

etcd 데이터 백업 및 복원방법

시작하는말

안녕하세요, 고니입니다.

이번에는 kubernetes에서 가장 중요한 역할을 하는 Component가 API서버와 kubernetes의 정보를 저장하고 있는 ETCD 백업/복구를 하기 위한 방법을 기술하려고 합니다.

⚠ etcd 3.6이후부터는 etcdctl 명령어는 deprecated 된다고 합니다. (etcdutil -> etcdctl)ㄴ

소 개

etcd는 정족수에 따라 $(n-1)/2$ 의 노드수 만큼 일시적인 장애를 허용하지만, H/W의 장애나 네트워크의 영구적인 문제 등 단기간내에 복구가 이루어 지지 않으면 ETCD 자체에 문제가 발생할 수도 있습니다, 이럴때 주기적으로 백업(snapshot)을 받아놓았다면, 해당 백업(snapshot)을 가지고 복구를 수행 후 kubernetes를 운영할 수 있습니다.

etcd 백업(snapshot)은 다중 노드에서 수행해도 되나, 복구는 동일한 백업(snapshot)을 가지고 복구를 해야 합니다, 특히 다중 노드에서 개별 노드에 백업(snapshot)된 데이터를 복구할 경우 오히려 kubernetes의 데이터의 문제로 장애가 발생 할 수 있으니, 복구는 동일한 snapshot을 가지고 복구를 수행하셔야 합니다.

백업(snapshot)수행시 무결성 검증을 위한 해쉬값이 포함되는데, 백업(snapshot)된 데이터의 수정을 하는 경우 해쉬데이터가 깨질수 있습니다. 불가피한 경우로 해쉬데이터를 무시하고 복구를 수행이 필요한 경우 `--skip-hash-check` 옵션을 추가하여 복구를 수행하면 복구가 가능합니다.

작업절차

1. etcd스크립트 생성

```
$> cat /usr/local/bin/etcdctl.sh
#!/bin/bash
# Ansible managed
# example invocation: etcdctl.sh get --keys-only --from-key ""

etcdctl \
  --cacert /etc/ssl/etcd/ssl/ca.pem \
  --cert /etc/ssl/etcd/ssl/admin-master1.pem \
  --key /etc/ssl/etcd/ssl/admin-master1-key.pem "$@"
```

2. /tmp/backup 파일로 etcd snapshot 수행

```
$> ./etcdctl.sh snapshot save /tmp/backup
{"level":"info","ts":"2024-09-23T23:43:44.67136+0900","caller":"snapshot/v3_snapshot.go:65","msg":"created temporary db file","path":"/tmp/backup.part"}
{"level":"info","ts":"2024-09-23T23:43:44.672148+0900","logger":"client","caller":"v3@v3.5.10/maintenance.go:212","msg":"opened snapshot stream; downloading"}
{"level":"info","ts":"2024-09-23T23:43:44.672741+0900","caller":"snapshot/v3_snapshot.go:73","msg":"fetching snapshot","endpoint":"127.0.0.1:2379"}
{"level":"info","ts":"2024-09-23T23:43:44.71756+0900","logger":"client","caller":"v3@v3.5.10/maintenance.go:220","msg":"completed snapshot read; closing"}
{"level":"info","ts":"2024-09-23T23:43:44.730295+0900","caller":"snapshot/v3_snapshot.go:88","msg":"fetched snapshot","endpoint":"127.0.0.1:2379","size":"6.2 MB","took":"now"}
{"level":"info","ts":"2024-09-23T23:43:44.730359+0900","caller":"snapshot/v3_snapshot.go:97","msg":"saved","path":"/tmp/backup"}
```

3. 백업(snapshot)된 데이터 확인

```
$> ./etcdctl.sh snapshot status /tmp/backup -w table
+-----+-----+-----+-----+
| HASH | REVISION | TOTAL KEYS | TOTAL SIZE |
+-----+-----+-----+-----+
| 844eca85 | 1914345 | 1100 | 6.2 MB |
+-----+-----+-----+-----+
```

