

云计算应用管理

NSD ENGINEER

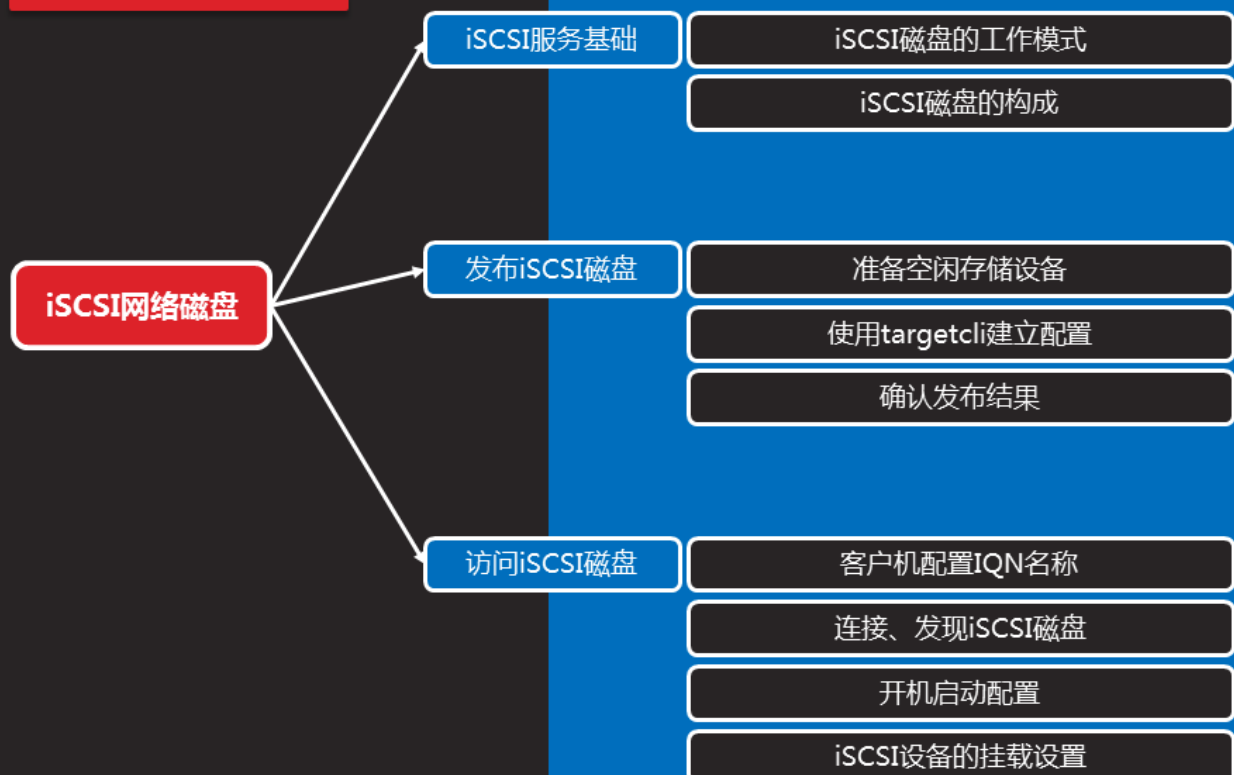
DAY05

内容

| | | |
|----|---------------|-----------|
| 上午 | 09:00 ~ 09:30 | 作业讲解和回顾 |
| | 09:30 ~ 10:20 | iSCSI共享存储 |
| | 10:30 ~ 11:20 | |
| | 11:30 ~ 12:00 | 数据库服务基础 |
| 下午 | 14:00 ~ 14:50 | |
| | 15:00 ~ 15:50 | 管理表数据 |
| | 16:10 ~ 17:00 | |
| | 17:10 ~ 18:00 | 总结和答疑 |



