Docker (/tags/#Docker) Etcd (/tags/#Etcd)

Etcd 使用入门

Posted by Mike on 2017-04-10

etcd简介

etcd是CoreOS团队于2013年6月发起的开源项目，它的目标是构建一个高可用的分布式键值(key-value)数据库。etcd内部采用 `raft` 协议作为一致性算法，etcd基于Go语言实现。

etcd作为服务发现系统，有以下的特点：

- 简单：安装配置简单，而且提供了HTTP API进行交互，使用也很简单
- 安全：支持SSL证书验证
- 快速：根据官方提供的benchmark数据，单实例支持每秒2k+读操作
- 可靠：采用raft算法，实现分布式系统数据的可用性和一致性

etcd项目地址：https://github.com/coreos/etcd/ (https://github.com/coreos/etcd/)

etcd应用场景

etcd比较多的应用场景是用于服务发现，服务发现(Service Discovery)要解决的是分布式系统中最常见的问题之一，即在同一个分布式集群中的进程或服务如何才能找到对方并建立连接。

从本质上说，服务发现就是要了解集群中是否有进程在监听upd或者tcp端口，并且通过名字就可以进行查找和链接。

要解决服务发现的问题，需要下面三大支柱，缺一不可。

- 一个强一致性、高可用的服务存储目录。

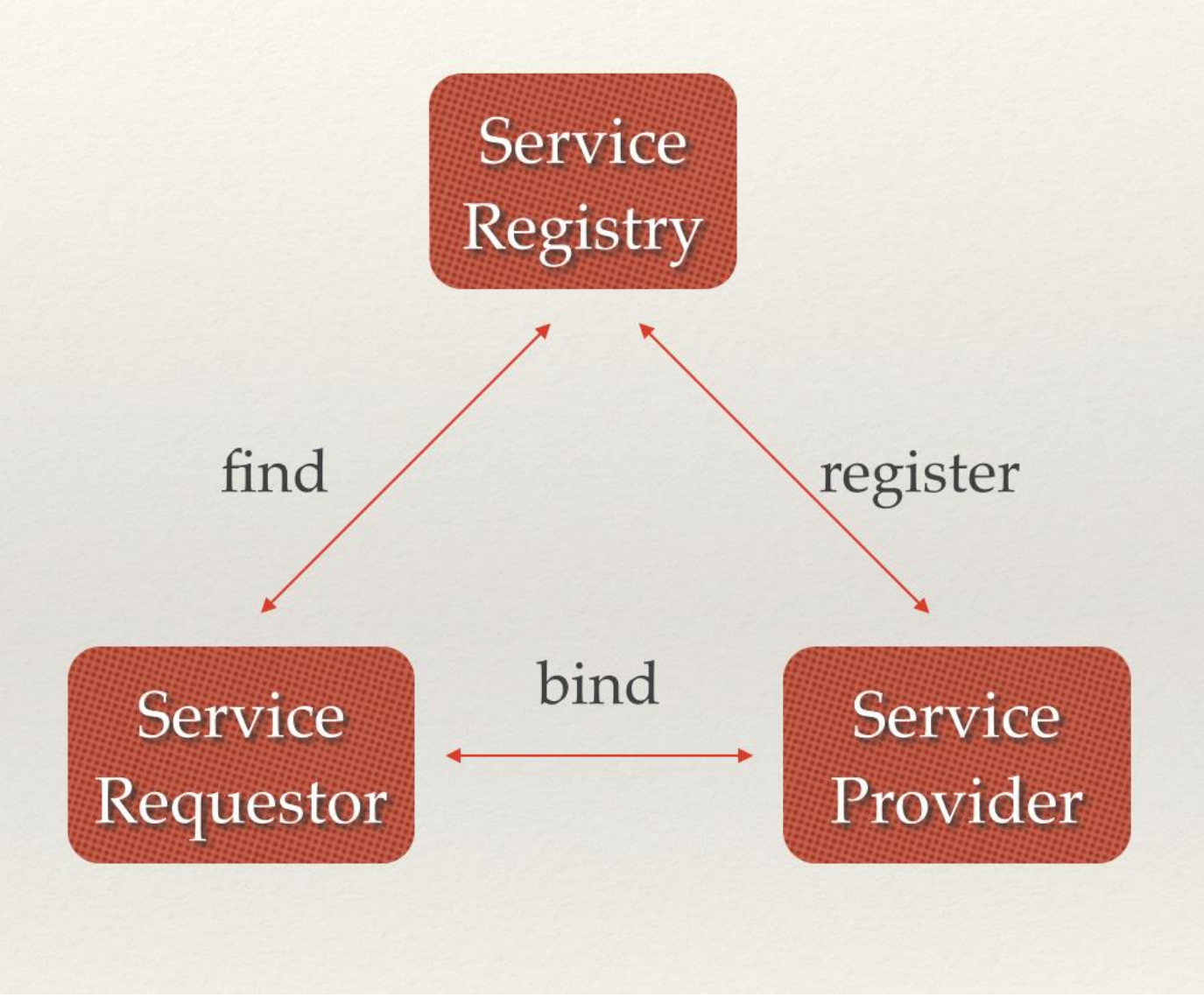
基于Ralf算法的etcd天生就是这样一个强一致性、高可用的服务存储目录。

- 一种注册服务和健康服务健康状况的机制。

用户可以在etcd中注册服务，并且对注册的服务配置key TTL，定时保持服务的心跳以达到监控健康状态的效果。

- 一种查找和连接服务的机制。

通过在etcd指定的主题下注册的服务业能在对应的主题下查找到。为了确保连接，我们可以在每个服务机器上都部署一个proxy模式的etcd，这样就可以确保访问etcd集群的服务都能够互相连接。



etcd安装

etcd在生产环境中一般推荐集群方式部署。本文定位为入门，主要讲讲单节点安装和基本使用。

etcd目前默认使用2379端口提供HTTP API服务，2380端口和peer通信(这两个端口已经被IANA官方预留给etcd)；在之前的版本中可能会分别使用4001和7001，在使用的过程中需要注意这个区别。

因为etcd是go语言编写的，安装只需要下载对应的二进制文件，并放到合适的路径就行。

下载软件包

```
$ wget https://github.com/coreos/etcd/releases/download/v3.1.5/etcd-v3.1.5-linux-amd64.t
$ tar xzvf etcd-v3.1.5-linux-amd64.tar.gz
$ mv etcd-v3.1.5-linux-amd64 /opt/etcd
```

解压后是一些文档和两个二进制文件etcd和etcdctl。etcd是server端，etcdctl是客户端。

```
$ ls

Documentation  etcd  etcdctl  README-etcdctl.md  README.md  READMEv2-etcdctl.md
```

如果在测试环境，启动一个单节点的etcd服务，只需要运行 `etcd` 命令就行。

```
$ ./etcd
2017-04-10 11:46:44.772465 I | etcdmain: etcd Version: 3.1.5
2017-04-10 11:46:44.772512 I | etcdmain: Git SHA: 20490ca
2017-04-10 11:46:44.772607 I | etcdmain: Go Version: go1.7.5
2017-04-10 11:46:44.772756 I | etcdmain: Go OS/Arch: linux/amd64
2017-04-10 11:46:44.772817 I | etcdmain: setting maximum number of CPUs to 2, total num
2017-04-10 11:46:44.772851 W | etcdmain: no data-dir provided, using default data-dir .
2017-04-10 11:46:44.773298 I | embed: listening for peers on http://localhost:2380
2017-04-10 11:46:44.773583 I | embed: listening for client requests on localhost:2379
2017-04-10 11:46:44.775967 I | etcdserver: name = default
