

etcd 安装 部署 使用 配置

简介

etcd 是一个分布式键值数据库。

etcd 是 Kubernetes 的组件之一，它提供了 Kubernetes 的 API 接口。etcd 是 Kubernetes 的 ETCD 组件，它提供了 Kubernetes 的 ETCD 接口。

etcd 3.6 版本 etcdctl 已经 deprecated 了，建议使用 (etcdutil -> etcdctl) [1](#)

安装

etcd 是一个分布式键值数据库，它提供了 Kubernetes 的 API 接口。etcd 是 Kubernetes 的 ETCD 组件，它提供了 Kubernetes 的 ETCD 接口。etcd 是一个分布式键值数据库，它提供了 Kubernetes 的 API 接口。etcd 是 Kubernetes 的 ETCD 组件，它提供了 Kubernetes 的 ETCD 接口。

etcd 是一个分布式键值数据库，它提供了 Kubernetes 的 API 接口。etcd 是 Kubernetes 的 ETCD 组件，它提供了 Kubernetes 的 ETCD 接口。etcd 是一个分布式键值数据库，它提供了 Kubernetes 的 API 接口。etcd 是 Kubernetes 的 ETCD 组件，它提供了 Kubernetes 的 ETCD 接口。

etcd 是一个分布式键值数据库，它提供了 Kubernetes 的 API 接口。etcd 是 Kubernetes 的 ETCD 组件，它提供了 Kubernetes 的 ETCD 接口。etcd 是一个分布式键值数据库，它提供了 Kubernetes 的 API 接口。etcd 是 Kubernetes 的 ETCD 组件，它提供了 Kubernetes 的 ETCD 接口。

部署

1. etcd

```
$> 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 etcd snapshot

```
$> ./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. ❷ (snapshot) ❸ ❹

```
$> ./etcdctl.sh snapshot status /tmp/backup -w table
```

```
+-----+-----+-----+-----+
| HASH | REVISION | TOTAL KEYS | TOTAL SIZE |
+-----+-----+-----+-----+
| 844eca85 | 1914345 | 1100 | 6.2 MB |
+-----+-----+-----+-----+
```

4. ❶ ❷ ❸ ❹

```
$> ./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. timer❶❷ ❸ ❹ ❺

- ❶❷❸❹ ❺

```
$> 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
```

```
# 脚本 名称 文件 类型
```

```
if [[ ! -d $BACK_DIR ]]
```

```
then
```

```
    mkdir -p $BACK_DIR
```

```
fi
```

```
# 脚本 名称
```

```
/usr/local/bin/etcdctl.sh snapshot save $BACK_DIR/etcd-$BACK_DATE
```

```
# 脚本 名称 文件 类型 文件 类型
```

```
if [[ ! -f $BACK_DIR/etcd-$BACK_DATE ]]
```

```
    echo "Backup failed"
```

```
    exit 1
```

```
fi
```

```
# 脚本 名称 文件 类型
```

```
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 9月 23 23:55 /root/etcd_backup.sh
```

- timer 创建 (每天 04:00 备份)

```
$> cat /etc/systemd/system/etcd_backup.timer

[Unit]
Description=ETCD Backup

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

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

- timer 的 service 文件

```
$> cat /etc/systemd/system/etcd_backup.service

[Unit]
Description=ETCD Backup script

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

- backup timer 的启动

```
$> 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 ☐

```
$> 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