iSCSI网络磁盘

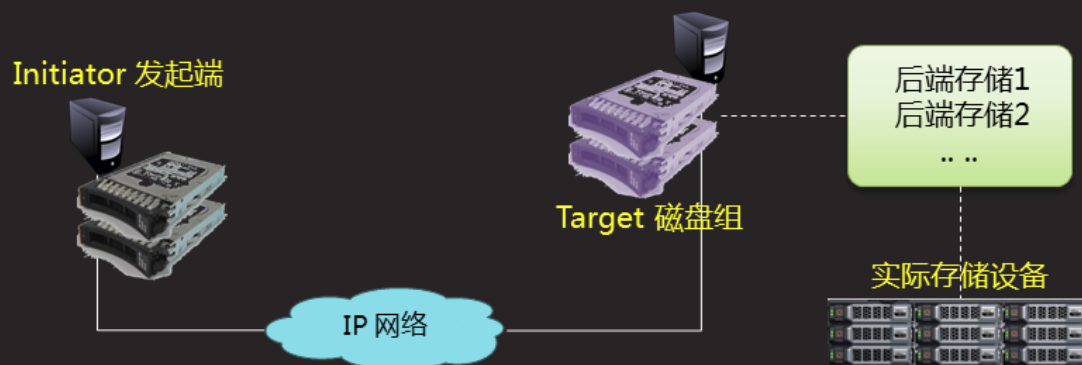


iSCSI服务基础

iSCSI磁盘的工作模式

- Internet **SCSI**，网际SCSI接口
 - 一种基于C/S架构的虚拟磁盘技术
 - 服务器提供磁盘空间，客户机连接并当成本地磁盘使用

知识讲解



iSCSI磁盘的构成

知识讲解

- backstore , 后端存储
 - 对应到服务端提供实际存储空间的设备 , 需要起一个管理名称
- target , 磁盘组
 - 是客户端的访问目标 , 作为一个框架 , 由多个lun组成
- lun , 逻辑单元
 - 每一个lun需要关联到某一个后端存储设备 , 在客户端会视为一块虚拟硬盘



发布iSCSI磁盘

准备空闲存储设备

知识讲解

- 使用 fdisk 分区工具

- fdisk /dev/vdb
- n、p、1、默认、+3G、w

```
[root@server0 ~]# fdisk -l /dev/vdb
```

```
.. ..
```

| 设备 | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|---------|---------|----|--------|
| /dev/vdb1 | | 2048 | 6293503 | 3145728 | 83 | Linux |



使用targetcli建立配置

知识讲解

- **ISCSI Qualified Name** 名称规范
 - iqn.yyyy-mm.倒序域名:自定义标识
 - 用来识别 target 磁盘组，也用来识别客户机身份
- 名称示例
 - iqn.2016-02.com.example:server0
 - iqn.2016-02.com.example:desktop0
- 所需软件包：targetcli

```
[root@server0 ~]# yum -y install targetcli
```

```
.. ..
```



使用targetcli建立配置 (续1)

知识讲解

- 基本交互操作
 - /backstore/block create 后端存储名 实际设备路径
 - /iscsi create 磁盘组的IQN名称
 - /iscsi/磁盘组名/tpg1/luns create 后端存储路径
 - /iscsi/磁盘组名/tpg1/acls create 客户机IQN标识
 - /iscsi/磁盘组名/tpg1/portals create IP地址 端口号
- 其他辅助配置指令
 - ls、saveconfig、exit



确认发布结果

知识讲解

- 重启 target 服务，检查配置结果

```
[root@server0 ~]# systemctl restart target
[root@server0 ~]# targetcli ls
o- / ..... [...]
.. ..
| | o- iscsi_store [/dev/vdb1 (3.0 GiB) write-thru activated]
o- iscsi ..... [Targets: 1]
| o- iqn.2016-02.com.example:server0 ..... [TPGs: 1]
|   o- acls ..... [ACLs: 1]
|   | o- iqn.2016-02.com.example:desktop0 [Mapped LUNs: 1]
.. ..
|   | o- lun0 ..... [block/iscsi_store (/dev/vdb1)]
|   o- portals ..... [Portals: 1]
|     o- 172.25.0.11:3260 ..... [OK]
.. ..
```



案例1：发布iSCSI网络磁盘

课堂练习

- 配置 server0 提供 iSCSI 服务，要求如下：
 - 磁盘名为iqn.2016-02.com.example:server0
 - 服务端口为 3260
 - 使用 iscsi_store 作其后端卷，其大小为 3GiB
 - 此磁盘服务只能被 desktop0.example.com 访问



访问iSCSI磁盘

客户机配置IQN名称

知识讲解

- 修改 /etc/iscsi/initiatorname.iscsi
 - 后续再更改时，需重启服务 iscsid 以更新IQN标识

```
[root@desktop0 ~]# vim /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2016-02.com.example:desktop0
```

```
[root@desktop0 ~]# systemctl restart iscsid //刷新IQN标识
```



