实验一：使用statefulset部署应用

1.部署持久卷

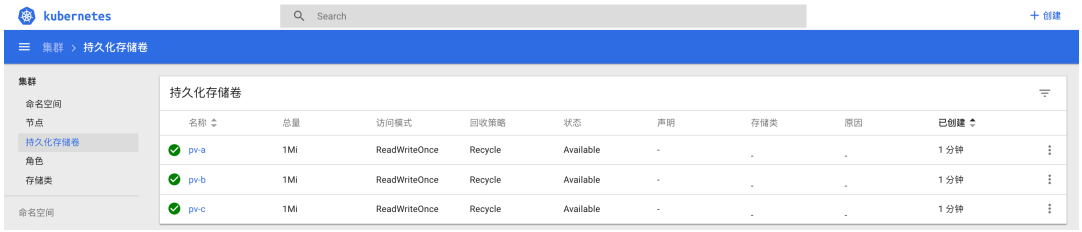
persistent-volumes-hostpath.yaml

```
kind: List
apiVersion: v1
items:
- apiVersion: v1
  kind: PersistentVolume
  metadata:
    name: pv-a
  spec:
    capacity:
      storage: 1Mi
    accessModes:
      - ReadWriteOnce
    persistentVolumeReclaimPolicy: Recycle
    hostPath:
      path: /tmp/pv-a
- apiVersion: v1
  kind: PersistentVolume
  metadata:
    name: pv-b
  spec:
    capacity:
      storage: 1Mi
    accessModes:
      - ReadWriteOnce
    persistentVolumeReclaimPolicy: Recycle
    hostPath:
      path: /tmp/pv-b
- apiVersion: v1
  kind: PersistentVolume
  metadata:
```

```
    name: pv-c
  spec:
    capacity:
      storage: 1Mi
    accessModes:
      - ReadWriteOnce
    persistentVolumeReclaimPolicy: Recycle
    hostPath:
      path: /tmp/pv-c
```

kubectl create -f persistent-volumes-hostpath.yaml

```
MacBook-Pro-2:chapter10 sunxi$ kubectl create -f persistent-volumes-hostpath.yaml
persistentvolume/pv-a created
persistentvolume/pv-b created
persistentvolume/pv-c created
MacBook-Pro-2:chapter10 sunxi$
```



2.创建headless service为有状态的pod之间提供网络标识

```
MacBook-Pro-2:chapter10 sunxi$ cat kubia-service-headless.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  clusterIP: None
  selector:
    app: kubia
  ports:
  - name: http
    port: 80
MacBook-Pro-2:chapter10 sunxi$
```

kubectl create -f kubia-service-headless.yaml

```
MacBook-Pro-2:chapter10 sunxi$ kubectl create -f kubia-service-headless.yaml
service/kubia created
MacBook-Pro-2:chapter10 sunxi$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   2d2h
kubia        ClusterIP   None         <none>        80/TCP    38s
MacBook-Pro-2:chapter10 sunxi$
```

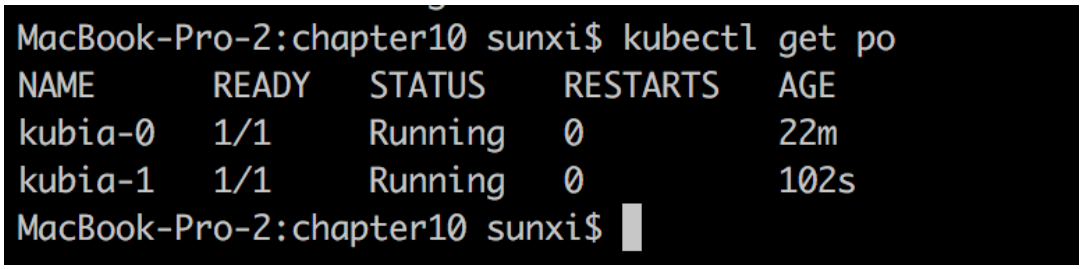
3.创建statefulset

```
MacBook-Pro-2:chapter10 sunxi$ cat kubia-statefulset.yaml
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: kubia
spec:
  serviceName: kubia
  replicas: 2
  template:
    metadata:
      labels:
        app: kubia
    spec:
      containers:
      - name: kubia
        image: luksa/kubia-pet
        ports:
        - name: http
          containerPort: 8080
        volumeMounts:
        - name: data
          mountPath: /var/data
  volumeClaimTemplates:
  - metadata:
      name: data
    spec:
      resources:
        requests:
          storage: 1Mi
      accessModes:
      - ReadWriteOnce
MacBook-Pro-2:chapter10 sunxi$
```

