSへのアクセスを拒否します



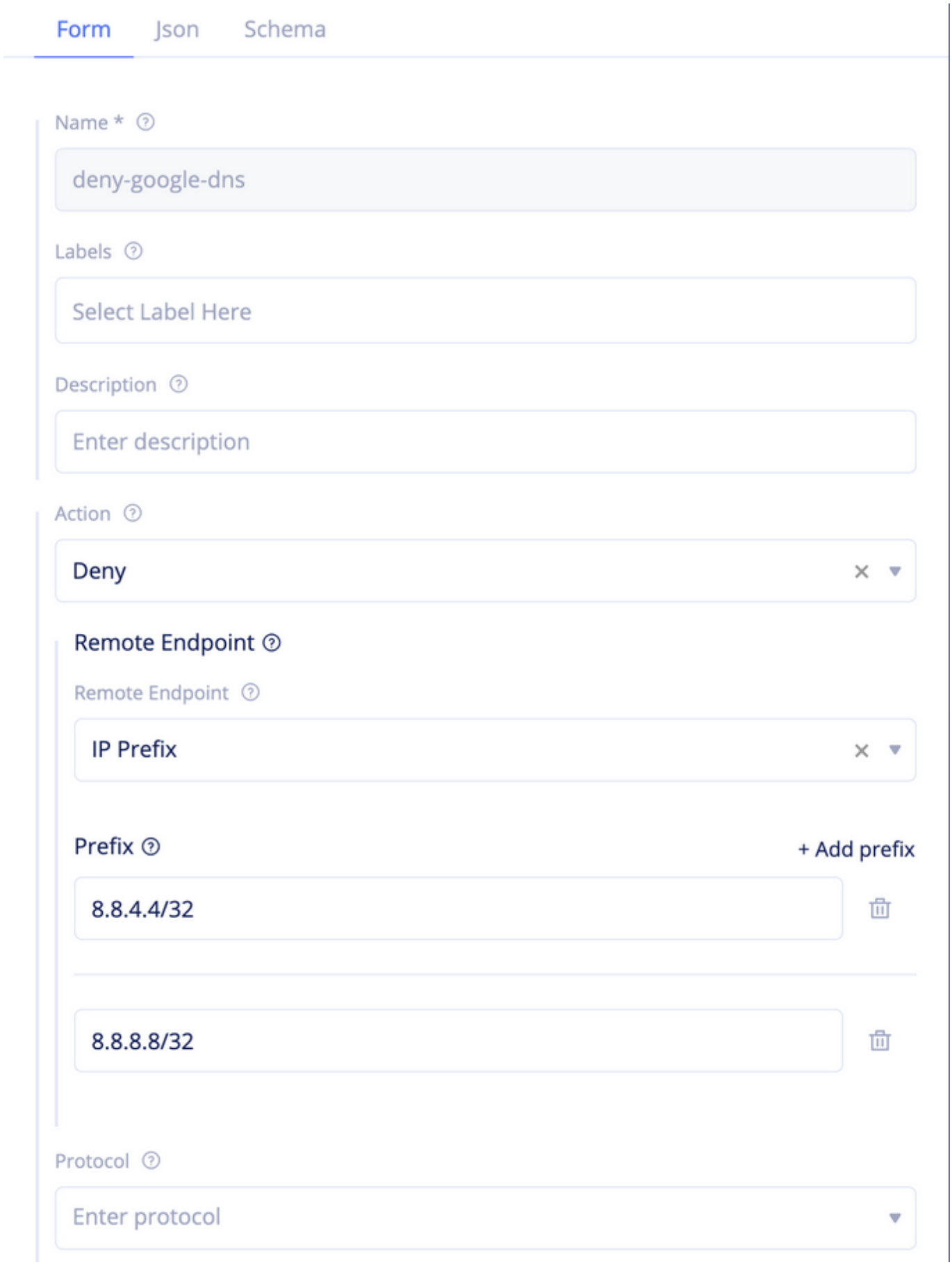
Network Policy Ruleを2つ作成します。

- allow-any Action: Allow

  (** 暗黙のDenyがあるため、設定しないとすべての通信が拒否される)

- deny-google-dns Action: Deny

  Remote Endpoint: IP Prefix: Prefix [8.8.8.8/32, 8.8.4.4/32]



