# tidb运维基础操作

- 配置文件变更
  - 1.中控机修改配置文件/home/tidb/tidb-ansible/conf
  - 2.修改对应文件
  - 3.执行更新命令
    - ansible-playbook rolling_update.yml --tags=tidb
  - 4.验证
    - cd /export/tidb/deploy/conf/
    - 配置文件是否修改
    - 验证配置是否生效，可以通过 log 日志进行确认，例如查看 tidb 日志是否生效：grep -i " welcome" tidb.log
- Sysbench压力测试
  - 安装Sysbench
  - 修改sysbench配置文件
  - tidb创建用户名等信息
  - 在客户端执行set global tidb_disable_txn_auto_retry = off;
    - 参数tidb_disable_txn_auto_retry = off
  - 压测数据准备，压测脚本建议按照官方文档去修改
  - 开始压测
- 扩缩容
  - 扩缩容tidb
    - 扩容tidb
      - 1.修改配置hosts.ini 和 inventory.ini文件 添加节点信息
      - 初始化新节点 创建用户 ntpd等
      - 初始化内核参数
        - ansible-playbook bootstrap.yml -l 172.16.0.15
      - 部署新节点
        - ansible-playbook deploy.yml -l 172.16.0.15
      - 启动新增节点
        - ansible-playbook start.yml -l 172.16.0.15
      - 更新prometheus配置并重启prometheus
        - ansible-playbook rolling_update_monitor.yml --tags=prometheus
    - 缩容tidb
      - 修改配置hosts.ini 和 inventory.ini文件 注释节点信息
      - 更新 prometheus配置并且重启
        - ansible-playbook rolling_update_monitor.yml --tags=prometheus
  - 扩缩容tikv
    - 扩容tikv 参考扩容tidb
    - 缩容tikv
      - pd-ctl 工具移除节点信息
      - 停止服务，停止服务前需要确认 store 状态变为 Tombstone
        - ansible-playbook stop.yml -l 172.16.0.6
      - 编辑inventory.ini 文件，移除节点信息
      - 新prometheus配置并重启prometheus
        - ansible-playbook rolling_update_monitor.yml --tags=prometheus
      - 查看状态
  - 扩缩容pd
    - 扩容pd，参考tidb部分内容
    - 修改run_pd.sh脚本
      - 在新的pd节点启动pd服务
    - 启动服务
      - ansible-playbook start.yml -l 172.16.0.15
    - 更新集群
      - ansible-playbook deploy.yml
    - 更新prometheus配置并重启prometheus
      - ansible-playbook rolling_update_monitor.yml --tags=prometheus
    - 缩容pd
      - pd-ctl 工具移除
      - 停止服务
        - ansible-playbook stop.yml -l 172.16.0.6
      - 更新集群的配置
        - ansible-playbook deploy.yml
      - 更新prometheus配置并重启prometheus
        - ansible-playbook rolling_update_monitor.yml --tags=prometheus
- 慢查询排查
  - 操作系统类
    - cpu
      - load
      - core
      - 操作命令
        - top
    - 内存
      - free
      - use
      - total
      - swap
      - 操作命令
        - vmstat
    - 磁盘io
      - use
      - free
      - total
      - iops
      - 操作命令
        - iostat
          - iostat -m -dx
        - iotop
    - 网络
      - recv
      - traffic
      - 网络延迟&丢包
      - 操作命令
        - ping
          - ping -f 大量的ping
          - ping -l 大包ping
        - iftop
  - tidb&tikv
    - tidb排查
      - 慢查询日志
        - select * from information_schema.slow_query order by query_time desc limit 10
        - tidb_slow_query.log
    - tikv排查
      - 通过tikv.log查看table_id
        - curl http://10.206.0.6:10080/schema?table_id=143 获取table 具体信息
      - 通过查tikv.log 查看wait_time或者其他指标
        - /home/tidb/tidb-ansible/resources/bin/pd-ctl -u http://10.206.0.12:2379 -d store ?