kubectl create -f kubia-statefulset.yaml

kubectl get po

```
MacBook-Pro-2:chapter10 sunxi$ kubectl create -f kubia-statefulset.yaml
statefulset.apps/kubia created
MacBook-Pro-2:chapter10 sunxi$ kubectl get po
NAME      READY   STATUS              RESTARTS   AGE
kubia-0   0/1     ContainerCreating   0          26s
```



kubectl get po kubia-0 -o yaml

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2020-06-20T14:57:05Z"
  generateName: kubia-
  labels:
    app: kubia
    controller-revision-hash: kubia-c94bcb69b
    statefulset.kubernetes.io/pod-name: kubia-0
  name: kubia-0
  namespace: default
  ownerReferences:
  - apiVersion: apps/v1
    blockOwnerDeletion: true
    controller: true
    kind: StatefulSet
    name: kubia
    uid: f339a6de-2753-4b4a-921e-c38a625b739d
  resourceVersion: "65292"
  selfLink: /api/v1/namespaces/default/pods/kubia-0
  uid: 096c32c4-a8fe-49eb-b336-655c0fd86d48
spec:
  containers:
  - image: luksa/kubia-pet
    imagePullPolicy: Always
    name: kubia
    ports:
    - containerPort: 8080
      name: http
      protocol: TCP
    resources: {}
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /var/data
      name: data
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: default-token-xp252
      readOnly: true
```

```
volumes:
- name: data
  persistentVolumeClaim:
    claimName: data-kubia-0
- name: default-token-xp252
  secret:
    defaultMode: 420
    secretName: default-token-xp252
```

4.查看pvc

kubectl get pvc

```
MacBook-Pro-2:chapter10 sunxi$ kubectl get pvc
NAME           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-kubia-0   Bound    pvc-68bd27f0-eb44-4ab5-ac35-e56611fbb1b5   1Mi        RWO            standard       27m
data-kubia-1   Bound    pvc-2a38b097-d2e6-4a57-8b2b-6105bfd722fa   1Mi        RWO            standard       6m50s
MacBook-Pro-2:chapter10 sunxi$
```

问题：为什么pvc挂不上pv?

比较两种pv的区别：

书中的PersistentVolume	minikube自动生成的Per
{ "kind": "PersistentVolume", "apiVersion": "v1", "metadata": { "name": "pv-a", "selfLink": "/api/v1/persistentvolumes/pv-a", "uid": "e6fbe169-88b6-4e04-8c18-74ac080d6e02", "resourceVersion": "83356", "creationTimestamp": "2020-06-	{ "kind": "PersistentVolu "apiVersion": "v1", "metadata": { "name": "pvc-cb9ebc 4f9e-a91d-fed29748c5 "selfLink": "/api/v1/persistentvolum cb9ebc62-6911-4f9e-a fed29748c5dd", "uid": "13c7f791-e52 39174deb4eee", "resourceVersion": "6 "creationTimestamp": 20T16:16:21Z",

```
    "creationTimestamp": "2020-06-
20T14:43:40Z",
    "annotations": {
      "pv.kubernetes.io/bound-by-
controller": "yes"
    },
    "finalizers": [
      "kubernetes.io/pv-protection"
    ]
  },
  "spec": {
    "capacity": {
      "storage": "1Mi"
    },
    "hostPath": {
      "path": "/tmp/pv-a",
      "type": ""
    },
    "accessModes": [
      "ReadWriteOnce"
    ],
    "claimRef": {
      "kind": "PersistentVolumeClaim",
      "namespace": "default",
      "name": "data-kubia-test-0",
      "uid": "e9e4bb27-fc9c-4de0-
9599-9af0d83ac749",
      "apiVersion": "v1",
      "resourceVersion": "83353"
    },
    "persistentVolumeReclaimPolicy":
"Recycle",
    "volumeMode": "Filesystem"
  },
  "status": {
    "phase": "Bound"
  }
}
```