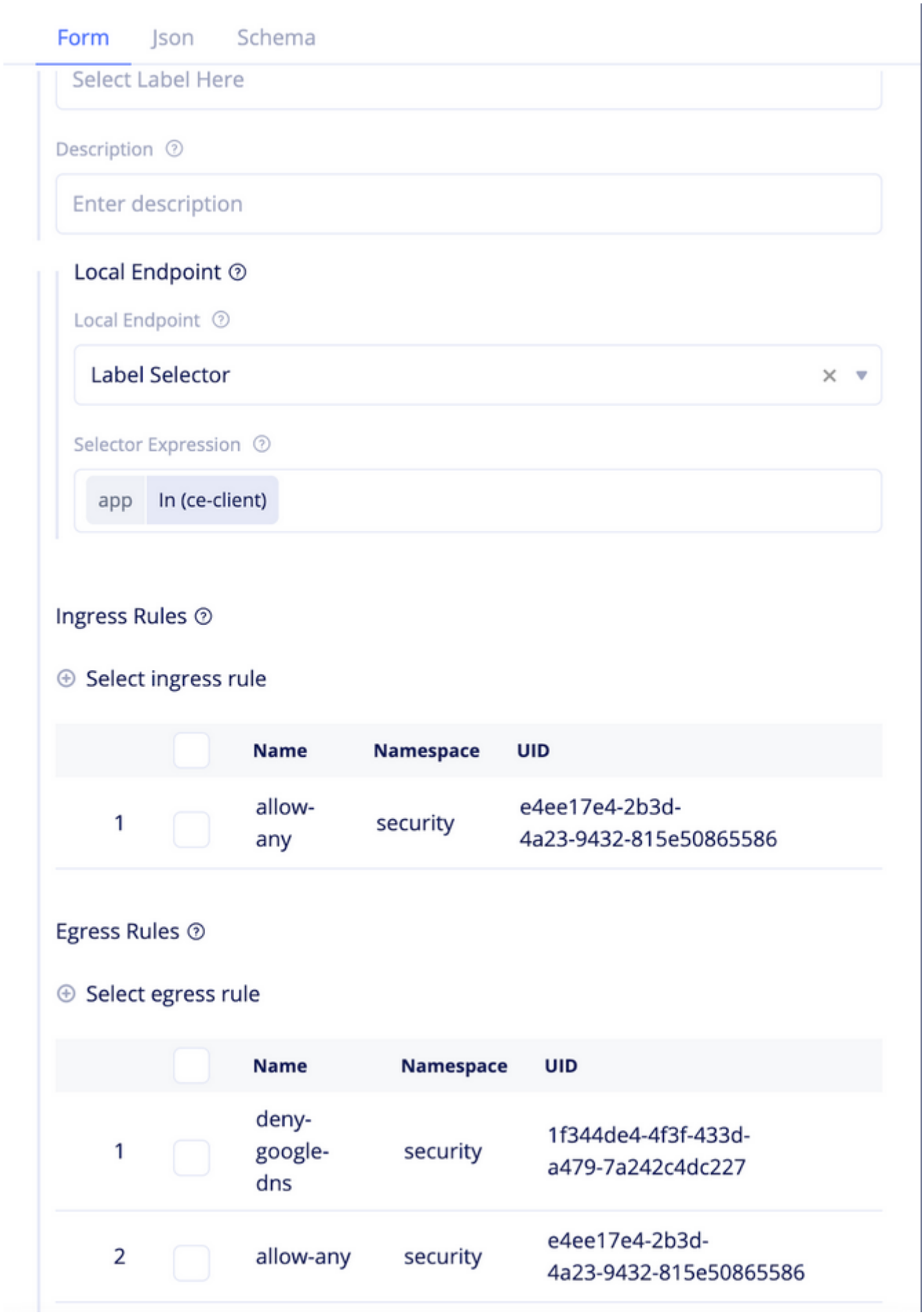
Network Policy を2つ作成します。

- ce-client-po
    - Local Endpoint: Label Selector, Selector

    - Expression: app:in(ce-client)
    - Ingress Rules: 1:allow-any
    - Egress Rules: 1: deny-google-dns, 2: allow-any