2017-04-10 11:46:44.775993 I | etcdserver: data dir = default.etcd
2017-04-10 11:46:44.776167 I | etcdserver: member dir = default.etcd/member
2017-04-10 11:46:44.776253 I | etcdserver: heartbeat = 100ms
2017-04-10 11:46:44.776264 I | etcdserver: election = 1000ms
2017-04-10 11:46:44.776270 I | etcdserver: snapshot count = 10000
2017-04-10 11:46:44.776285 I | etcdserver: advertise client URLs = http://localhost:237
2017-04-10 11:46:44.776293 I | etcdserver: initial advertise peer URLs = http://localho
2017-04-10 11:46:44.776306 I | etcdserver: initial cluster = default=http://localhost:2
2017-04-10 11:46:44.781171 I | etcdserver: starting member 8e9e05c52164694d in cluster
2017-04-10 11:46:44.781323 I | raft: 8e9e05c52164694d became follower at term 0
2017-04-10 11:46:44.781351 I | raft: newRaft 8e9e05c52164694d [peers: [], term: 0, comm
2017-04-10 11:46:44.781883 I | raft: 8e9e05c52164694d became follower at term 1
2017-04-10 11:46:44.795542 I | etcdserver: starting server... [version: 3.1.5, cluster
2017-04-10 11:46:44.796453 I | etcdserver/membership: added member 8e9e05c52164694d [ht
2017-04-10 11:46:45.083350 I | raft: 8e9e05c52164694d is starting a new election at ter
2017-04-10 11:46:45.083494 I | raft: 8e9e05c52164694d became candidate at term 2
2017-04-10 11:46:45.083520 I | raft: 8e9e05c52164694d received MsgVoteResp from 8e9e05c
2017-04-10 11:46:45.083598 I | raft: 8e9e05c52164694d became leader at term 2
2017-04-10 11:46:45.083654 I | raft: raft.node: 8e9e05c52164694d elected leader 8e9e05c
2017-04-10 11:46:45.084544 I | etcdserver: published {Name:default ClientURLs:[http://l
2017-04-10 11:46:45.084638 I | etcdserver: setting up the initial cluster version to 3.
2017-04-10 11:46:45.084857 I | embed: ready to serve client requests
2017-04-10 11:46:45.085918 E | etcdmain: forgot to set Type=notify in systemd service f
2017-04-10 11:46:45.086668 N | embed: serving insecure client requests on 127.0.0.1:237
2017-04-10 11:46:45.087004 N | etcdserver/membership: set the initial cluster version t
2017-04-10 11:46:45.087195 I | etcdserver/api: enabled capabilities for version 3.1
```