```
    "annotations": {
      "hostPathProvisione
"0e7090d0-b15d-11ea-
08002719be7b",
      "pv.kubernetes.io/pr
by": "k8s.io/minikube-h
    },
    "finalizers": [
      "kubernetes.io/pv-p
    ]
  },
  "spec": {
    "capacity": {
      "storage": "1Mi"
    },
    "hostPath": {
      "path": "/tmp/hostpa
provisioner/pvc-cb9ebc
4f9e-a91d-fed29748c5
      "type": ""
    },
    "accessModes": [
      "ReadWriteOnce"
    ],
    "claimRef": {
      "kind": "PersistentVc
      "namespace": "defa
      "name": "data-kubia
      "uid": "cb9ebc62-69
a91d-fed29748c5dd",
      "apiVersion": "v1",
      "resourceVersion": "
    },
    "persistentVolumeRe
"Delete",
    "storageClassName":
    "volumeMode": "Files
  },
  "status": {
    "phase": "Bound"
```

```
    phase": "Bound"
  }
}
```

修改statefulset的配置文件：

```
MacBook-Pro-2:chapter10 sunxi$ cat kubia-test-statefulset.yaml
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: kubia-test
spec:
  serviceName: kubia-test
  replicas: 3
  template:
    metadata:
      labels:
        app: kubia-test
    spec:
      containers:
      - name: kubia
        image: luksa/kubia-pet-peers
        ports:
        - name: http
          containerPort: 8080
        volumeMounts:
        - name: data
          mountPath: /var/data
  volumeClaimTemplates:
  - metadata:
      name: data
    spec:
      resources:
        requests:
          storage: 1Mi
      accessModes:
      - ReadWriteOnce
      storageClassName: ""
```

按照上面的步骤创建headless service和statefulset：

```
MacBook-Pro-2:chapter10 sunxi$ kubectl get svc
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP   3d15h
kubia          ClusterIP   None            <none>        80/TCP    36h
kubia-public   ClusterIP   10.111.255.12   <none>        80/TCP    35h
kubia-test     ClusterIP   None            <none>        80/TCP    7m50s
MacBook-Pro-2:chapter10 sunxi$ kubectl get sts
NAME         READY   AGE
kubia        3/3     36h
kubia-test   3/3     8m2s
MacBook-Pro-2:chapter10 sunxi$ kubectl get po
NAME           READY   STATUS    RESTARTS   AGE
kubia-0        1/1     Running   0          35h
kubia-1        1/1     Running   0          35h
kubia-2        1/1     Running   0          35h
kubia-test-0   1/1     Running   0          8m5s
kubia-test-1   1/1     Running   0          7m38s
kubia-test-2   1/1     Running   0          7m17s
MacBook-Pro-2:chapter10 sunxi$ kubectl get pvc
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-kubia-0        Bound    pvc-68bd27f0-eb44-4ab5-ac35-e56611fbb1b5   1Mi        RWO            standard       36h
data-kubia-1        Bound    pvc-2a38b097-d2e6-4a57-8b2b-6105bfd722fa   1Mi        RWO            standard       36h
data-kubia-2        Bound    pvc-cb9ebc62-6911-4f9e-a91d-fed29748c5dd   1Mi        RWO            standard       35h
data-kubia-test-0   Bound    pv-a                                       1Mi        RWO                           8m8s
data-kubia-test-1   Bound    pv-b                                       1Mi        RWO                           7m41s
data-kubia-test-2   Bound    pv-c                                       1Mi        RWO                           7m20s
MacBook-Pro-2:chapter10 sunxi$
```

storageClassName：此配置用于绑定PVC和PV。这表明这个PVC希望使用`storageClassName=""`的PV

实验二：通过API服务器与pod通信

API服务器的作用：通过代理直接连接到指定的pod

1.运行代理

kubectl proxy