- ce-other-po

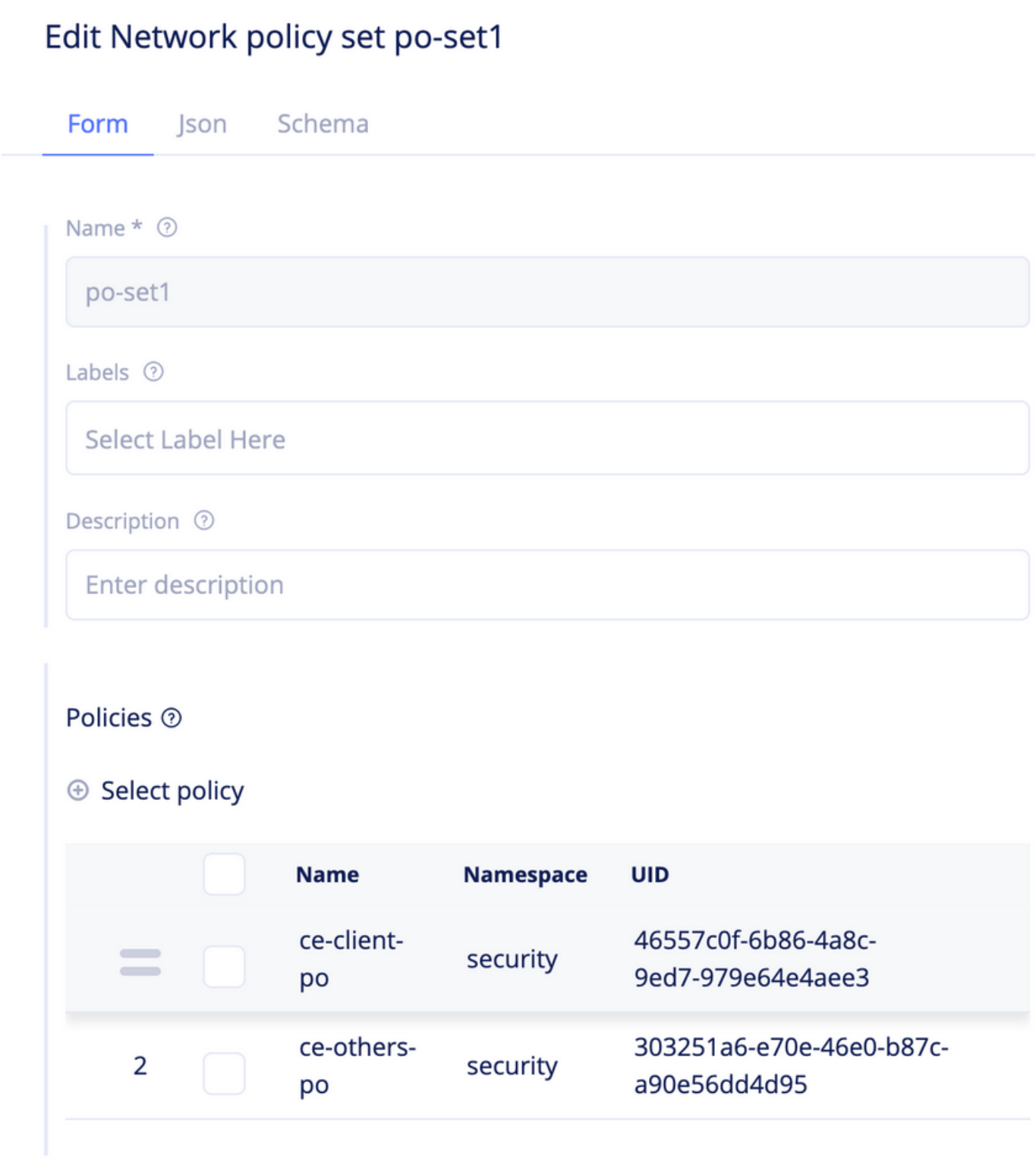
    - Ingress Rules: 1:allow-any
    - Egress Rules: 1:allow-any



