Работа с PV/PVC

В данном руководстве описывается практическая работа с PV/PVC на примере кластера k3d и деплоймента mobsf (до начала работ их нужно развернуть самостоятельно, либо с помощью руководства “Методический материал по работе с deployment”)

Перед началом работ создадим манифест с PV и PVC под названием "`mobsf-pv-pvc.yml`" следующего содержания (подтсавьте делаемый до директории на хосте в поле `hostPath.path`):

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: mobsf-pv
  labels:
    type: local
spec:
  capacity:
    storage: 2Gi
  hostPath:
    path: /home/filipp/Desktop/mobsf/
    type: ''
  accessModes:
    - ReadWriteOnce
  claimRef:
    kind: PersistentVolumeClaim
    namespace: mobsf
    name: mobsf-pvc
    apiVersion: v1
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem

---

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mobsf-pvc
  namespace: mobsf
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
  volumeName: mobsf-pv
  volumeMode: Filesystem
```

Приступим к созданию PV и PVC для неймспейса mobsf, используя заранее подготовленный манифест:

```
$ kubectl apply -f mobsf-pv-pvc.yml -n mobsf
```

```
filipp@filipp-notebook:~/Desktop$ kubectl apply -f mobsf-pv-pvc.yml -n mobsf
persistentvolume/mobsf-pv created
persistentvolumeclaim/mobsf-pvc created
```

Выполним команду просмотра существующих PV в кластере:

```
$ kubectl get pv
```

```
filipp@filipp-notebook:~/Desktop$ kubectl get pv
NAME       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM            STORAGECLASS   REASON   AGE
mobsf-pv   2Gi        RWO            Retain           Bound    mobsf/mobsf-pvc                          65s
```

Выаполним команду просмотра существующих PVC в неймспейсе `mobsf`:

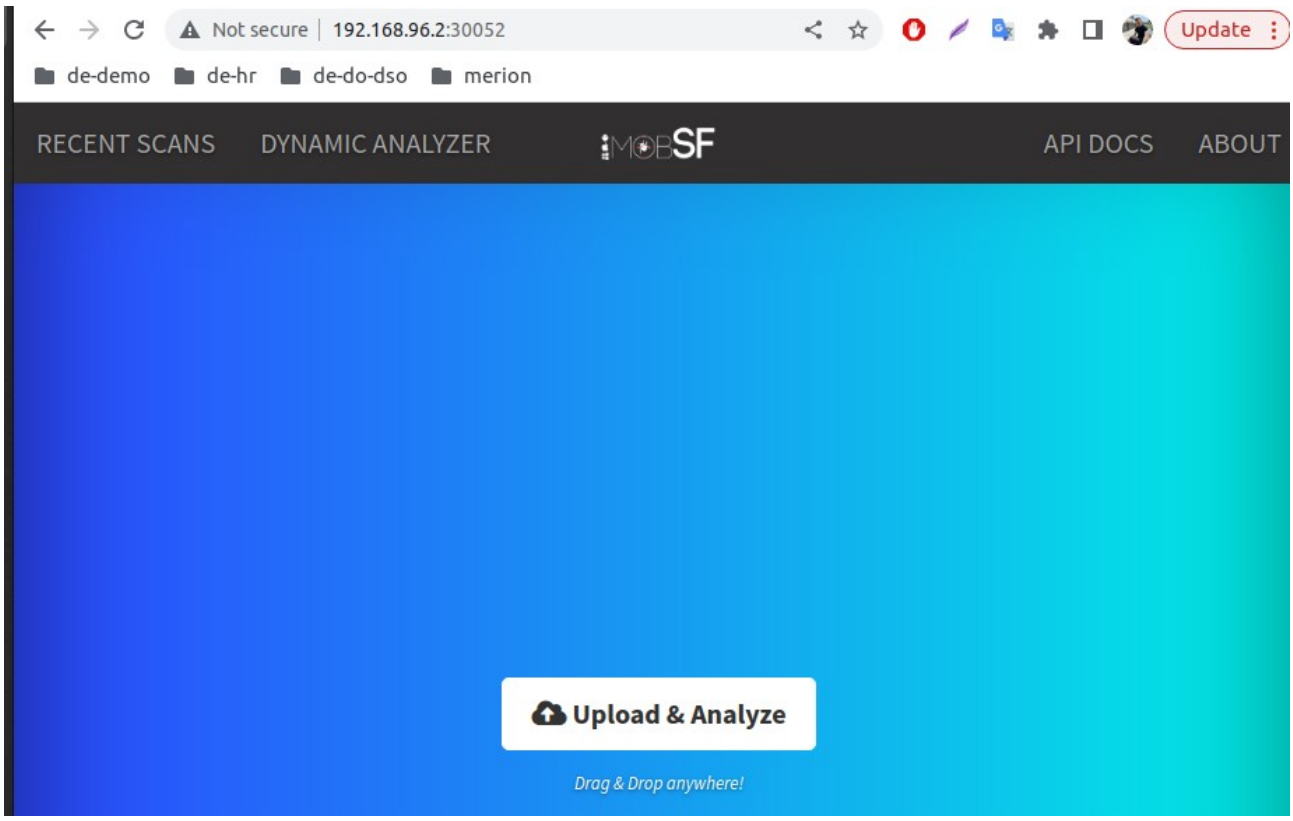
```
$ kubectl get pvc -n mobsf
```