```
MacBook-Pro-2:chapter10 sunxi$ kubectl proxy
Starting to serve on 127.0.0.1:8001
```

2.发送请求

GET请求：

curl localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/

```
MacBook-Pro-2:~ sunxi$ curl localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/
You've hit kubia-0
Data stored on this pod: No data posted yet
MacBook-Pro-2:~ sunxi$
```

通过API服务器与pod通信，每个请求都会经过两个代理：

（1）kubectl代理

（2）将请求代理到pod的API服务器

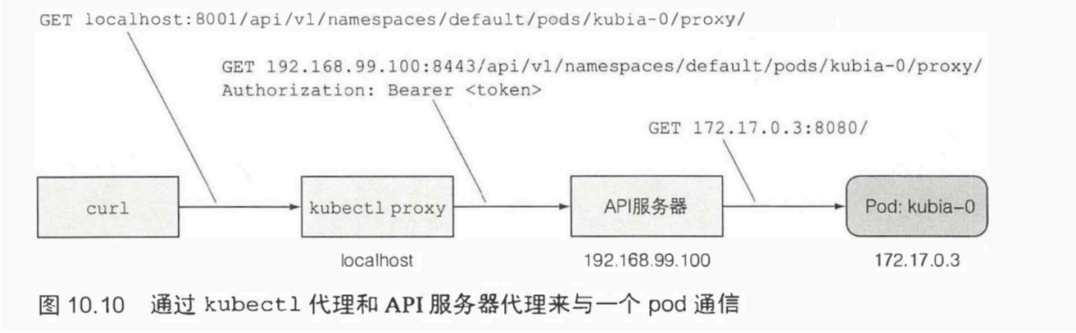


图 10.10　通过 kubectl 代理和 API 服务器代理来与一个 pod 通信

POST请求：

curl -XPOST -d "Hey there! This greeting was submitted to kubia-0." localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/

```
MacBook-Pro-2:~ sunxi$ curl -XPOST -d "Hey there! This greeting was submitted to kubia-0." localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/
Data stored on pod kubia-0
MacBook-Pro-2:~ sunxi$
```

curl localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/

```
MacBook-Pro-2:~ sunxi$ curl localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/
You've hit kubia-0
Data stored on this pod: Hey there! This greeting was submitted to kubia-0.
MacBook-Pro-2:~ sunxi$
```

curl localhost:8001/api/v1/namespaces/default/pods/kubia-1/proxy/

```
MacBook-Pro-2:~ sunxi$ curl localhost:8001/api/v1/namespaces/default/pods/kubia-1/proxy/
You've hit kubia-1
Data stored on this pod: No data posted yet
MacBook-Pro-2:~ sunxi$
```

kubectl exec -it kubia-0 bash

```
MacBook-Pro-2:~ sunxi$ kubectl exec -it kubia-0 bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl kubectl exec [POD] -- [COMMAND] instead.
root@kubia-0:/# pwd
/
root@kubia-0:/# cd /var/data
root@kubia-0:/var/data# ls -l
total 4
-rw-r--r-- 1 root root 50 Jun 20 15:36 kubia.txt
root@kubia-0:/var/data# cat kubia.txt
Hey there! This greeting was submitted to kubia-0.root@kubia-0:/var/data#
```

实验三：删除一个有状态pod检查重新调度的pod是否关联了相同的存储

kubectl delete po kubia-0

```
MacBook-Pro-2:~ sunxi$ kubectl delete po kubia-0
pod "kubia-0" deleted
MacBook-Pro-2:~ sunxi$
```