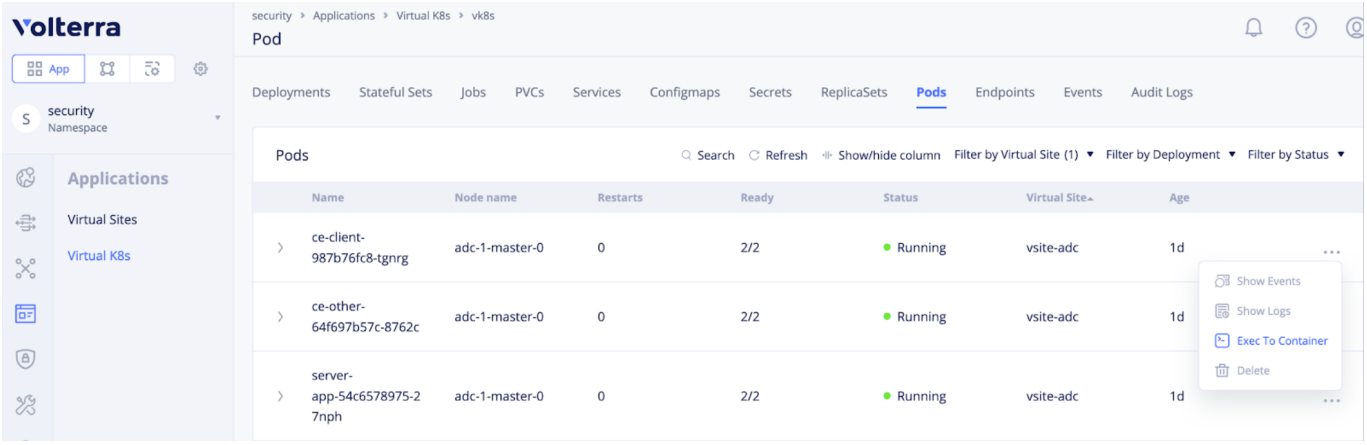
Network Policy Setを作成します。

- po-set1

- Policies: Select policy: [1: ce-client-po, 2: ce-others-po]



フィルターの確認はPodから行えます。Virtual K8sの Pods から対象のPodに Exec to Containerより接続できます。



選択後、Container to exec toから ce-clientやce-otherを選択し、Command to executeにbashを入れるとコンテナにbashで接続できます。

- kubeconfigをダウンロードし、kubectlで接続することも可能です。

ce-clientはgoogle-dnsのポリシーがかかっているため8.8.8.8にはpingできませんが、ce-otherはpingできることが確認できます。

Exec to container on pod ce-client-987b76fc8-tgnrg

```
root@ce-client-987b76fc8-tgnrg:/# ping -c 5 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=2.70 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=2.81 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=2.67 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=55 time=2.99 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=55 time=2.59 ms

--- 1.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 2.593/2.756/2.996/0.143 ms
root@ce-client-987b76fc8-tgnrg:/# ping -c 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4132ms

root@ce-client-987b76fc8-tgnrg:/#
```