```
filipp@filipp-notebook:~/Desktop$ kubectl get pvc -n mobsf
NAME        STATUS   VOLUME     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mobsf-pvc   Bound    mobsf-pv   2Gi        RWO            local-path     2m7s
```

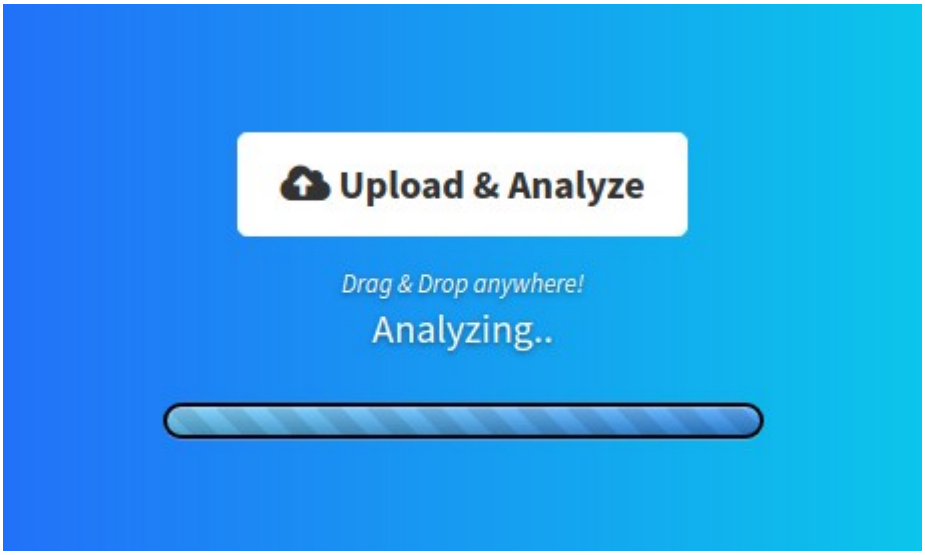
PV и PVC успешно созданы и и сообщены друг с другом

--

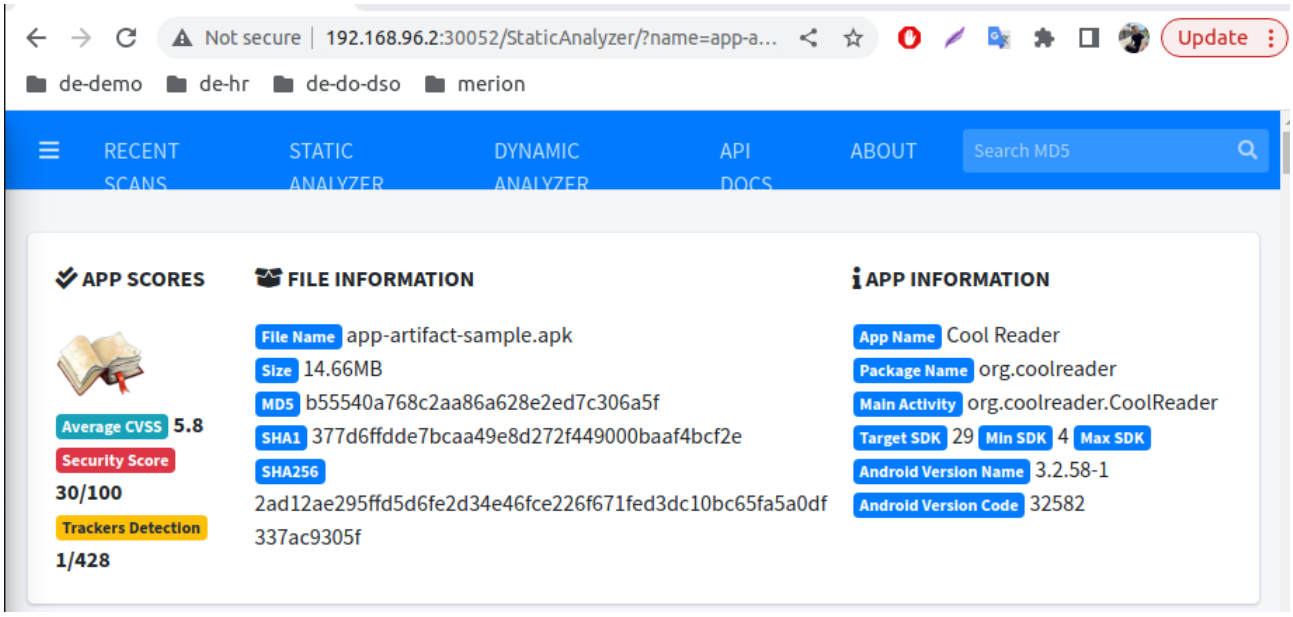
Перейдем в развернутое приложение mobsf:



Запустим сканирование артефакта:



Дождемся завершения анализа:



Выполним дескалирование деплоймента `mobsf` до одной репликации:

```
$ kubectl scale deployment mobsf -n mobsf --replicas=1
```

```
filipp@filipp-notebook:~/Desktop$ kubectl scale deployment mobsf -n mobsf -
-replicas=1
deployment.apps/mobsf scaled
```

Выполним команду получения pod в нейспейсе `mobsf`:

```
$ kubectl get pods -n mobsf
```

```
filipp@filipp-notebook:~/Desktop$ kubectl get pods -n mobsf
NAME                     READY   STATUS    RESTARTS   AGE
mobsf-5db69649fc-vwmwd   1/1     Running   0          116s
```

Используя наименование pod в неймспейсе `mobsf`, выполним команду в режиме псевдотерминала:

```
$ kubectl exec -ti mobsf-764d9fc5bb-qwxtv -n mobsf sh
```

```
filipp@filipp-notebook:~/Desktop$ kubectl exec -ti mobsf-5db69649fc-vwmwd -n mobsf sh
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use
 kubectl exec [POD] -- [COMMAND] instead.
#
```

В режиме псевдотерминала выполним следующие команды для нахождения просканированного mobsf артефакта:

```
$ cd /root/.MobSF
```

```
$ ls uploads
```

```
filipp@filipp-notebook:~/Desktop$ kubectl exec -ti mobsf-5db69649fc-vwmwd -n mobsf sh
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use
 kubectl exec [POD] -- [COMMAND] instead.
# cd /root/.MobSF
# ls uploads
b55540a768c2aa86a628e2ed7c306a5f
#
```

Обнаруживаем артефакт в директории `/root/.MobSF/uploads`. Закрываем сессию в псевдотерминала pod соечатнием клавиш CTRL+D

Произведем дескалирование деплоймента до 0 репликаций и скалирование в 1 репликацию:

```
$ kubectl scale deployment mobsf -n mobsf --replicas=0
$ kubectl scale deployment mobsf -n mobsf --replicas=1
```

```
filipp@filipp-notebook:~/Desktop$ kubectl scale deployment mobsf -n mobsf --replicas=0
deployment.apps/mobsf scaled
filipp@filipp-notebook:~/Desktop$ kubectl scale deployment mobsf -n mobsf --replicas=1
deployment.apps/mobsf scaled
```

Повторим выполнение команд по обнаружению наименования pod в неймспейсе `mobsf`, используем его для перехода в псевдотерминал и повторим команды для обнаружения артефакта:

```
filipp@filipp-notebook:~/Desktop$ kubectl exec -ti mobsf-5db69649fc-6wrf6 -n mobsf sh
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use
 kubectl exec [POD] -- [COMMAND] instead.
# cd /root/.MobSF
# ls uploads
#
```

Артефакт не найден, поскольку отсутствует сообщенный с деплойментом PV

--

Модифицируем деплоймент `mobsf` для работы с созданными нами ранее PV и PVC:

```
$ kubectl edit deployment mobsf -n mobsf
```

```
  spec:
      containers:
      - image: opensecurity/mobile-security-framework-mobsf:v3.1.1
        imagePullPolicy: IfNotPresent
        name: mobile-security-framework-mobsf
        resources: {}
        terminationMessagePath: /dev/termination-log
```

```
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /root/.MobSF
      name: mobsf-data
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  terminationGracePeriodSeconds: 30
  volumes:
  - name: mobsf-data
    persistentVolumeClaim:
      claimName: mobsf-pvc
```

```
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: mobsf
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: mobsf
    spec:
      containers:
      - image: opensecurity/mobile-security-framework-mobsf:v3.1.1
        imagePullPolicy: IfNotPresent
        name: mobile-security-framework-mobsf
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
        volumeMounts:
        - mountPath: /root/.MobSF
          name: mobsf-data
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
      volumes:
      - name: mobsf-data
        persistentVolumeClaim:
          claimName: mobsf-pvc
```

Сохраняем изменения в полях `volumeMounts` и `volumes`:

```
filipp@filipp-notebook:~/Desktop$ kubectl edit deployment mobsf -n mobsf
deployment.apps/mobsf edited
```

--

Повторим сканирование мобильного артефакта в mobsf и выполним команды в псевдотерминале текущего pod в неймспейсе `mobsf` для обнаружения просканировнного артефакта:

```
filipp@filipp-notebook:~/Desktop$ kubectl exec -ti mobsf-764d9fc5bb-58jtv -n mobsf sh
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Us
 kubectl exec [POD] -- [COMMAND] instead.
# cd /root/.MobSF
# ls uploads
b55540a768c2aa86a628e2ed7c306a5f
#
```

Выполним дескалирование деплоймента в 0 репликаций и скалирование в 1 репликацию:

```
$ kubectl scale deployment mobsf -n mobsf --replicas=0
$ kubectl scale deployment mobsf -n mobsf --replicas=1
```

```
#
filipp@filipp-notebook:~/Desktop$ kubectl scale deployment mobsf -n mobsf --replicas=0
deployment.apps/mobsf scaled
filipp@filipp-notebook:~/Desktop$ kubectl scale deployment mobsf -n mobsf --replicas=1
deployment.apps/mobsf scaled
```

Выполняем команды в псевдотерминалом текущего pod в неймспейсе `mobsf` для обнаружения ранее просканированного артефакта (после дескалирования и обратного скалирования деплоймента):

```
filipp@filipp-notebook:~/Desktop$ kubectl exec -ti mobsf-764d9fc5bb-58jtv -n mobsf sh
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Us
 kubectl exec [POD] -- [COMMAND] instead.
# cd /root/.MobSF
# ls uploads
b55540a768c2aa86a628e2ed7c306a5f
#
```

Артефакт сохранился после перезапуска pod благодаря использованию сообщенных PV и PVC в неймспейсе

--

Для завершения работы с кластером k3d выполните команду:

```
$ k3d cluster delete mycluster
```

```
filipp@filipp-notebook:~/Desktop$ k3d cluster delete mycluster
INFO[0000] Deleting cluster 'mycluster'
INFO[0004] Deleting cluster network 'k3d-mycluster'
INFO[0004] Deleting image volume 'k3d-mycluster-images'
INFO[0004] Removing cluster details from default kubeconfig...
INFO[0004] Removing standalone kubeconfig file (if there is one)...
INFO[0004] Successfully deleted cluster mycluster!
```

--