```
MacBook-Pro-2:~ sunxi$ kubectl get po
NAME      READY   STATUS        RESTARTS   AGE
kubia-0   0/1     Terminating   0          47m
kubia-1   1/1     Running       0          26m
MacBook-Pro-2:~ sunxi$ kubectl get po
NAME      READY   STATUS              RESTARTS   AGE
kubia-0   0/1     ContainerCreating   0          1s
kubia-1   1/1     Running             0          26m
MacBook-Pro-2:~ sunxi$ kubectl get po
NAME      READY   STATUS              RESTARTS   AGE
kubia-0   0/1     ContainerCreating   0          2s
kubia-1   1/1     Running             0          26m
MacBook-Pro-2:~ sunxi$ kubectl get po
NAME      READY   STATUS              RESTARTS   AGE
kubia-0   0/1     ContainerCreating   0          12s
kubia-1   1/1     Running             0          26m
MacBook-Pro-2:~ sunxi$ kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
kubia-0   1/1     Running   0          45s
kubia-1   1/1     Running   0          27m
MacBook-Pro-2:~ sunxi$
```

curl localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/

```
MacBook-Pro-2:~ sunxi$ curl localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/
You've hit kubia-0
Data stored on this pod: Hey there! This greeting was submitted to kubia-0.
MacBook-Pro-2:~ sunxi$
```

实验四：创建一个普通的非headless的service暴露statefulset的pod

1.创建service

```
MacBook-Pro-2:chapter10 sunxi$ cat kubia-service-public.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia-public
spec:
  selector:
    app: kubia
  ports:
  - port: 80
    targetPort: 8080
MacBook-Pro-2:chapter10 sunxi$
```

kubectl create -f kubia-service-public.yaml

```
MacBook-Pro-2:chapter10 sunxi$ kubectl get svc
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP   2d3h
kubia          ClusterIP   None            <none>        80/TCP    62m
kubia-public   ClusterIP   10.111.255.12   <none>        80/TCP    7s
MacBook-Pro-2:chapter10 sunxi$
```

2.通过API服务器访问集群内部的服务

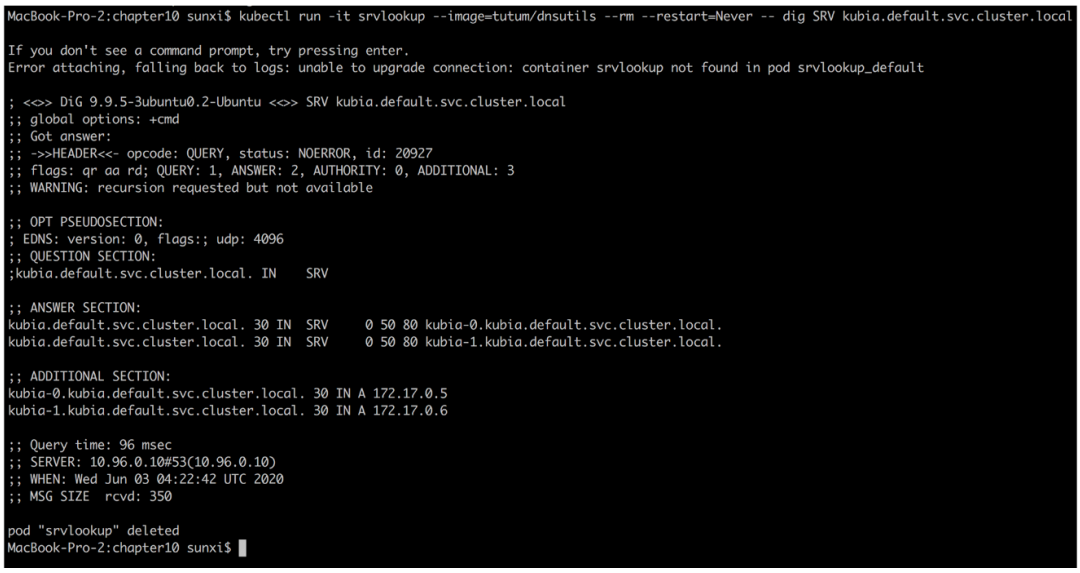
curl localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/