Exec to container on pod ce-other-64f697b57c-8762c

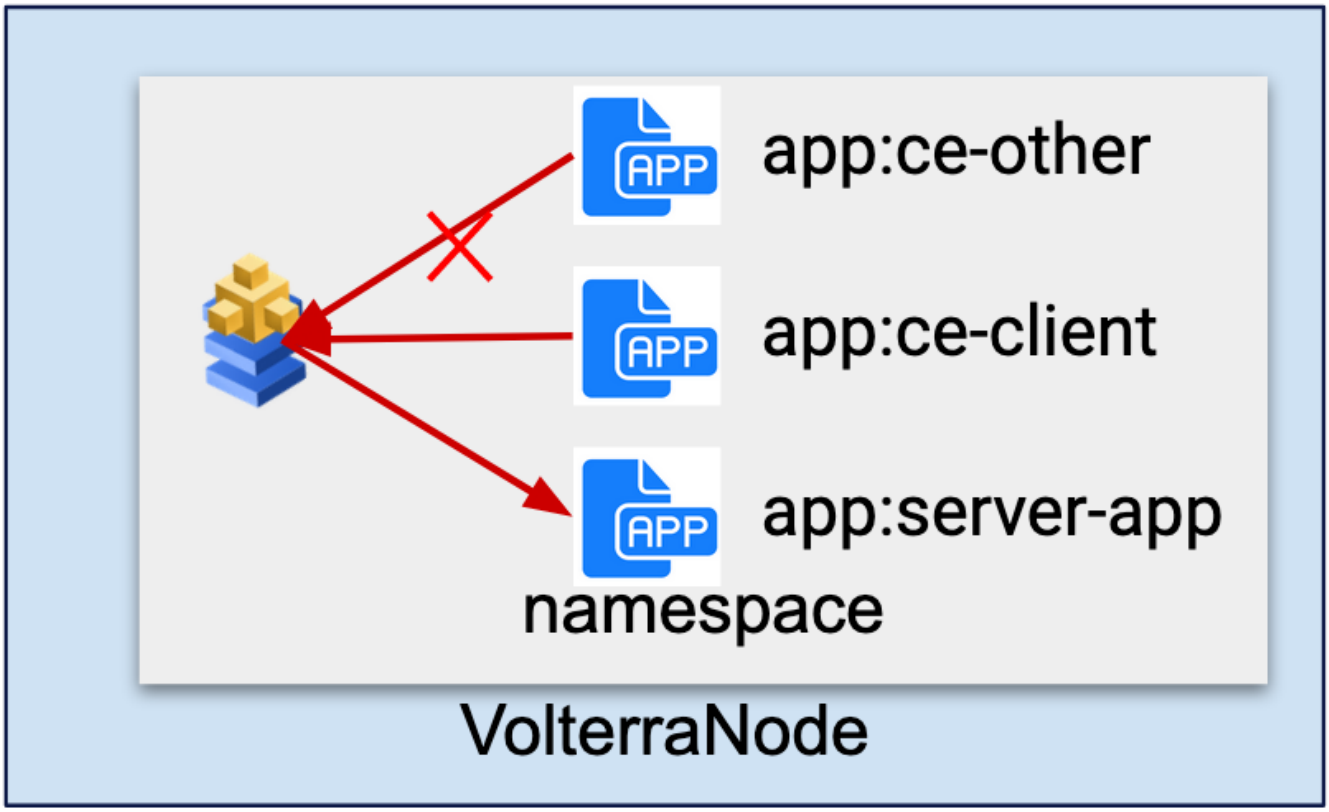
```
root@ce-other-64f697b57c-8762c:/# ping -c 5 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=2.94 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=2.67 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=3.04 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=55 time=2.81 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=55 time=2.66 ms

--- 1.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 2.669/2.829/3.048/0.160 ms
root@ce-other-64f697b57c-8762c:/# ping -c 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=2.61 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=2.52 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=2.49 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=4.54 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=2.48 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 2.484/2.933/4.547/0.810 ms
root@ce-other-64f697b57c-8762c:/#
```

同一Kubernetes Clouster内での通信制御

namespaceは`seurity`とし、virtual-siteは`vsite-adc`を作成します。 ラベルが異なる2つのPod, app:allow-serverとapp:deny-serverを作成します。 app:ce-client からのみapp:server-appへの通信を許可し、 app:ce-otherは拒否します