从上面的输出中，我们可以看到很多信息。以下是几个比较重要的信息：

- name表示节点名称，默认为default。
- data-dir 保存日志和快照的目录，默认为当前工作目录default.etcd/目录下。
- 在http://localhost:2380和集群中其他节点通信。
- 在http://localhost:2379提供HTTP API服务，供客户端交互。
- heartbeat为100ms，该参数的作用是leader多久发送一次心跳到followers，默认值是100ms。
- election为1000ms，该参数的作用是重新投票的超时时间，如果follow在该时间间隔没有收到心跳包，会触发重新投票，默认为1000ms。
- snapshot count为10000，该参数的作用是指定有多少事务被提交时，触发截取快照保存到磁盘。

- 集群和每个节点都会生成一个uuid。
- 启动的时候会运行raft，选举出leader。

上面的方法只是简单的启动一个etcd服务，但要长期运行的话，还是做成一个服务好一些。下面将以systemd为例，介绍如何建立一个etcd服务。

创建systemd服务

- 设定etcd配置文件

建立相关目录

```
$ mkdir -p /var/lib/etcd/
$ mkdir -p /opt/etcd/config/
```

- 创建etcd配置文件

```
$ cat <<EOF | sudo tee /opt/etcd/config/etcd.conf
#节点名称
ETCD_NAME=$(hostname -s)
#数据存放位置
ETCD_DATA_DIR=/var/lib/etcd
EOF
```

- 创建systemd配置文件

```
$ cat <<EOF | sudo tee /etc/systemd/system/etcd.service

[Unit]
Description=Etcd Server
Documentation=https://github.com/coreos/etcd
After=network.target

[Service]
User=root
Type=notify
EnvironmentFile=-/opt/etcd/config/etcd.conf
ExecStart=/opt/etcd/etcd
Restart=on-failure
RestartSec=10s
LimitNOFILE=40000

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

- 启动etcd


```
$ systemctl daemon-reload && systemctl enable etcd && systemctl start etcd
```

etcd基本使用

etcdctl是一个命令行客户端，它能提供一些简洁的命令，供用户直接跟etcd服务打交道，而无需基于 HTTP API方式。可以方便我们在对服务进行测试或者手动修改数据库内容。建议刚刚接触etcd时通过etdctl来熟悉相关操作。这些操作跟HTTP API基本上是对应的。

etcd项目二进制发行包中已经包含了etcdctl工具，etcdctl支持的命令大体上分为数据库操作和非数据库操作两类。

```
$ etcd --version
etcd Version: 3.1.5
Git SHA: 20490ca
Go Version: go1.7.5
Go OS/Arch: linux/amd64

$ etcdctl -h
NAME:
   etcdctl - A simple command line client for etcd.

USAGE:
   etcdctl [global options] command [command options] [arguments...]

