작성일: 20210330

문서 목적: 인프라 (REDMINE) 인수인계

REDMINE 설치

- 1. Jupyter HUB 접속
- 2. oc login 을 통한 okd 클러스터 접속
- 3. 코드 내려받기

git clone https://gitlab.com/aibigdata/helmcharts.git

4. 레드마인을 배포할 네임스페이스 생성 후 현재 네임스페이스 변경

jovyan@jupyter-minwoo-2elee:~/notebook/helmcharts/customRedmine/redminecustom\$ oc create ns redmine-test namespace/redmine-test created jovyan@jupyter-minwoo-2elee:~/notebook/helmcharts/customRedmine/redminecustom\$ oc project redmine-test Now using project "redmine-test" on server "https://192.168.1.10:6443".

5. scc 제한 해제 (sh 파일에서 Namespace 명이나 serviceAccount 명 확인)

oc adm policy add-scc-to-user \$권한명 system:serviceaccount:\$네임스페이스명:\$서비스어카 운트명

```
oc adm policy add-scc-to-user anyuid system:serviceaccount:redmine-test:redmine-test
oc adm policy add-scc-to-user anyuid system:serviceaccount:redmine-test:default

oc adm policy add-scc-to-user privileged system:serviceaccount:redmine-test:redmine-test
oc adm policy add-scc-to-user privileged system:serviceaccount:redmine:default
```

serviceaccount 명은 value 파일에서 관리

```
serviceAccount:

# Specifies whether a service account should be created
create: true

# The name of the service account to use.

# If not set and create is true, a name is generated using the fullname template
name: redmine
```

ex)scc.sh

```
oc adm policy add-scc-to-user anyuid system:serviceaccount:redmine:redmine
oc adm policy add-scc-to-user anyuid system:serviceaccount:redmine:default
oc adm policy add-scc-to-user privileged system:serviceaccount:redmine:redmine
oc adm policy add-scc-to-user privileged system:serviceaccount:redmine:default
```

6. PV 생성

• PV name 이나 라벨 중첩되지 않게 변경

```
piversion: v1
kind: PersistentVolume
metadata:
   name: redmine-dmsf
   labels:
    app: redmine-dmsf
spec:
   capacity:
    storage: 8Gi
   accessModes:
   - ReadWriteOnce
   persistentVolumeReclaimPolicy: Retain
   nfs:
    server: 192.168.1.7
   path: "/var/nfs-share/redmine/dmsf"
```

```
oc create -f redmine-pv.yaml
oc create -f redmine-board-pv.yaml
```

7. pvc 에서 PV 명 매핑 및 pvc 명도 유니크하게 구성 PV 라벨값과 매핑

```
{{- if and .Values.persistence.enabled }}
kind: PersistentVolumeClaim
apiVersion: v1
 name redmine-dmsf
spec :
   matchLabels:
   app: redmine-dmsf-test
   - {{ .Values.persistence.accessMode | quote }}
  resources
     storage: {{ .Values.persistence.size | quote }}
 storageClassName: {{ .Values.persistence.storageClass }}
{{- end }}
"pvc_redmine.yaml" 17L, 399C written
```

8. Deployment 에 pvc 명 메핑 template/deployment.yaml 하단에 pvc 명 매핑

9. nfs-server 에 디렉토리 생성

```
apiVersion: v1
kind: PersistentVolume
metadata:
   name: redmine-dmsf
   labels:
    app: redmine-dmsf
spec:
   capacity:
    storage: 8Gi
   accessModes:
    - ReadWriteOnce
   persistentVolumeReclaimPolicy: Retain
   nfs:
        server: 192.168.1.7
        path: "/var/nfs-share/redmine/dmsf"
[root@nfs nfs-share]# mkdir /var/nfs-share/redmine/dmsf-test
[root@nfs nfs-share]# mkdir /var/nfs-share/redmine/board_files-test
[root@nfs nfs-share]#
```

접속 IP:192.168.1.7

ID: root

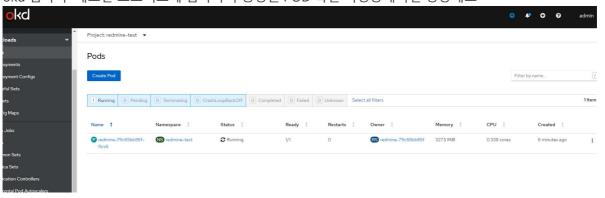
password: !!aibig\$\$2020

9. 레드마인 배포

helm upgrade --install \$helm배포이름 customRedmine/ --values \$value파일경로 -n \$네임스페이스명

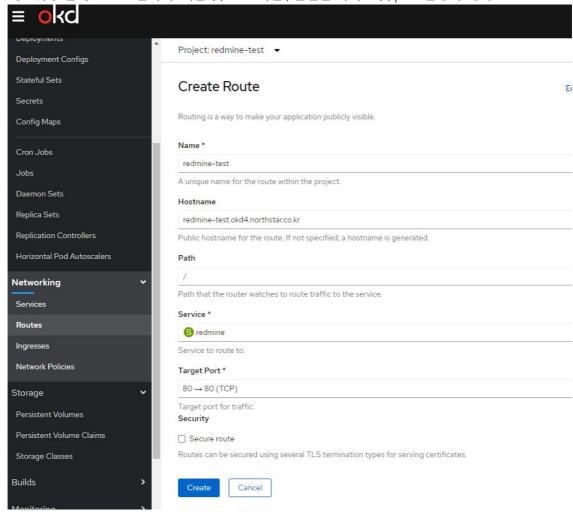
helm upgrade --install redmine-test redminecustom/ --values redminecustom/values.yaml -n redmine-test

okd 접속 후 배포한 프로젝트에 접속하여 생성된 POD 확인 러닝상태이면 정상배포



10. 라우트 생성

네트워킹 탭에 routes 접속 후 이름 및 host 기입, 연결할 서비스 및 port 설정 후 생성



생성한 route 에 접속하여 레드마인 실행

참고자료

쿠버네티스 scc: https://kubernetes.io/ko/docs/concepts/policy/pod-security-policy/

openshift scc 듀토리얼: http://wiki.rockplace.co.kr/display/OP/01.+Service+Accounts+and+SCC