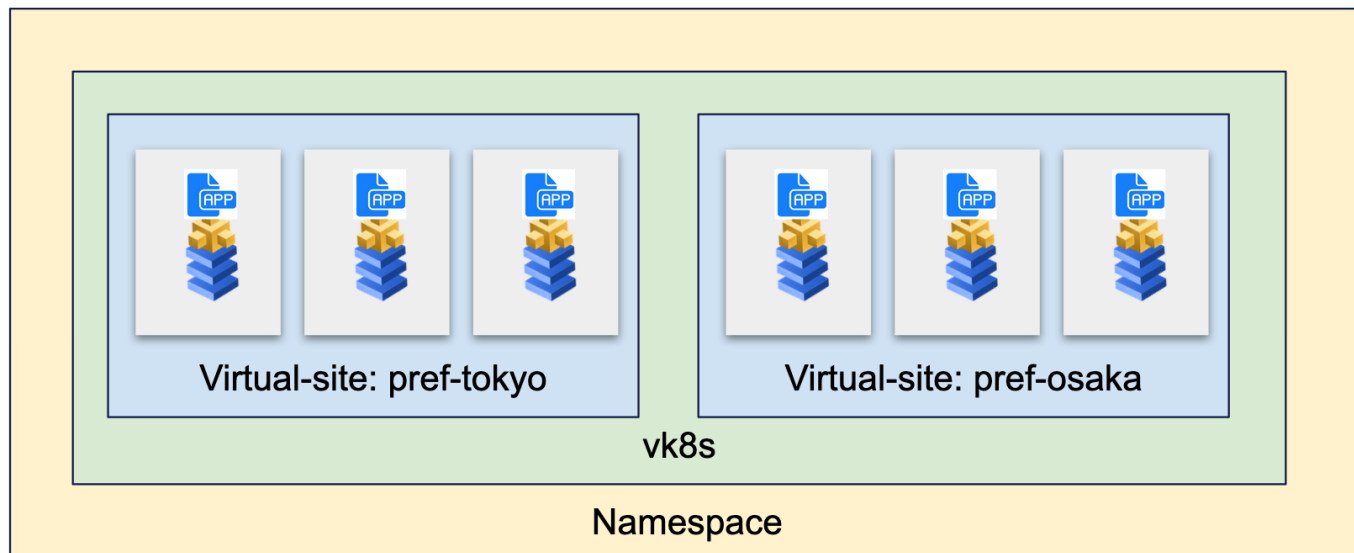


複数virtual-siteの使い方

Namespaceには複数のVirtual-siteを設定できます。例えば pref-tokyoやpref-osakaを作成し、実際のSiteが複数Virtual-site内に存在します。Vk8sを作成すると、所属させるVirtual-siteを選択できます。図にすると以下のような階層型構造になります。



DeploymentやServiceなどのManifestを作成すると、Vk8s内の全てのVirtual siteに反映されます。特定のVirtual-siteだけにManifestを反映させてい場合はAnnotation `ves.io/sites`を使用します。

例えば、virtual-site: pref-tokyoとpref-osakaの2つがありpref-tokyoのみに反映させる場合、Annotationは

```
annotations:
  ves.io/virtual-sites: namespace/pref-tokyo
```

となります。

複数指定する場合は

```
annotations:
  ves.io/virtual-sites: namespace/pref-tokyo,namespace/pref-osaka
```

となります。

複数virtual siteをもつワークロードの作成

namespace:`multi-sites`を作成し、vk8sに2つの以下の2つのVirtual siteを設定します。Name: `pref-tokyo` Site type: `CE` Site Selector Expression: `pref:tokyo`

Name: `pref-osaka` Site type: `CE` Site Selector Expression: `pref:osaka`

- 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。

Form
JSON
Documentation

New: Virtual K8s

Virtual K8s

Metadata

[Create Configuration Specification](#)

Reset All Fields

Metadata *

Reset All Fields ^

Name* ⓘ

Labels ⓘ

Description ⓘ

vk8s

Select Label Here

Enter description

Virtual Sites

Reset All Fields ^

Virtual Sites ⓘ

Name	Namespace	UID
pref-osaka	multi-sites	48fdc638-e5de-4388-8134-e2b2ff095a54
pref-tokyo	multi-sites	b71ce936-65d7-4d9d-882f-0231154c382f

