### ETCD环境搭建

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官方文档：https://etcd.io/docs/v3.4.0/op-guide/maintenance/#auto-compaction

ETCD文档：https://doczhcn.gitbook.io/etcd/

###### etcd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。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也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环境搭建：

* ########################### Part 1 三个节点执行相同操作部分 Start ########################  
    
  ## 按照前准备工作：  
  ## 对三个节点的hostname做修改，自己对三个节点的注解名称进行设置：  
  vim /etc/hostname   
  vim /etc/hosts  
    
  ## 解压安装  
  tar -zxvf etcd-v3.3.18-linux-amd64.tar.gz -C /usr/local/  
  ## 改名：  
  mv /usr/local/etcd-v3.3.18-linux-amd64/ /usr/local/etcd  
  ## 添加环境变量  
  vim /etc/profile  
  ## 内容  
  export ETCDCTL\_API=3  
  ## 执行刷新  
  source /etc/profile  
    
  ## 制作服务，服务内容为：（114、115、116 三节点执行）  
  vim /usr/lib/systemd/system/etcd.service  
    
  ## 制作服务配置  
  [Unit]  
  Description=Etcd Server  
  After=network.target  
  After=network-online.target  
  Wants=network-online.target  
    
  [Service]  
  Type=notify  
  WorkingDirectory=/var/lib/etcd/  
  EnvironmentFile=-/etc/etcd/etcd.conf  
  # set GOMAXPROCS to number of processors  
  ExecStart=/bin/bash -c "GOMAXPROCS=$(nproc) /usr/local/etcd/etcd --name=\"${ETCD\_NAME}\" --data-dir=\"${ETCD\_DATA\_DIR}\" --listen-client-urls=\"${ETCD\_LISTEN\_CLIENT\_URLS}\" --listen-peer-urls=\"${ETCD\_LISTEN\_PEER\_URLS}\" --advertise-client-urls=\"${ETCD\_ADVERTISE\_CLIENT\_URLS}\" --initial-advertise-peer-urls=\"${ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS}\" --initial-cluster=\"${ETCD\_INITIAL\_CLUSTER}\" --initial-cluster-token=\"${ETCD\_INITIAL\_CLUSTER\_TOKEN}\" --initial-cluster-state=\"${ETCD\_INITIAL\_CLUSTER\_STATE}\" --auto-compaction-retention=\"${ETCD\_AUTO\_COMPACTION\_RETENTION}\" --auto-compaction-mode=\"${ETCD\_AUTO\_COMPACTION\_MODE}\" --snapshot-count=\"${ETCD\_SNAPSHOT\_COUNT}\" --quota-backend-bytes=\"${ETCD\_QUOTA\_BACKEND\_BYTES}\""  
    