连接、发现iSCSI磁盘

知识讲解

- 所需软件包：iscsi-initiator-utils
- 发现iSCSI磁盘
 - iscsiadm -m discovery -t st -p 服务器地址[:端口]
- 连接iSCSI磁盘
 - iscsiadm -m node -L all

```
[root@desktop0 ~]# iscsiadm -m discovery -t st -p server0
172.25.0.11:3260,1 iqn.2016-02.com.example:server0
```

```
[root@desktop0 ~]# iscsiadm -m node -L all
Login to [iface: default, target: iqn.2016-02.com.example:server0,
portal: 172.25.0.11,3260] successful.
```



连接、发现iSCSI磁盘 (续1)

知识讲解

- 确认连接情况
 - 检查多出的磁盘设备 : `lsblk`
 - 检查磁盘记录 : `iscsiadm -m node -T target名称`
- 断开、删除记录
 - `iscsiadm -m node -U all`
 - `iscsiadm -m node -T target名称 -o delete`

```
[root@desktop0 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0 3.0G 0 disk
vda   253:0   0 10G 0 disk
└─vda1 253:1   0 10G 0 part /
... ..
```

后续就像使用本地磁盘一样了，
作分区、格式化等操作



开机启动配置

知识讲解

- 新发现的 iSCSI 磁盘记录
 - 自动存放在 `/var/lib/iscsi/nodes/.. ..` 相关目录下
 - 若需要开机自动连接，需要调整记录配置

```
[root@desktop0 ~]# vim /var/lib/iscsi/nodes/*/*/default
... ..
node.startup = automatic
... ..
node.conn[0].startup = automatic      //开机时自动连接
... ..
```

```
[root@desktop0 ~]# systemctl enable iscsi
```



iSCSI设备的挂载设置

知识讲解

- 关于 iSCSI 磁盘的名称
 - 当连接多个可用的 target 时，磁盘顺序可能会变化
 - 在 /etc/fstab 挂载配置中，建议通过 `UUID=.. ..` 的方式来取代设备名

```
[root@desktop0 ~]# blkid /dev/sda1  
/dev/sda1: UUID="5b584829-.. .." TYPE="ext4"
```

```
.. ..  
[root@desktop0 ~]# vim /etc/fstab
```

```
.. ..  
UUID="5b584829-.. .." /mnt/data ext4 _netdev 0 0
```

```
[root@desktop0 ~]# mount -a
```



案例2：访问iSCSI网络磁盘

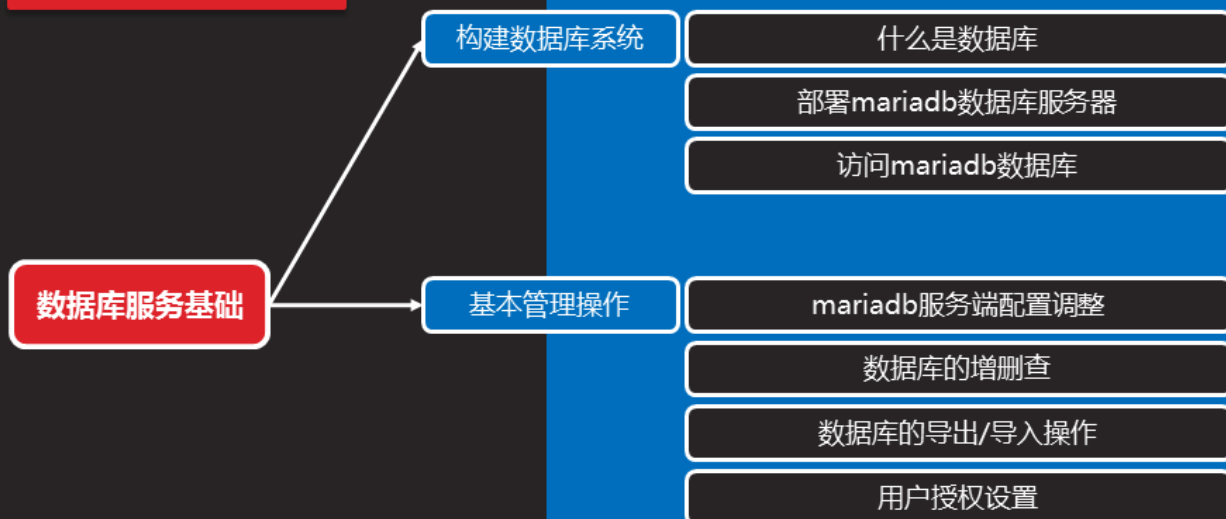
课堂练习

配置 desktop0 使用 iSCSI 磁盘，要求如下：

- 1) 目标磁盘组为 `iqn.2016-02.com.example:server0`
- 2) 此 iSCSI 设备在系统启动的期间自动加载
- 3) 此 iSCSI 块设备上包含一个大小为 2100MiB 的分区，并将其格式化为 ext4 文件系统
- 4) 此分区应该挂载到 /mnt/data 上，同时在系统启动的期间自动挂载