```
MacBook-Pro-2:chapter10 sunxi$ curl localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
You've hit kubia-1
Data stored on this pod: No data posted yet
MacBook-Pro-2:chapter10 sunxi$ curl localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
You've hit kubia-0
Data stored on this pod: Hey there! This greeting was submitted to kubia-0.
MacBook-Pro-2:chapter10 sunxi$
```

实验五：SVR记录

SVR记录用来指向提供服务的服务器的主机名和端口号，k8s通过一个headless service创建SVR记录来指向pod的主机名。

kubectl run -it srvlookup --image=tutum/dnsutils --rm --restart=Never -- dig SRV kubia.default.svc.cluster.local



实验六：通过DNS实现伙伴间彼此发现

1.更新statefulset

kubectl edit statefulset kubia

修改副本数为3

修改image为：luksa/kubia-pet-peers

```
MacBook-Pro-2:chapter10 sunxi$ kubectl get po
NAME      READY   STATUS              RESTARTS   AGE
kubia-0   1/1     Running             0          32m
kubia-1   1/1     Running             0          58m
kubia-2   0/1     ContainerCreating   0          4s
```

```
MacBook-Pro-2:chapter10 sunxi$ kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
kubia-0   1/1     Running   0          33m
kubia-1   1/1     Running   0          59m
kubia-2   1/1     Running   0          80s
MacBook-Pro-2:chapter10 sunxi$
```

2.手动删除kubia-0、kubia-1使得statefulset根据新的pod模板重新调度启动它们

kubectl delete po kubia-0 kubia-1

```
MacBook-Pro-2:chapter10 sunxi$ kubectl delete po kubia-0 kubia-1
pod "kubia-0" deleted
pod "kubia-1" deleted
MacBook-Pro-2:chapter10 sunxi$ kubectl get po
NAME      READY   STATUS              RESTARTS   AGE
kubia-0   0/1     ContainerCreating   0          7s
kubia-2   1/1     Running             0          2m27s
```

```
MacBook-Pro-2:chapter10 sunxi$ kubectl get po
NAME      READY   STATUS              RESTARTS   AGE
kubia-0   1/1     Running             0          41s
kubia-1   0/1     ContainerCreating   0          20s
kubia-2   1/1     Running             0          3m1s
MacBook-Pro-2:chapter10 sunxi$
```

```
MacBook-Pro-2:chapter10 sunxi$ kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
kubia-0   1/1     Running   0          62s
kubia-1   1/1     Running   0          41s
kubia-2   1/1     Running   0          3m22s
MacBook-Pro-2:chapter10 sunxi$
```

3.写数据

curl localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/

```
MacBook-Pro-2:chapter10 sunxi$ curl localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
You've hit kubia-1
Data stored in the cluster:
- kubia-1.kubia.default.svc.cluster.local: No data posted yet
- kubia-0.kubia.default.svc.cluster.local: Hey there! This greeting was submitted to kubia-0.
- kubia-2.kubia.default.svc.cluster.local: No data posted yet
MacBook-Pro-2:chapter10 sunxi$
```

curl -XPOST -d "The sun is shining." localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/

```
MacBook-Pro-2:chapter10 sunxi$ curl -XPOST -d "The sun is shining." localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
Data stored on pod kubia-2
MacBook-Pro-2:chapter10 sunxi$ curl -XPOST -d "The sun is sweet." localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
Data stored on pod kubia-2
MacBook-Pro-2:chapter10 sunxi$ curl -XPOST -d "The sun is kitty." localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
Data stored on pod kubia-2
MacBook-Pro-2:chapter10 sunxi$ curl -XPOST -d "The sun is 1." localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
Data stored on pod kubia-1
MacBook-Pro-2:chapter10 sunxi$ curl -XPOST -d "The sun is 2." localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
Data stored on pod kubia-0
MacBook-Pro-2:chapter10 sunxi$
```

curl localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/

```
MacBook-Pro-2:chapter10 sunxi$ curl localhost:8001/api/v1/namespaces/default/services/kubia-public/proxy/
You've hit kubia-2
Data stored in the cluster:
- kubia-0.kubia.default.svc.cluster.local: The sun is 2.
- kubia-1.kubia.default.svc.cluster.local: The sun is 1.
- kubia-2.kubia.default.svc.cluster.local: The sun is kitty.
MacBook-Pro-2:chapter10 sunxi$
```