namespace별 helm 설치하기

1. 개요

cluster 당 하나의 tiller 가 설치되면 네임스페이별로 격리되지 않아 다른 네임스페이스의 helm list 가 모두 공유 될 뿐만 아니라 upgrade, delete 등의 명령어를 다른 네임스페이스에서 사용할 수 있어서 보안상 취약하다.

그러므로 각 프로젝트 별로 업무격리가 될 수 있도록 namespace 안에서 각각 helm 설치하는 방법을 알아보자.

테스트 환경은 다음과 같이 millet-admin 계정이며 현재 default 권한이 최소화 되어 있는 상태이다.

항목	값	비고
os계정	milletos	
oc계정	millet-admin	
namespace	millet	

2. helm 실행파일 설정

helm실행파일은 현재는 root권한으로만 실행되도록 되어 있다. milletos 계정으로 helm 을 실행할 수 있도록 helm 실행파일을 복사 및 권한 설정해야 한다.

```
# millet 권한으로 수행
$ cd /home/milletos/
$ mkdir bin

# root 권한으로 수행
$ cp /root/linux-amd64/helm /home/milletos/bin/
$ chown millet /home/milletos/bin/helm
$ chgrp millet /home/milletos/bin/helm
```

3. service account [tiller] 생성

tiller라는 service account 로 helm tiller 를 실행하므로 sa 를 생성해야 한다.

```
oc -n millet create sa tiller
```

4. rolebinding

tiller 라는 SA 와 ClusterRole 를 binding 한다.

```
$ cat > tiller_rolebinding.yaml
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: tiller_rolebinding
  namespace: millet
subjects:
  - kind: ServiceAccount
    name: tiller
    namespace: millet
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
---

$ oc create -f tiller_rolebinding.yaml
```

5. helm 설치(server, client)

helm init 명령으로 client 와 server 에 각각 helm 을 설치한다.

5.1 설치 방법

아래와 같이 tiller 와 tiller-namespace 를 지정하고 init 명령으로 설치를 진행할 수 있다.

```
$ helm init --service-account tiller --tiller-namespace millet
```

하지만 tiller 라는 image 경로가 맞지 않아 사내에서는 설치되지 않는다. 그러므로 아래와 같이 client 와 server 에 설치하는 방법을 각각 실행해야 한다.

5.2. client 설치

client 의 경우 `.helm` 에 환경 설정하는 역할을 수행한다.

```
$ helm init --client-only
```

5.3. server tiller 설치 방법

server 의 경우 millet 네임스페이스에 deployment 와 service 를 설치한다.

image 경로가 맞지 않아 설치가 되지 않으므로 아래와 같이 dry-run 으로 실행가능한 yaml 파일을 catch 하여 유효한 image 값으로 update 후 실행한다.

```
$ helm init --service-account tiller --tiller-namespace millet --dry-run --debug >
helm_init.yaml
```

image: ktis-bastion01.container.ipc.kt.com:5000/openshift/tiller:v2.9.0

```
$ cat helm_init.yaml
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: helm
    name: tiller
  name: tiller-deploy
  namespace: millet
spec:
  replicas: 1
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: helm
        name: tiller
    spec:
      automountServiceAccountToken: true
      containers:
      - env:
        - name: TILLER_NAMESPACE
          value: millet
        - name: TILLER_HISTORY_MAX
          value: "0"
        image: ktis-bastion01.container.ipc.kt.com:5000/openshift/tiller:v2.9.0
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
            path: /liveness
```

```
            port: 44135
          initialDelaySeconds: 1
          timeoutSeconds: 1
        name: tiller
        ports:
        - containerPort: 44134
          name: tiller
        - containerPort: 44135
          name: http
        readinessProbe:
          httpGet:
            path: /readiness
            port: 44135
          initialDelaySeconds: 1
          timeoutSeconds: 1
        resources: {}
      serviceAccountName: tiller
status: {}
---
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    app: helm
    name: tiller
  name: tiller-deploy
  namespace: millet
spec:
  ports:
  - name: tiller
    port: 44134
    targetPort: tiller
  selector:
    app: helm
    name: tiller
  type: ClusterIP
status:
  loadBalancer: {}
```

6. 확인

6.1 helm 실행

- helm 명령실행시 아래와 같이 수행해야 함

```
$ helm ls --tiller-namespace millet

or

$ export TILLER-NAMESPACE=millet
$ helm ls
```

※ tiller-namsespace 는 tiller 가 설치되어 있는 namespace 를 의미한다.

- helm install 및 list 확인

```
$ helm create nginx
$ cd nginx
$ helm install . --name nginx
NAME:   nginx
LAST DEPLOYED: Thu Apr 18 15:06:25 2019
NAMESPACE: millet
STATUS: DEPLOYED

RESOURCES:
==> v1/Service
NAME   TYPE       CLUSTER-IP     EXTERNAL-IP  PORT(S)  AGE
nginx  ClusterIP  172.30.194.70  <none>       80/TCP   0s

==> v1beta2/Deployment
NAME   DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
nginx  1        0        0           0          0s

==> v1/Pod(related)
NAME                    READY  STATUS   RESTARTS  AGE
nginx-6d995b94f-mjbft   0/1    Pending  0         0s


NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace millet -l
"app=nginx,release=nginx" -o jsonpath="{.items[0].metadata.name}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl port-forward $POD_NAME 8080:80




$ helm list --tiller-namespace millet
NAME    REVISION        UPDATED                         STATUS          CHART
    NAMESPACE
nginx   1               Thu Apr 18 14:02:25 2019        DEPLOYED        nginx-0.1.0
    millet
```

6.2 네임스페이스별 업무 격리가 되는지 확인

- 다른 네임스페이스에 설치되어 있는 tiller 를 조회 할때 에러 발생함.

```
$ helm list --tiller-namespace dev-song
Error: pods is forbidden: User "millet-admin" cannot list pods in the namespace
"dev-song": no RBAC policy matched
```

- 다른 네임스페이스에 설치 시도하려면 아래와 같이 에러 발생함

```
$ helm install . --name nginx2 --namespace dev-song
Error: release nginx2 failed: namespaces "dev-song" is forbidden: User
"system:serviceaccount:millet:tiller" cannot get namespaces in the namespace "dev-
song": no RBAC policy matched
```