数据库服务基础



构建数据库系统

什么是数据库

知识讲解

- DB，数据库：一批数据的集合，主流的数据库多用来存放关系型表格数据
- 关系型数据：以二维表格记录大量实体的属性信息
- DBMS，数据库管理系统
 - 用来操作和管理数据库的大型软件平台

7号库

| | | |
|-------|----------|-------------|
| 编号 | TTS账号 | 月考成绩 |
| 编号 | 姓名 | 手机号 |
| 1 | zhangsan | 13712345678 |
| 2 | lisi | 13712345679 |
| 3 | wangwu | 13712345680 |

每一行

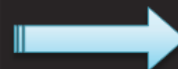
每一列



什么是数据库（续1）

知识讲解

- 常见的关系型 数据库管理系统
 - 微软的 SQL Server
 - IBM的 DB2
 - 甲骨文的 Oracle、MySQL
 - 社区开源版 MariaDB
 -



部署mariadb数据库服务器

知识讲解

- RHEL7 中的 MariaDB 相关包
 - mariadb-server : 提供服务端有关的系统程序
 - mariadb : 提供客户端及管理工具

```
[root@server0 ~]# yum -y install mariadb-server mariadb //装包
.. ..
[root@server0 ~]# systemctl restart mariadb //起服务
.. ..
[root@server0 ~]# netstat -antpu | grep :3306 //查看监听端口
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN 1981/mysqld
```



访问mariadb数据库

知识讲解

- 使用 mysql 命令
 - 登入交互界面，实现数据库管理操作
 - mysql [-u用户名] [-p[密码]]

```
[root@server0 ~]# mysql -uroot //默认无密码
.. ..
MariaDB [(none)]> SHOW DATABASES; //测试列库操作
.. ..
MariaDB [(none)]> QUIT; //退出
Bye
[root@server0 ~]#
```

管理指令不区分大小写，要求以分号结尾



案例3：搭建mariadb数据库系统

课堂练习

1. 安装 MariaDB 数据库系统

- 1) 安装 mariadb-server、mariadb 软件包
- 2) 启动 mariadb 服务，并确认监听状态

2. 客户端访问验证

- 1) 使用 mysql 命令访问本机的数据库服务，用户名为 root，密码为空
- 2) 执行 SHOW DATABASES; 指令列出有哪些库
- 3) 退出 mysql 交互界面



基本管理操作

mariadb服务端配置调整

知识讲解

- 禁止监听，只服务于本机

```
[root@server0 ~]# vim /etc/my.cnf
[mysqld]
skip-networking                      //跳过网络监听
.. ..
[root@server0 ~]# systemctl restart mariadb
```

- 为数据库账号修改密码

– `mysqladmin [-u用户名] [-p[旧密码]] password '新密码'`

```
[root@server0 ~]# mysqladmin -u root password 'atenorth'
[root@server0 ~]# mysql -uroot -patenorth
MariaDB [(none)]>                      //测试新密码登录成功
.. ..
```



数据库的增删查

知识讲解

- MariaDB [(none)]> 交互指令

- 列出数据库：`SHOW DATABASES;`
- 使用/选择数据库：`USE 数据库名;`
- 列出库里有哪些表：`SHOW TABLES;`
- 创建数据库：`CREATE DATABASE 数据库名;`
- 删除数据库：`DROP DATABASE 数据库名;`

```
MariaDB [(none)]> CREATE DATABASE Contacts;
.. ..                      //创建名为 Contacts 的库
```



数据库的导出/导入操作

知识讲解

- 导出/备份数据库
 - `mysqldump [-u用户名] [-p[密码]] 数据库名 > 备份文件.sql`
- 导入/恢复到数据库
 - `mysql [-u用户名] [-p[密码]] 数据库名 < 备份文件.sql`

```
[root@server0 ~]# wget http://classroom/pub/materials/users.sql
[root@server0 ~]# mysql -u root -p'atenorth' Contacts < users.sql
```

```
.. ..
```