app:webのPodとServiceを作成します。

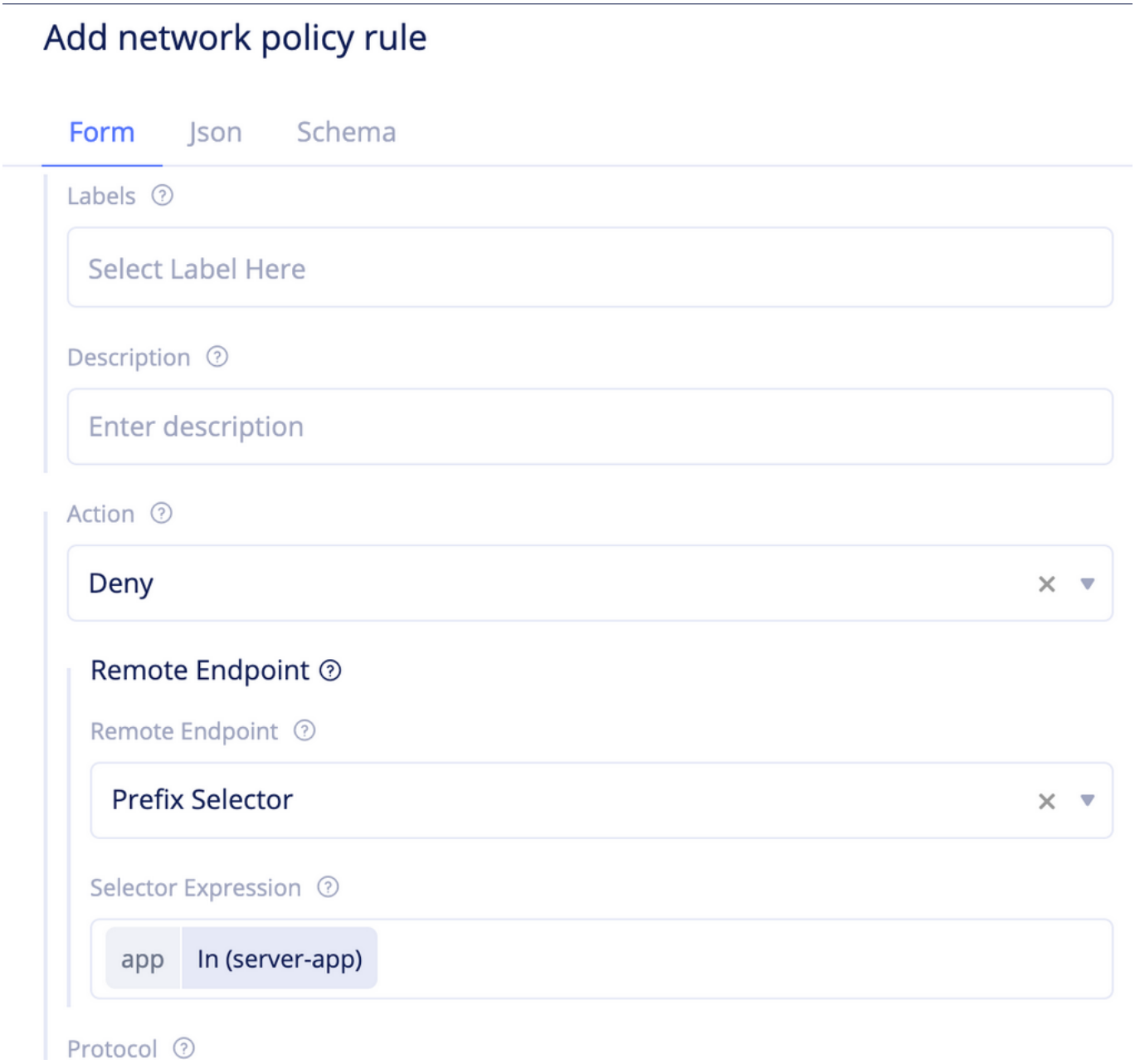
```
apiVersion: apps/v1
metadata:
  name: server-app
  namespace: security
  annotations:
    ves.io/virtual-sites: security/vsite-adc
spec:
  replicas: 1
  selector:
    matchLabels:
      app: server-app
  template:
    metadata:
      labels:
        app: server-app
    spec:
      containers:
        - name: server-app
          image: dnakajima/inbound-app:1.0
          ports:
            - containerPort: 8080
              protocol: TCP
```

```
apiVersion: v1
metadata:
```

```
    name: web
    namespace: security
    labels:
      app: server-app
    annotations:
      ves.io/virtual-sites: security/vsite-adc
  spec:
    ports:
      - protocol: TCP
        port: 80
        targetPort: 8080
    selector:
      app: server-app
    type: ClusterIP
```