  Restart=on-failure  
  LimitNOFILE=65536  
    
  [Install]  
  WantedBy=multi-user.target  
    
  ## 创建文件夹  
  mkdir /var/lib/etcd/  
    
  ## 查看文件：  
  cat /usr/lib/systemd/system/etcd.service  
    
  ## 创建资源 （三个节点操作）  
  mkdir /etc/etcd  
  touch /etc/etcd/etcd.conf  
    
  ########################### Part 1 三个节点执行相同操作部分 End ########################  
    
    
  ########################### Part 2 三节点分别添加配置 Start ########################  
    
  ## 三节点分别添加配置（第一个etcd服务：192.168.11.114）  
  vim /etc/etcd/etcd.conf  
    
  ## etcd实例的名称, 默认是default, 在集群中需要唯一,通常使用hostname  
  ETCD\_NAME=bhz114  
  ## 数据的保存路径, 默认是${name}.etcd   
  ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"  
  ## 和同伴通讯的地址  
  ETCD\_LISTEN\_PEER\_URLS="http://192.168.11.114:2380"  
  ## 对外提供服务的地址  
  ETCD\_LISTEN\_CLIENT\_URLS="http://192.168.11.114:2379,http://127.0.0.1:2379"  
  ## 该节点同伴监听地址  
  ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.11.114:2380"  
  ## 对外公告的该节点客户端监听地址  
  ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.11.114:2379"  
  ## 新创建集群的时候,这个值为new,已经存在的集群,这个值为existing  
  ETCD\_INITIAL\_CLUSTER\_STATE="new"  
  ## 创建集群的token, 这个token需要在集群中保持唯一  
  ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster"  
  ## 集群中所有节点的信息  
  ETCD\_INITIAL\_CLUSTER="bhz114=http://192.168.11.114:2380,bhz115=http://192.168.11.115:2380,bhz116=http://192.168.11.116:2380"  
  ## 保持一个小时的历史  
  ETCD\_AUTO\_COMPACTION\_RETENTION="1"  
  ## 使用版本号进行压缩  
  ETCD\_AUTO\_COMPACTION\_MODE="revision"  
  ## 快照触发次数阈值  
  ETCD\_SNAPSHOT\_COUNT=10000  
  ## 设置2G配额, 超过配额告警  
  ETCD\_QUOTA\_BACKEND\_BYTES=2147483648  
    
  ## 添加配置（第二个etcd服务：192.168.11.115）  
  vim /etc/etcd/etcd.conf  
    
  ## etcd实例的名称, 默认是default, 在集群中需要唯一,通常使用hostname  
  ETCD\_NAME=bhz115  
  ## 数据的保存路径, 默认是${name}.etcd   
  ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"  
  ## 和同伴通讯的地址  
  ETCD\_LISTEN\_PEER\_URLS="http://192.168.11.115:2380"  
  ## 对外提供服务的地址  
  ETCD\_LISTEN\_CLIENT\_URLS="http://192.168.11.115:2379,http://127.0.0.1:2379"  
  ## 该节点同伴监听地址  
  ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.11.115:2380"  
  ## 对外公告的该节点客户端监听地址  
  ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.11.115:2379"  
  ## 新创建集群的时候,这个值为new,已经存在的集群,这个值为existing  
  ETCD\_INITIAL\_CLUSTER\_STATE="new"  
  ## 创建集群的token, 这个token需要在集群中保持唯一  
  ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster"  
  ## 集群中所有节点的信息  
  ETCD\_INITIAL\_CLUSTER="bhz114=http://192.168.11.114:2380,bhz115=http://192.168.11.115:2380,bhz116=http://192.168.11.116:2380"  
  ## 保持一个小时的历史  
  ETCD\_AUTO\_COMPACTION\_RETENTION="1"  
  ## 使用版本号进行压缩  
  ETCD\_AUTO\_COMPACTION\_MODE="revision"  
  ## 快照触发次数阈值  
  ETCD\_SNAPSHOT\_COUNT=10000  
  ## 设置2G配额, 超过配额告警  
  ETCD\_QUOTA\_BACKEND\_BYTES=2147483648  
    
  ## 添加配置（第三个etcd服务：192.168.11.116）  
  vim /etc/etcd/etcd.conf  
    
  ## etcd实例的名称, 默认是default, 在集群中需要唯一,通常使用hostname  
  ETCD\_NAME=bhz116  
  ## 数据的保存路径, 默认是${name}.etcd   
  ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"  
  ## 和同伴通讯的地址  
  ETCD\_LISTEN\_PEER\_URLS="http://192.168.11.116:2380"  
  ## 对外提供服务的地址  
  ETCD\_LISTEN\_CLIENT\_URLS="http://192.168.11.116:2379,http://127.0.0.1:2379"  
  ## 该节点同伴监听地址  
  ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.11.116:2380"  
  ## 对外公告的该节点客户端监听地址  
  ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.11.116:2379"  
  ## 新创建集群的时候,这个值为new,已经存在的集群,这个值为existing  
  ETCD\_INITIAL\_CLUSTER\_STATE="new"  
  ## 创建集群的token, 这个token需要在集群中保持唯一  
  ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster"  
  ## 集群中所有节点的信息  
  ETCD\_INITIAL\_CLUSTER="bhz114=http://192.168.11.114:2380,bhz115=http://192.168.11.115:2380,bhz116=http://192.168.11.116:2380"  
  ## 保持一个小时的历史  
  ETCD\_AUTO\_COMPACTION\_RETENTION="1"  
  ## 使用版本号进行压缩  
  ETCD\_AUTO\_COMPACTION\_MODE="revision"  
  ## 快照触发次数阈值  
  ETCD\_SNAPSHOT\_COUNT=10000  
  ## 设置2G配额, 超过配额告警  
  ETCD\_QUOTA\_BACKEND\_BYTES=2147483648  
    