//下载备份，导入到 Contacts 库



用户授权设置

知识讲解

- MariaDB [(none)]> 交互指令
 - `GRANT 权限列表 ON 数据库名.表名 TO 用户名@客户机地址 IDENTIFIED BY '密码';`

```
MariaDB [(none)]> GRANT select ON Contacts.* to
Raikon@localhost IDENTIFIED BY 'atenorth';
```

```
.. ..
```

//授予用户 Raikon 查询 Contacts 库的权限



案例4：配置一个数据库

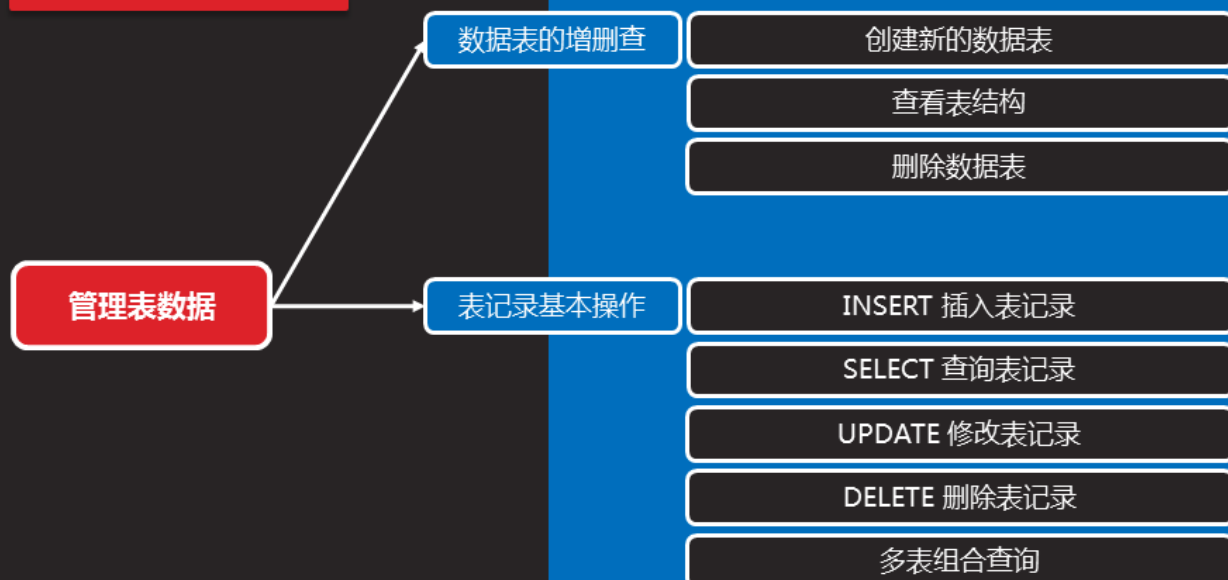
在 server0 上部署 MariaDB 数据库

- 此数据库系统只能被 localhost 访问
- 新建一个数据库名为 Contacts，其中应该包含来自数据库复制的内容，复制文件的 URL 为：
<http://classroom/pub/materials/users.sql>
- 除了 root 用户，此数据库只能被用户 Raikon 查询，此用户的密码为 atenorth
- root 用户的密码为 atenorth

课堂练习



管理表数据



数据表的增删查

创建新的数据表

- MariaDB [(none)]> 交互指令
 - CREATE TABLE 表名(字段1名 类型(长度), 字段2名 类型(长度),);
- ```
MariaDB [(none)]> CREATE DATABASE lianxi; //建库
MariaDB [(none)]> USE lianxi; //切换库
MariaDB [Contacts]> CREATE TABLE base(id int, name varchar(20),
password varchar(24)); //建表
.. ..
```



## 查看表结构

- MariaDB [(none)]> 交互指令

– DESCRIBE 表名;

知识讲解

```
MariaDB [lianxi]> DESCRIBE base; //查看表的字段信息
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id | int(11) | YES | | NULL | |
| name | varchar(20) | YES | | NULL | |
| password | varchar(24) | YES | | NULL | |
+-----+-----+-----+-----+-----+
```



## 删除数据表

- MariaDB [(none)]> 交互指令

– DROP TABLE 表名;

知识讲解

```
MariaDB [lianxi]> DROP TABLE base; //删除表
.. ..
MariaDB [lianxi]> DESC base; //确认已删除
ERROR 1146 (42S02): Table 'lianxi.base' doesn't exist
```