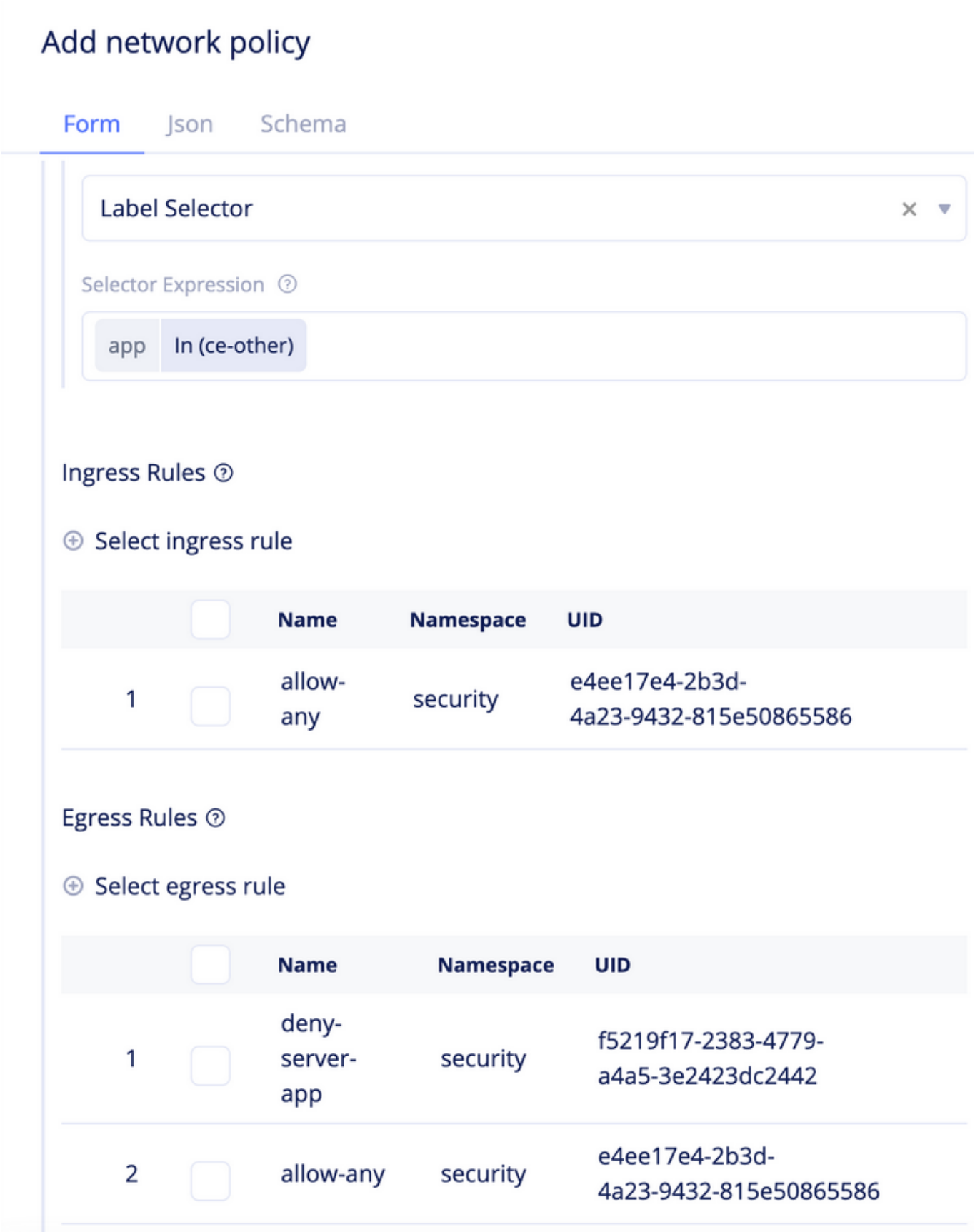
Network Policy Ruleを1つ作成します。

- deny-server-app
    - Action: Deny
    - Remote Endpoint: Prefix Selector
    - Selection Expression: app:in(server-app)



