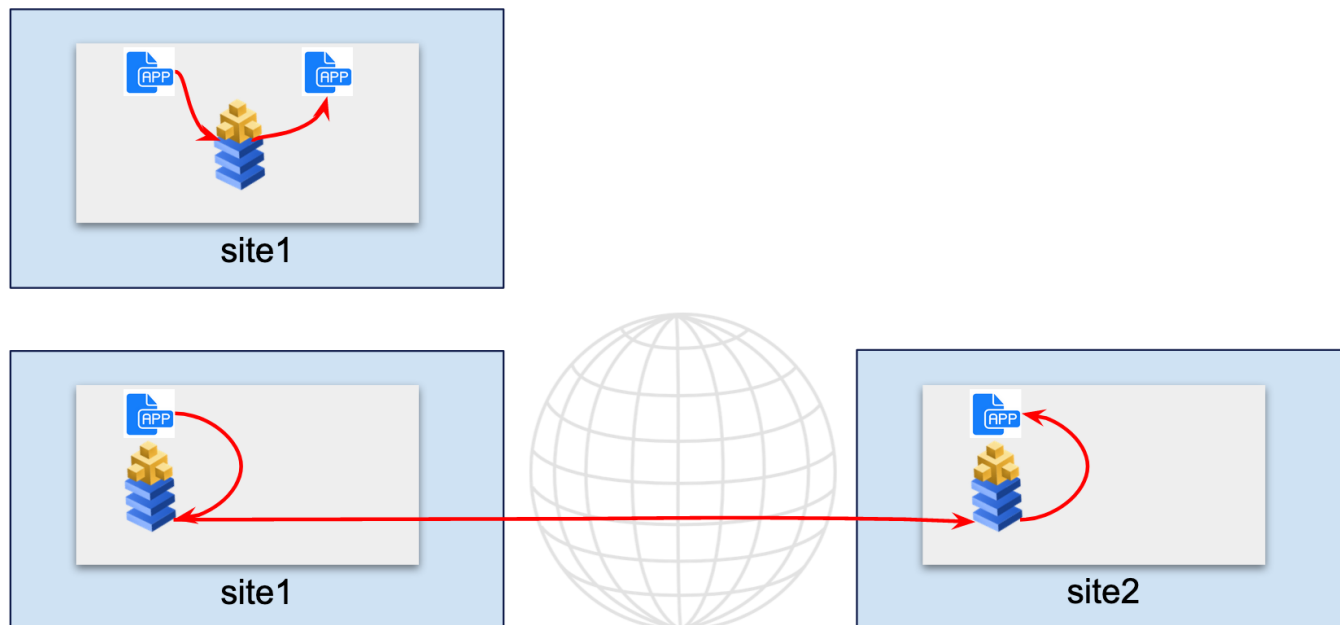


# App to App接続

同一Site内では通常のKubernetesと同様にPodはService経由で他のPodにアクセスできます。異なるSiteのServiceにアクセスする場合はHTTP/TCP Loadbalancerの設定が必要です。



## vk8s manifest の作成

namespace: **app-app**を作成し、vk8sに以下の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 \*