  - 案例 应用慢 tidb不慢
    - 1.网络方向
      - 应用到tidb server直接的网络
      - tidb server 到tikv和pd 节点直接的网络 集群网络状态
    - 2.应用连接池配置
    - 3.高并发场景
  - 案例二 应用慢 tidb慢
    - sql方向
      - 1.隐式转换
      - 2.无索引
      - 3.执行计划
      - 4.统计信息
    - 热点key方向
    - 集群操作系统性能指标方向
      - tidb/pd性能高
        - 整体性能搞
        - 某一台性能高
        - 所有tidb/pd性能都高
      - tikv性能高
        - 某一台tikv性能高
          - 大概率是有热点key
          - 无热点key 排查系统方向
        - 整体性能都高
          - 参数配置
          - 是否执行过升级等变更
    - 锁方向
      - 普通的锁
      - 备份锁或其他全局锁
- tidb集群基础命令
  - ansible-playbook -i hosts.ini create_users.yml -u root -k
    - 配置用户 ssh 免密 sudo等
  - ansible-playbook local_prepare.yml
    - 联网下载tidb 安装包
  - ansible-playbook -i hosts.ini deploy_ntp.yml -u tidb -b
    - 配置ntpd
  - ansible -i inventory.ini all -m shell -a ' whoami' -b
    - 检测用户
  - ansible-playbook bootstrap.ym
    - 环境检测 内核设置等
      - 测试环境初始化可跳过检查 ansible-playbook bootstrap.yml -e dev_mode=true
  - ansible-playbook deploy.yml
    - 集群部署
  - ansible-playbook rolling_update.yml
    - 更新集群
  - ansible-playbook start.yml
    - 启动集群
  - ansible-playbook stop.yml
    - 关闭集群
  - ansible-playbook rolling_update_monitor.yml --tags=prometheus
    - 更新 prometheus配置
- 架构简介
  - tidb
    - 1.负责接收SQL请求，处理SQL相关的逻辑
    - 2.PD找到存储计算所需数据的TiKV地址，TiKV交互数据，最终返回结果
    - 3.TiDB不存储数据，只负责计算
    - 4.可以使用LVS、HAProxy、F5对外提供统一地址
  - tikv
    - 1.分布式提供事务的Key-Value存储引擎
    - 2.存储数据的基本单位是Rejoin
    - 3.tikv使用Raft协议，保证数据的一致性和容灾
    - 4.副本以 Region 为单位进行管理，不同节点上的多个 Region 构成一个 Raft Group
  - pd
    - 1. 存储集群的元信息（某个 Key 存储在哪个 TiKV 节点）
    - 2.是对 TiKV 集群进行调度和负载均衡（如数据的迁移、Raft group leader 的迁移等）
    - 3.三是分配全局唯一且递增的事务 ID
    - 4.PD通过Raft协议保证数据的安全性
- 安装部署
  - 1.控制机（Ansible）安装系统依赖包
    - yum -y install python2-pip epel-release git curl sshpass
  - 2.控制机 创建tidb用户设置密码，生成ssh秘钥和sudo规则，安装ansible包
    - 注:执行ansible --version 会报错 NameError: name 'platform_system' is not defined 执行pip install --upgrade pip pip install --upgrade setuptools 即可
  - 3. 在git中拉取tidb-ansible包。控制机执行命令下载tidb 二进制文件包
    - ansible-playbook local_prepare.yml 联网下载tidb到控制机
  - 4.修改配置hosts.ini文件和inventory.ini
  - 5.所有主机创建用户，配置tidb用户免密
    - ansible -i inventory.ini all -m shell -a ' whoami' -b 表示tidb 用户sudo免密密码配置成功
  - 6.配置ntpd、cpureq、ext4
    - ntpd： ansible-playbook -i hosts.ini deploy_ntp.yml -u tidb -b
      - 注：在配置的过程中重启主机 可能会报ntpd错误，重启ntpd服务即可
    - 配置cpu调节模式（非必需）：ansible -i hosts.ini all -m shell -a "cpupower frequency-set --governor performance" -u tidb -b
    - 手动操作所有tikv主机data盘: ext4需要添加挂载参数 nodelalloc,noatime
  - 7.执行初始化任务：优化所有主机内核参数，检查系统硬件信息
    - ansible-playbook bootstrap.yml 修改内核参数
    - 测试环境 ansible-playbook bootstrap.yml -e dev_mode=true
  - 8.执行部署任务(包括部署监控任务)
    - ansible-playbook deploy.yml 部署tidb集群
  - 9.启动集群和监控
    - ansible-playbook start.yml 启动tidb
- 升级版本
  - 1.备份原先的ansible
  - 2.git中拉取新版本的tidb-ansible
  - 3.修改新版本的hosts.ini和inventory.ini 以及旧的配置文件同步，如果有修改过配置文件，需要修改 conf 目录下对应组件的配置文件
  - 4.执行升级操作，滚动升级
    - ansible-playbook rolling_update.yml 升级命令
  - 5.验证升级是否成功
  - 6.建议业务低峰期运行，做好配置文件备份，旧的参数信息需要手动同步到新的集群中
- DDL工具
  - ddl调度参数
    - 查看正在执行的 DDL 任务
      - ADMIN SHOW DDL;
    - 查看最近 N条已经执行完成的 DDL 任务
      - ADMIN SHOW DDL JOBS N;
    - 查看未执行完成的 DDL 任务
      - ADMIN SHOW DDL JOBS N WHERE state!='synced'
    - 根据 JOB_ID 查询具体的 DDL 语句
      - ADMIN SHOW DDL JOB QUERIES job_id
    - 取消正在执行中的 DDL 任务
      - ADMIN CANCEL DDL JOBS job_id
    - 注：在执行一些 DDL 操作时（如 ADD INDEX ），由于执行时间较长,不会立即返回执行结果,此时可以放心的 ctrl+c 来终止该连接，不会影响 DDL 的实际执行
  - ddl相关参数
    - tidb_ddl_reorg_worker_cnt
      - 控制数据回填（re-organize）阶段的并发度
    - tidb_ddl_reorg_batch_size
      - 控制数据回填（re-organize）阶段一次回填的数据量
    - tidb_ddl_reorg_priority
      - 控制数据回填（re-organize）阶段执行的优先级
    - tidb_ddl_error_count_limit
      - 控制 DDL 操作失败重试的次数，重试次数超过该值，则取消 DDL 操作
  - 为啥加索引时间那么长
    - 普通 DDL
      - 普通 DDL 操作执行时间短，一般秒级就可以执行完成
    - 添加索引
      - 索引操作由于需要回填数据，因此执行时间略长，而在回填数据期间，需要将回填的数据写入 TiKV，对 TiKV 会产生额外的写入压力，从而造成一些性能影响
- 权限认证
  - 权限记录表
    - mysql.user
    - mysql.tables_priv
    - mysql.db
  - RBAC
    - 介绍：RBAC 基于角色的权限访问控制
    - 感觉就是MySQL8.0的角色
  - 安全注意
    - 1.tidb 默认localhost密码为空
    - 2.是否考虑参考MySQL5.7一样创建好集群随设置机密码
    - 3.MySQL5.7 最新版本中默认设置了密码复杂度

XMind：ZEN | 试用模式