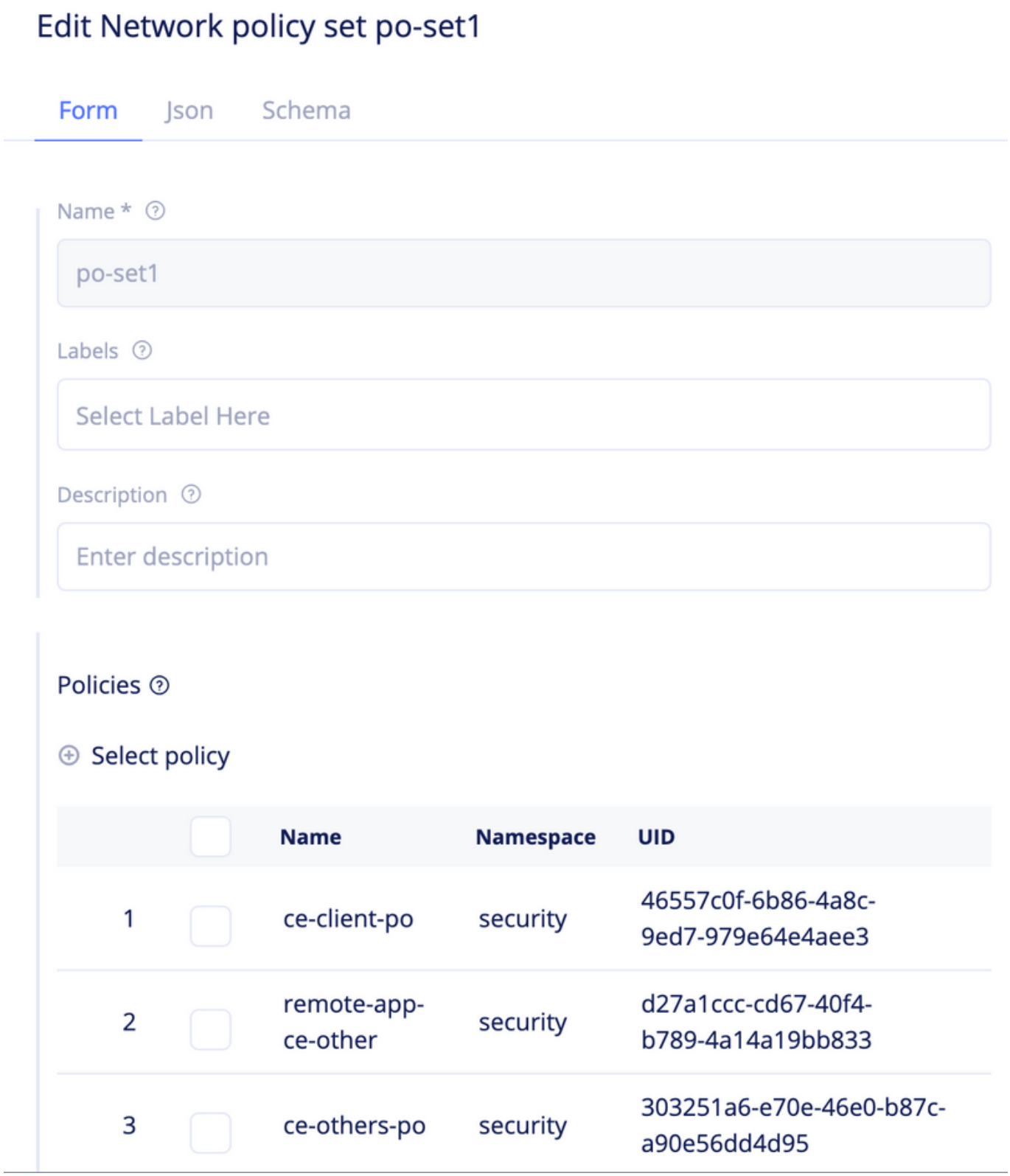
Network Policy を1つ作成します。

- remote-app-ce-other
  - Local Endpoint: Label Selector
  - Selector Expression: app:in(ce-other)
  - Ingress Rules: 1:allow-any
  - Egress Rules: 1: deny-server-dns, 2: allow-any



Network Policy SetにNetwork Policyを追加します

- po-set1
  - Policies: Select policy: [1: ce-client-po
  - 2:remote-app-ce-other, 3: ce-others-po]



ce-otherはremote-app-ce-otherのポリシーがかかっているためserver-appにはcurlできませんが、ce-clientはcurlできることが確認できます。

Exec to container on pod ce-other-64f697b57c-8762c

```
root@ce-other-64f697b57c-8762c:/# curl -m 2 server-app
curl: (28) Connection timed out after 2001 milliseconds
root@ce-other-64f697b57c-8762c:/#
```

Exec to container on pod ce-client-987b76fc8-tgnrg

```
root@ce-client-987b76fc8-tgnrg:/# curl server-app
<!doctype html>
<title>Hello Volterra</title>

<body style="background: #2980b9;"></body>
<div style="color: #e4e4e4;
    text-align:  center;
    height: 90px;
    vertical-align:  middle;">
<h1>
This pod is running on server-app-54c6578975-27nph
</h1>
root@ce-client-987b76fc8-tgnrg:/#
```