    
  ########################### Part 2 三节点分别添加配置 End ########################  
    
    
  ########################### Part 3 三个节点同时操作 Start ########################  
    
  ### 配置  
  systemctl daemon-reload  
    
  ## 注意：设置开启启动会创建一个软连接：  
  ln -s '/usr/lib/systemd/system/etcd.service' '/etc/systemd/system/multi-user.target.wants/etcd.service'  
    
  systemctl enable etcd  
    
  systemctl start etcd  
    
  ## 查看状态  
  systemctl status etcd  
    
    
  ## systemctl查看启动服务日志：   
  journalctl -f  
  journalctl -u 服务名  
  ########################### Part 3 三个节点同时操作 END ########################  
    
    
    
  ########################### Part 4 验证集群操作 ########################  
    
    
  ## 查看集群成员：  
  /usr/local/etcd/etcdctl member list  
  ## 查看集群基本信息：  
  /usr/local/etcd/etcdctl --write-out=json --endpoints=192.168.11.114:2379,192.168.11.115:2379,192.168.11.116:2379 endpoint status  
    
  /usr/local/etcd/etcdctl --write-out=table --endpoints=192.168.11.114:2379,192.168.11.115:2379,192.168.11.116:2379 endpoint status  
    
  ## 参考部分PS: 如果你想自己手动启动：  
  nohup ./etcd --config-file ./conf/etcd.cluster.conf > ./logs/etcd.log 2>&1 &

###### 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（类似zookeeper）；注意：CRUD即Create,Read,Update,Delete是符合REST风格的一套API操作；

###### 常用命令操作：

* ### etc 实时日志查看：  
  journalctl -f -u etcd  
    
  ## 进入etcd目录，使用etcdctl脚本进行常用命令操作  
  cd /usr/local/etcd  
    
    
  ## 1. put 指定某个键的值。例如:  
  etcdctl put /test/key1 "Hello world"  
    
    
  ## 2. get 获取指定键的值。例如：  
  etcdctl get /test/key1  
  etcdctl --write-out="json" get /test/key1  
    
  ## 3. put 更新指定的值。例如：  
  etcdctl put /test/key1 "baihezhuo"  
    
    
  ## 4. del 删除指定的key。例如：  
  etcdctl put /test/key2 1234  
  etcdctl del /test/key2  
    
    
  ## 5. 获取所有前缀的信息  
  etcdctl get --prefix /test  
  ## 指定最大获取2条信息  
  etcdctl get --prefix --limit=2 /test  
    
    
  ## 6. 删除所有前缀的信息  
  etcdctl del --prefix /test  
    
    
  ## 7. watch key   
  ## 114节点watch test：  
  etcdctl watch /test  
  ## 115节点进行修改内容：  
  etcdctl put /test "base"  
  ## 116节点进行删除内容，这里千万注意，他们不是真正的目录结构，删除/test 并不会删除/test/key1等子节点  
  etcdctl del /test  
    
  ## 前缀watch，可以实现类似zk的子节点监听  
  etcdctl watch /test --prefix  
    
    
  ## 8. lease 申请租约，续约，查看租约时间  
    
  ## 查看租期列表  
  etcdctl lease list  
  found 0 leases  
  ## 申请一个200s的租约  
  etcdctl lease grant 200  
  -- lease 6fa06eb09efa986e granted with TTL(200s)  
  ## 续约租期  
  etcdctl lease keep-alive 6fa06eb09efa986e  
  ## 查看租约剩余时间  
  etcdctl lease timetolive 6fa06eb09efa986e  
  ## 撤销租约  
  etcdctl lease revoke 6fa06eb09efa986e  
  ## 添加内容并配置相应的租约  
  etcdctl put /test/key3 "linghui" --lease=6fa06eb09efa986e  
  ## 获取内容：  
  etcdctl get /test/key3  
    