4. 백업된 데이터 복구 방법

```
$> ./etcdctl.sh snapshot restore /tmp/backup
2024-09-23T23:48:49+09:00 info snapshot/v3_snapshot.go:260 restoring snapshot {"path": "/tmp/backup", "wal-dir":
"default.etcd/member/wal", "data-dir": "default.etcd", "snap-dir": "default.etcd/member/snap"}
2024-09-23T23:48:49+09:00 info membership/store.go:141 Trimming membership information from the backend...
2024-09-23T23:48:49+09:00 info membership/cluster.go:421 added member {"cluster-id": "cdf818194e3a8c32", "local-member-id": "0",
"added-peer-id": "8e9e05c52164694d", "added-peer-peer-urls": ["http://localhost:2380"]}
2024-09-23T23:48:49+09:00 info snapshot/v3_snapshot.go:287 restored snapshot {"path": "/tmp/backup", "wal-dir":
"default.etcd/member/wal", "data-dir": "default.etcd", "snap-dir": "default.etcd/member/snap"}
```

5. timer기반의 정기 백업 절차 구성

- 백업스크립트 구성

```
$> vi /root/etcd_backup.sh

#!/bin/bash

PATH=/usr/local/bin:$PATH

BACK_DATE=$(date +%Y-%m-%d)
ORI_DATE=$(date +%Y-%m-%d -d '7days')
BACK_DIR=/tmp/backup

#백업 디렉토리 없으면 생성
if [[ ! -d $BACK_DIR ]]
then
    mkdir -p $BACK_DIR
fi

#백업 수행
/usr/local/bin/etcdctl.sh snapshot save $BACK_DIR/etcd-$BACK_DATE

#백업파일 확인 후 미생성시 에러
if [[ ! -f $BACK_DIR/etcd-$BACK_DATE ]]
then
    echo "Backup failed"
    exit 1
fi

#오래된 데이터 삭제
rm -f $BACK_DIR/etcd-$ORI_DATE
```

```
$> chmod +x /root/etcd_backup.sh
$> ls -l /root/etcd_backup.sh
-rwxr-xr-x 1 root root 483 9월 23 23:55 /root/etcd_backup.sh
```

- timer 구성 (매일 04시에 백업수행)

```
$> cat /etc/systemd/system/etcd_backup.timer
[Unit]
Description=ETCD Backup

[Timer]
OnCalendar=*-*-* 04:00:00

[Install]
WantedBy=multi-user.target
```

- timer에 연결된 서비스 구성

```
$> cat /etc/systemd/system/etcd_backup.service
[Unit]
Description=ETCD Backup script

[Service]
Type=oneshot
ExecStart= /root/etcd_backup.sh
```

- backup timer 활성화

```
$> systemctl daemon-reload
$> systemctl enable etcd_backup.timer --now
Created symlink /etc/systemd/system/multi-user.target.wants/etcd_backup.timer → /etc/systemd/system/etcd_backup.timer.
```

- timer 확인

```
$> systemctl status etcd_backup.timer
● etcd_backup.timer - ETCD Backup
   Loaded: loaded (/etc/systemd/system/etcd_backup.timer; enabled; vendor preset: disabled)
   Active: active (waiting) since Mon 2024-06-23 23:58:42 KST; 1s ago
     Until: Mon 2024-06-23 23:58:42 KST; 1s ago
    Trigger: Tue 2024-06-24 04:00:00 KST; 6h left
    Triggers: ● etcd_backup.service
```

Reference

- <https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/#etcd-verify-snapshot-1>
- <https://etcd.io/docs/v3.5/op-guide/maintenance/>

🔄Revision #4

★Created 23 September 2024 23:34:17 by artop0420

✎Updated 24 September 2024 00:46:56 by artop0420