Name\*
vk8s

Labels
Select Label Here

Description
Enter description

Reset All Fields

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。

1 / 6

## pref-tokyo

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

## pref-osaka

```
apiVersion: apps/v1
metadata:
  name: osaka-app
  namespace: app-app
  annotations:
    ves.io/virtual-sites: app-app/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

app-app > Applications > Virtual K8s > vk8s

Deployment

Deployments Stateful Sets Jobs PVCs Services Configmaps Secrets ReplicaSets Pods Events Audit Logs

+ Add Deployment Search Refresh Show/hide column

Virtual Site

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

vk8sにVirtual-site:pref-osakaにserviceを作成します

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

app-app > Applications > Virtual K8s > vk8s

Service

Deployments Stateful Sets Jobs PVCs Services Configmaps Secrets ReplicaSets Pods Events Audit Logs

+ Add Service Search Refresh Show/hide column

Virtual Site

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

## Ingress gatewayの作成

### origin poolの作成

作成したosaka-appワークロードをOrigin-poolに登録します。

- Name: **osaka-app**
- Basic Configuration: "Select Type of Origin Server"は**k8sService Name of Origin Ser...**を選択します。
- Service Name: **osaka-app.app-app**
- Select Site or Virtual Site: **Virtual Site**を選択し、**app-app/pref-osaka**を指定します。
- Select Network on the Site: **Vk8s Networks on Site**を指定します。

- Port: **8080**を設定します。

[Form](#)
[JSON](#)
[Documentation](#)

Edit: **osaka-app**

**Origin Pool**

- Metadata
- Basic Configuration
- List of Health Check(s)
- TLS Configuration

### Basic Configuration \*

Origin Servers \* ⓘ

Select Type of Origin Server \* ⓘ

k8s Service Name of Origin Server on giv... ▼

Service Name\* ⓘ

osaka-app.app-app

Select Site or Virtual Site \* ⓘ

Virtual Site ▼

Virtual Site ⓘ

app-app/pref-osaka × ▼

Select Network on the site \* ⓘ

Vk8s Networks on Site ▼

⊕ Add item

Port ⓘ
8080 ⓘ

LoadBalancer Algorithm ⓘ
Round Robin × ▼

Endpoint Selection ⓘ
Local Endpoints Preferred × ▼

Reset All Fields ^

## HTTP loadbalancerの作成

HTTP loadbalancerを作成し、Origin poolを設定します。ここで設定したdomain名(osaka-app-1)はk8s上のpodに登録され、Podがリモートサービスに接続するのに使用します。

Name: **osaka-app-lb** Domains: **osaka-app-1** Select Type of Load Balancer: **http** Default Origin Servers: **app-app/osaka-app** VIP Configuration: VIP Configuration を有効化し、**Advertise Custom** を選択し **Configure**を選択 Select Where to Advertise: **virtual-site** Site Network: **Inside and Outside Network** Virtual Site Reference: **app-app/pref-tokyo**

Form JSON Documentation

Edit: osaka-app-lb

**HTTP Load Balancer**

- Metadata
- Basic Configuration
- Default Origin Servers
- Routes Configuration
- VIP Configuration
- Security Configuration
- Advanced Configuration

**Default Origin Servers** Reset All Fields

Origin Pools Edit

**Routes Configuration** Show Advanced Fields

**VIP Configuration** Reset All Fields Show Advanced Fields

Where to Advertise the VIP \*

Advertise Custom

Advertise Custom Edit

Form JSON Documentation

Edit: osaka-app-lb

HTTP Load Balancer

**Advertise Custom**

Custom Advertise VIP Configuration

**Custom Advertise VIP Configuration \*** Reset All Fields Show Advanced Fields

List of Sites to Advertise \*

Select Where to Advertise \*

Virtual Site

Site Network \* × Virtual Site Reference \* ×

Inside and Outside Network × app-app/pref-tokyo ×

TCP Listen Port Choice \*

Use Default TCP Listen Port

+ Add item

Back Apply

## アクセス確認

podへのアクセスはvk8sのPodからコンテナのコンソールにアクセスできます。app-clientの**Exec to Container**を選択します

**Volterra**

app-app > Applications > Virtual K8s > vk8s

Pod

Deployments Stateful Sets Jobs PVCs Services Configmaps Secrets ReplicaSets **Pods** Events Audit Logs

+ Add Pod Search Refresh Show/hide column Last 5 minutes

Site Virtual Site Deployment (1) Status (1)

Node name	Restarts	Ready	Status	Virtual Site	Age	Resource
b75- site1-master-0	0	3/3	Running	pref-tokyo	<1m	CPU 0 Memory 0b

Show Events  
Show Logs  
Exec To Container  
Delete

tokyo-appを選択し、bashを入力してConnectからコンソールに接続します。dig osaka-app-1やcurl osaka-app-1を入力すると、接続用のIPアドレスや、実際のosaka-appへの接続が確認できます。

Pod: app-client-6cb7755b75-zqghk

✕

Events Logs **Terminal** Resource Utilization

Container to exec to \*

app-client

✕

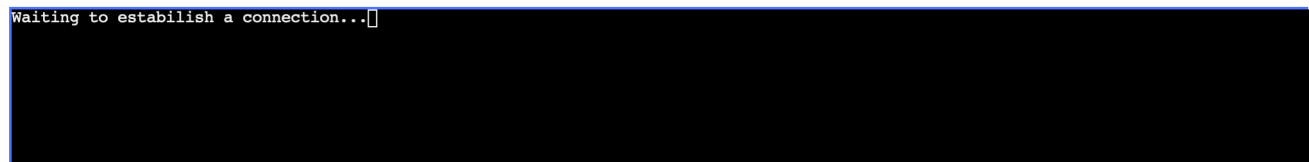
▼

Command to execute\*

bash

Connect

Terminal font size ▾

Waiting to establish a connection...

Exec to container on pod app-client-76dd4b7898-27qnw

```
root@app-client-76dd4b7898-27qnw:/# dig osaka-app
; <<>> DiG 9.11.3-lubuntu1.12-Ubuntu <<>> osaka-app
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17758
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;osaka-app.                IN      A

;; ANSWER SECTION:
osaka-app.                3600    IN      A      192.168.2.200

;; Query time: 8 msec
;; SERVER: 100.127.192.10#53(100.127.192.10)
;; WHEN: Tue Jul 28 08:24:55 UTC 2020
;; MSG SIZE rcvd: 52

root@app-client-76dd4b7898-27qnw:/# curl osaka-app
<html>
<body>
This pod is running on osaka-app-7cc7958f77-2x4br
</body>
</html>

root@app-client-76dd4b7898-27qnw:/#
```