  ## 当租约到期时会打印如下：  
  [root@bhz114 etcd]# etcdctl lease timetolive 6fa06eb09efa986e  
  lease 6fa06eb09efa986e already expired  
  [root@bhz114 etcd]# etcdctl get /test/key3  
    
    
  ## 9. 分布式锁 lock  
  etcdctl --endpoints=$ENDPOINTS lock mutex1  
  mutex1/6fa06eb09efa98d1  
  # another client with the same name blocks  
  etcdctl --endpoints=$ENDPOINTS lock mutex1  
    
  ## 10. txn 事务操作  
  etcdctl put /test/key4 "txn1"  
    
  ## 进入事务  
  etcdctl txn --interactive  
  ## 比较：/test/key4=txn 则执行success requests操作，/test/key4 != txn 则执行failure requests操作  
  compares:  
  value("/test/key4") = "txn2"   
    
  success requests (get, put, delete):  
  put /test/key4 ok  
    
  failure requests (get, put, delete):  
  put /test/key4 no  
    
  ## 最终结果/test/key4 为no  
  etcdctl get /test/key4
* etcdKeeper可视化插件：
* ## 解压  
  unzip etcdkeeper-v0.7.5-linux\_x86\_64.zip -d /usr/local/  
  ## 启动命令   
  ./etcdkeeper -h 192.168.11.114 -p 8080 &  
  ## 访问地址  
  http://192.168.11.114:8080
* java操作etcd：https://www.jianshu.com/p/4be30c0716b4
* 常用命令列表：
* COMMANDS:  
   get Gets the key or a range of keys  
   put Puts the given key into the store  
   del Removes the specified key or range of keys [key, range\_end)  
   txn Txn processes all the requests in one transaction  
   compaction Compacts the event history in etcd  
   alarm disarm Disarms all alarms  
   alarm list Lists all alarms  
   defrag Defragments the storage of the etcd members with given endpoints  
   endpoint health Checks the healthiness of endpoints specified in `--endpoints` flag  
   endpoint status Prints out the status of endpoints specified in `--endpoints` flag  
   endpoint hashkv Prints the KV history hash for each endpoint in --endpoints  
   move-leader Transfers leadership to another etcd cluster member.  
   watch Watches events stream on keys or prefixes  
   version Prints the version of etcdctl  
   lease grant Creates leases  
   lease revoke Revokes leases  
   lease timetolive Get lease information  
   lease list List all active leases  
   lease keep-alive Keeps leases alive (renew)  
   member add Adds a member into the cluster  
   member remove Removes a member from the cluster  
   member update Updates a member in the cluster  
   member list Lists all members in the cluster  
   snapshot save Stores an etcd node backend snapshot to a given file  
   snapshot restore Restores an etcd member snapshot to an etcd directory  
   snapshot status Gets backend snapshot status of a given file  
   make-mirror Makes a mirror at the destination etcd cluster  
   migrate Migrates keys in a v2 store to a mvcc store  
   lock Acquires a named lock  
   elect Observes and participates in leader election  
   auth enable Enables authentication  
   auth disable Disables authentication  
   user add Adds a new user  
   user delete Deletes a user  
   user get Gets detailed information of a user  
   user list Lists all users  
   user passwd Changes password of user  
   user grant-role Grants a role to a user  
   user revoke-role Revokes a role from a user  
   role add Adds a new role  
   role delete Deletes a role  
   role get Gets detailed information of a role  
   role list Lists all roles  
   role grant-permission Grants a key to a role  
   role revoke-permission Revokes a key from a role  
   check perf Check the performance of the etcd cluster  
   help Help about any command
* 1
* 1
* 1