VERSION:
   3.1.5

COMMANDS:
     backup          backup an etcd directory
     cluster-health  check the health of the etcd cluster
     mk              make a new key with a given value
     mkdir           make a new directory
     rm              remove a key or a directory
     rmdir           removes the key if it is an empty directory or a key-value pair
     get             retrieve the value of a key
     ls              retrieve a directory
     set             set the value of a key
     setdir          create a new directory or update an existing directory TTL
     update          update an existing key with a given value
     updatedir       update an existing directory
     watch           watch a key for changes
     exec-watch      watch a key for changes and exec an executable
     member          member add, remove and list subcommands
     user            user add, grant and revoke subcommands
     role            role add, grant and revoke subcommands
     auth            overall auth controls
     help, h         Shows a list of commands or help for one command

GLOBAL OPTIONS:
   --debug                          output cURL commands which can be used to reproduce
   --no-sync                        don't synchronize cluster information before sendin
   --output simple, -o simple       output response in the given format (simple, `exten
   --discovery-srv value, -D value  domain name to query for SRV records describing clu
   --insecure-discovery             accept insecure SRV records describing cluster endp
   --peers value, -C value          DEPRECATED - "--endpoints" should be used instead
   --endpoint value                 DEPRECATED - "--endpoints" should be used instead
   --endpoints value                a comma-delimited list of machine addresses in the
   --cert-file value                identify HTTPS client using this SSL certificate fi
   --key-file value                 identify HTTPS client using this SSL key file
   --ca-file value                  verify certificates of HTTPS-enabled servers using
   --username value, -u value       provide username[:password] and prompt if password
   --timeout value                  connection timeout per request (default: 2s)
   --total-timeout value            timeout for the command execution (except watch) (d
   --help, -h                       show help
   --version, -v                    print the version
```

常用命令选项:

```
--debug 输出CURL命令，显示执行命令的时候发起的请求
--no-sync 发出请求之前不同步集群信息
--output, -o 'simple' 输出内容的格式(simple 为原始信息，json 为进行json格式解码，易读性好一些
--peers, -C 指定集群中的同伴信息，用逗号隔开(默认为: "127.0.0.1:4001")
--cert-file HTTPS下客户端使用的SSL证书文件
--key-file HTTPS下客户端使用的SSL密钥文件
--ca-file 服务端使用HTTPS时，使用CA文件进行验证
--help, -h 显示帮助命令信息
--version, -v 打印版本信息
```

数据库操作

数据库操作围绕对键值和目录的CRUD完整生命周期的管理。

etcd在键的组织上采用了层次化的空间结构(类似于文件系统中目录的概念)，用户指定的键可以为单独的名字，如:testkey，此时实际上放在根目录 / 下面，也可以为指定目录结构，
如 `/cluster1/node2/testkey` ，则将创建相应的目录结构。

> *注：CRUD即Create,Read,Update,Delete是符合REST风格的一套API操作。*

- set

指定某个键的值。例如:

```
$ etcdctl set /testdir/testkey "Hello world"
Hello world
```

支持的选项包括:

```
--ttl '0' 该键值的超时时间(单位为秒)，不配置(默认为0)则永不超时
--swap-with-value value 若该键现在的值是value，则进行设置操作
--swap-with-index '0'   若该键现在的索引值是指定索引，则进行设置操作
```

- get

获取指定键的值。例如：

```
$ etcdctl get /testdir/testkey
Hello world
```

当键不存在时，则会报错。例如：

```
$ etcdctl get /testdir/testkey2
Error:  100: Key not found (/testdir/testkey2) [5]
```

支持的选项为:

```
--sort 对结果进行排序
--consistent 将请求发给主节点，保证获取内容的一致性。
```

- update

当键存在时，更新值内容。例如：

```
$ etcdctl update /testdir/testkey "Hello"
Hello
```

当键不存在时，则会报错。例如:

```
$ etcdctl update /testdir/testkey2 "Hello"
Error:  100: Key not found (/testdir/testkey2) [6]
```

支持的选项为:

```
--ttl '0' 超时时间(单位为秒)，不配置(默认为 0)则永不超时。
```

- rm

删除某个键值。例如:

```
1   $ etcdctl rm /testdir/testkey
2   PrevNode.Value: Hello
```

当键不存在时，则会报错。例如:

```
1   $ etcdctl rm /testdir/testkey
2   Error:  100: Key not found (/testdir/testkey) [7]
```

支持的选项为:

```
1   --dir 如果键是个空目录或者键值对则删除
2   --recursive 删除目录和所有子键
3   --with-value  检查现有的值是否匹配
4   --with-index '0'检查现有的index是否匹配
```

- mk

如果给定的键不存在，则创建一个新的键值。例如:

```
1   $ etcdctl mk /testdir/testkey "Hello world"
2   Hello world
```

当键存在的时候，执行该命令会报错，例如:

```
1   $ etcdctl mk /testdir/testkey "Hello world"
2   Error:  105: Key already exists (/testdir/testkey) [8]
```

支持的选项为:

```
1   --ttl '0'  超时时间(单位为秒），不配置(默认为 0)。则永不超时
```

- mkdir

如果给定的键目录不存在，则创建一个新的键目录。例如：

```
$ etcdctl mkdir testdir2
```

当键目录存在的时候，执行该命令会报错，例如：

```
$ etcdctl mkdir testdir2
Error:  105: Key already exists (/testdir2) [9]
```

支持的选项为：

```
--ttl '0' 超时时间(单位为秒)，不配置(默认为0)则永不超时。
```

- setdir

创建一个键目录。如果目录不存在就创建，如果目录存在更新目录TTL。

```
$ etcdctl setdir testdir3
```

支持的选项为:

```
--ttl '0' 超时时间(单位为秒)，不配置(默认为0)则永不超时。
```

- updatedir

更新一个已经存在的目录。

```
$ etcdctl updatedir testdir2
```

支持的选项为:

```
--ttl '0' 超时时间(单位为秒)，不配置(默认为0)则永不超时。
```

- rmdir

删除一个空目录，或者键值对。

```
1   $ etcdctl setdir dir1
2   $ etcdctl rmdir dir1
```

若目录不空，会报错:

```
1   $ etcdctl set /dir/testkey hi
2   hi
3   $ etcdctl rmdir /dir
4   Error:  108: Directory not empty (/dir) [17]
```

- ls

列出目录(默认为根目录)下的键或者子目录，默认不显示子目录中内容。

例如:

```
1   $ etcdctl ls
2   /testdir
3   /testdir2
4   /dir
5
6   $ etcdctl ls dir
7   /dir/testkey
```

支持的选项包括:

```
1   --sort 将输出结果排序
2   --recursive 如果目录下有子目录，则递归输出其中的内容
3   -p 对于输出为目录，在最后添加/进行区分
```

非数据库操作

- backup

备份etcd的数据。

```
$ etcdctl backup --data-dir /var/lib/etcd  --backup-dir /home/etcd_backup
```

支持的选项包括:

```
--data-dir  etcd的数据目录
--backup-dir 备份到指定路径
```

- watch

监测一个键值的变化，一旦键值发生更新，就会输出最新的值并退出。

例如:用户更新testkey键值为Hello watch。

```
$ etcdctl get /testdir/testkey
Hello world
$ etcdctl set /testdir/testkey "Hello watch"
Hello watch
$ etcdctl watch testdir/testkey
Hello watch
```

支持的选项包括:

```
--forever  一直监测直到用户按CTRL+C退出
--after-index '0' 在指定index之前一直监测
--recursive 返回所有的键值和子键值
```

- exec-watch

监测一个键值的变化，一旦键值发生更新，就执行给定命令。

例如：用户更新testkey键值。

```
$ etcdctl exec-watch testdir/testkey -- sh -c 'ls'
config  Documentation  etcd  etcdctl  README-etcdctl.md  README.md  READMEv2-etcdctl.md
```

支持的选项包括:

```
--after-index '0' 在指定 index 之前一直监测
--recursive 返回所有的键值和子键值
```

- member

通过 `list` 、 `add` 、 `remove` 命令列出、添加、删除etcd实例到etcd集群中。

查看集群中存在的节点

```
$ etcdctl member list
8e9e05c52164694d: name=dev-master-01 peerURLs=http://localhost:2380 clientURLs=http://lo
```

删除集群中存在的节点

```
$ etcdctl member remove 8e9e05c52164694d
Removed member 8e9e05c52164694d from cluster
```

向集群中新加节点

```
$ etcdctl member add etcd3 http://192.168.1.100:2380
Added member named etcd3 with ID 8e9e05c52164694d to cluster
```

参考文档

http://www.google.com (http://www.google.com)
http://t.cn/R5Fw20j (http://t.cn/R5Fw20j)
http://cizixs.com/2016/08/02/intro-to-etcd (http://cizixs.com/2016/08/02/intro-to-etcd)