Select vsite ref object

Save and Exit

vk8sに2つのVirtual-site **pref-tokyo**と**pref-osaka**に、Deploymentを作成します。

pref-tokyo

```

apiVersion: apps/v1
metadata:
  name: tokyo-app
  namespace: multi-sites
  annotations:
    ves.io/virtual-sites: multi-sites/pref-tokyo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: tokyo-app
  template:
    metadata:
      labels:
        app: tokyo-app
    spec:
      containers:
        - name: tokyo-app
          image: dnakajima/inbound-app:1.0
          ports:
            - containerPort: 8080

```

pref-osaka

```

apiVersion: apps/v1
metadata:
  name: osaka-app
  namespace: multi-sites

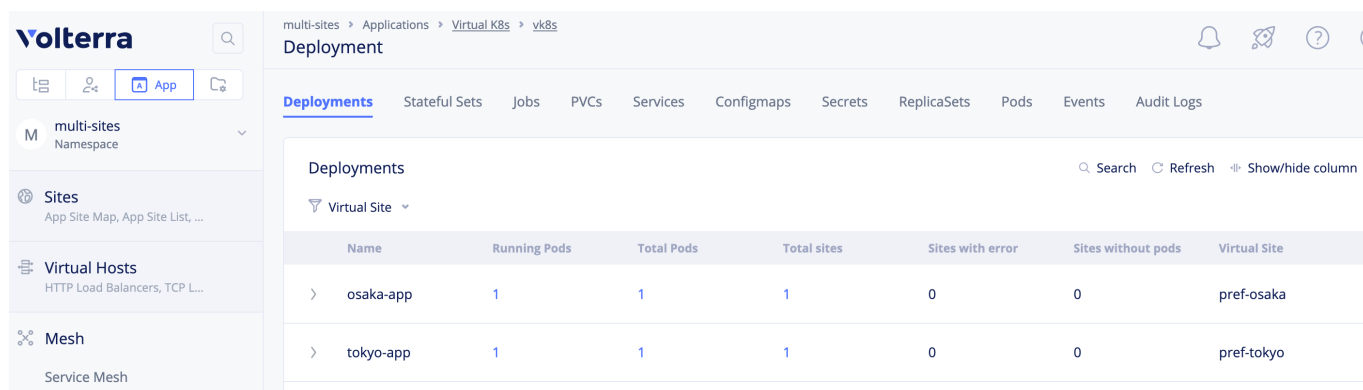
```

```

annotations:
  ves.io/virtual-sites: multi-sites/pref-osaka
spec:
  replicas: 1
  selector:
    matchLabels:
      app: osaka-app
  template:
    metadata:
      labels:
        app: osaka-app
    spec:
      containers:
        - name: osaka-app
          image: dnakajima/inbound-app:1.0
          ports:
            - containerPort: 8080

```

それぞれのVirtual siteにDeploymentが作成されます。



Name	Running Pods	Total Pods	Total sites	Sites with error	Sites without pods	Virtual Site
osaka-app	1	1	1	0	0	pref-osaka
tokyo-app	1	1	1	0	0	pref-tokyo

vk8sに2つのVirtual-site **pref-tokyo**と**pref-osaka**に、Serviceを作成します

pref-tokyo

```

kind: Service
metadata:
  name: tokyo-app
  namespace: multi-sites
  labels:
    app: tokyo-app
  annotations:
    ves.io/virtual-sites: multi-sites/pref-tokyo
spec:
  ports:
    - port: 8080
      targetport: 8080
      protocol: TCP
  selector:
    app: tokyo-app

```

pref-osaka

```

kind: Service
metadata:
  name: osaka-app
  namespace: multi-sites
  labels:
    app: osaka-app
  annotations:
    ves.io/virtual-sites: multi-sites/pref-osaka
spec:
  ports:
  - port: 8080
    targetport: 8080
    protocol: TCP
  selector:
    app: osaka-app

```

Ingress Gatewayの設定

作成した2つのサービスをロードバランスして公開します。

The screenshot shows the Volterra web interface. On the left is a sidebar with navigation options: Sites, Virtual Hosts, and Mesh. The main panel displays the 'Services' tab for the 'multi-sites' namespace. A table lists the following services:

Name	Type	Endpoints	Ports	Total sites	Virtual Site	Age
osaka-app	ClusterIP	1	8080:8080/TCP		pref-osaka	<1m
tokyo-app	ClusterIP	1	8080:8080/TCP		pref-tokyo	<1m

Origin poolの作成

作成したワークロードをそれぞれ**tokyo-app**と**osaka-app**としてOrigin-poolに登録します。

- Name: **tokyo-app**
- Select Type of Origin Server: **k8sService Name of Origin Ser...**
- Service Name: **tokyo-app.multi-sites**を入力します。(kubernetes service名.namespaceのフォーマット)
- Select Site or Virtual Site: **Virtual Site -> multi-sites/pref-tokyo**
- Select Network on the Site: **Vk8s Networks on Site**
- Port: **8080**
- Name: **osaka-app**

- Select Type of Origin Server: **k8sService Name of Origin Ser...**
- Service Name: **osaka-app.multi-sites**を入力します。(kubernetes service名.namespaceのフォーマット)
- Select Site or Virtual Site: **Virtual Site -> multi-sites/pref-osaka**
- Select Network on the Site: **Vk8s Networks on Site**
- Port: **8080**

The screenshot shows the Volterra web interface for configuring an Origin Pool. The left sidebar contains a navigation menu with options like 'Origin Pool', 'Metadata', 'Basic Configuration', 'List of Health Check(s)', and 'TLS Configuration'. The main area is titled 'Basic Configuration *' and includes the following fields:

- Origin Servers *:** A dropdown menu set to 'k8s Service Name of Origin Server on giv...'. Below it, a text field contains 'tokyo-app.multi-sites'.
- Select Site or Virtual Site *:** A dropdown menu set to 'Virtual Site'. Below it, a text field contains 'multi-sites/pref-tokyo'.
- Select Network on the site *:** A dropdown menu set to 'Vk8s Networks on Site'.
- Port:** A text field containing '8080'.
- LoadBalancer Algorithm:** A dropdown menu set to 'Round Robin'.
- Endpoint Selection:** A dropdown menu set to 'Local Endpoints Preferred'.

At the bottom, there is a table titled 'multi-sites > Manage Origin Pools' with the following data:

Name	Namespace
osaka-app	multi-sites
tokyo-app	multi-sites

HTTP loadbalancerの作成

Manage -> HTTP Load Balancers で "Add HTTP load balancer"を選択します。

- Name: multi-vsited-lb
- Domains: dummy.localhost (設定するとDNS infoにVolterraからdomain名が払い出されます。設定後に払い出されたドメイン名を設定してください。)
- Select Type of Load Balancer: **HTTP**
- Default Origin Pools: 2つのOrigin poolを設定します。Weightは100,100にしていますが、比率を変えることで、ローバランスレシオを調節できます。

Form

JSON

Documentation

New: HTTP Load Balancer

HTTP Load Balancer

Metadata

Basic Configuration

Default Origin Servers

Routes Configuration

VIP Configuration

Security Configuration

Advanced Configuration

Metadata *

Reset All Fields

Name* ⓘ

multi-vs-site-lb

Labels ⓘ

Select Label Here

Description ⓘ

Enter description

Basic Configuration *

Reset All Fields

Domains* ⓘ

dummy.localhost

⊕ Add Item

Select Type of Load Balancer* ⓘ

HTTP

Default Origin Servers

Origin Pools ⓘ

Configure ▶

Form

JSON

Documentation

Edit: multi-vs-site-lb

HTTP Load Balancer

Origin Pools

Origin Pools

Reset All Fields Show Advanced Fields

Origin Pools ⓘ

Select Origin Pool Method* ⓘ

Origin Pool

Origin Pool ⓘ

multi-sites/osaka-app

Weight ⓘ

100

Select Origin Pool Method* ⓘ

Origin Pool

Origin Pool ⓘ

multi-sites/tokyo-app

Weight ⓘ

100

⊕ Add Item

Cancel and Exit

Curlなどで確認すると、tokyo-app, osaka-appでロードバランスされることが確認できます。

```
<html>
<body>
This pod is running on tokyo-app-767948955-jpbnx
</body>
</html>

curl http://ves-io-3b89b61f-b82b-4140-915a-96f56818fd56.ac.vh.ves.io/
<html>
<body>
```

```
This pod is running on osaka-app-7cc7958f77-2x4br  
</body>  
</html>
```