# 表记录基本操作

## INSERT 插入表记录

- MariaDB [(none)]> 交互指令
  - INSERT INTO [数据库.]表名 VALUES(值1, 值2, .. ..);

知识讲解

```
MariaDB [lianxi]> CREATE TABLE base(id int, name varchar(20),
password varchar(24)); //重建表，方便测试
.. ..
MariaDB [lianxi]> INSERT INTO base VALUES(1,'Tom','123');
MariaDB [lianxi]> INSERT INTO base VALUES(2,'Barbara','456');
.. .. //插入2条记录
```



# SELECT 查询表记录

知识讲解

- MariaDB [(none)]> 交互指令
  - SELECT \* FROM [数据库.]表名;
  - SELECT 字段1,字段2,... FROM [数据库.]表名;
  - SELECT ... WHERE 字段1='值' and|or 字段2='值';

```
MariaDB [lianxi]> SELECT name FROM base WHERE
password='456'; //查询name字段
```

```
+-----+
| name |
+-----+
| Barbara |
+-----+
... ..
```



# UPDATE 修改表记录

知识讲解

- MariaDB [(none)]> 交互指令
  - UPDATE [数据库名.]表名 SET 字段名=新值
    - WHERE 条件语句;

```
MariaDB [lianxi]> UPDATE base SET password='pwd123'
WHERE name='Tom'; //修改密码
```

```
.. ..
MariaDB [lianxi]> SELECT * FROM base; //检查修改结果
```

```
+-----+-----+-----+
| id | name | password |
+-----+-----+-----+
| 1 | Tom | pwd123 |
| 2 | Barbara | 456 |
+-----+-----+-----+
```



# DELETE 删除表记录

- MariaDB [(none)]> 交互指令
  - DELETE FROM [数据库.]表名 WHERE 条件语句;

```
MariaDB [lianxi]> DELETE FROM base WHERE id=1;
Query OK, 1 row affected (0.02 sec) //删除记录
```

```
MariaDB [lianxi]> SELECT * FROM base; //检查修改结果
```

```
+-----+-----+-----+
| id | name | password |
+-----+-----+-----+
| 2 | Barbara | 456 |
+-----+-----+-----+
```

知识讲解



# 多表组合查询

- 统计查询结果数量 : SELECT count(\*) .. ..;
- 关联查询 : ... FROM 表1,表2 WHERE 表1.字段1="值..." AND 表2.字段2="值..." AND 表1.字段1=表2.字段2;

```
MariaDB [lianxi]> use Contacts; //切换库
MariaDB [Contacts]> SELECT count(*) FROM base,location
WHERE base.name="Barbara" AND location.city="Sunnyvale"
AND base.id=location.id; //组合查询
```

```
+-----+
| count(*) |
+-----+
| 1 |
+-----+
```

知识讲解



## 案例5：使用数据库查询

课堂练习

1. 禁止空密码root用户访问 mariadb 数据库
2. 在系统 server0 上使用数据库 Contacts，并使用相应的 SQL 查询以回答下列问题：
  - 1) 密码是 solicitous 的人的名字？
  - 2) 有多少人的姓名是 Barbara 同时居住在 Sunnyvale？



### 总结和答疑

总结和答疑

mariadb数据库

问题现象

故障分析及排除

# mariadb数据库

## 问题现象

- 维护 mariadb 数据库时操作失败
  - 问题1：导入 users.sql 数据库失败
  - 问题2：丢失数据库用户 root 的密码，无法连接

知识讲解

```
[root@server0 ~]# mysqldump -uroot -patenorth < users.sql
Usage: mysqldump [OPTIONS] database [tables]
```

```
.. ..
```

```
[root@server0 ~]# mysql -uroot -patenorth
ERROR 1045 (28000): Access denied for user 'root'@'localhost'
(using password: YES)
```





# 故障分析及排除

## 知识讲解

- 原因分析
  - 问题1：命令用错了，恢复库应使用 `mysql` 命令
  - 问题2：当密码不匹配时会拒绝登录
- 解决办法
  - 问题1：将 `mysqldump` 改为 `mysql` 再执行导入
  - 问题2：清除库再覆盖安装 `mariadb-server`

```
[root@server0 ~]# systemctl stop mariadb //停服务
[root@server0 ~]# rm -rf /var/lib/mysql/mysql/ //清理用户库
[root@server0 ~]# yum -y reinstall mariadb-server //覆盖安装
...
[root@server0 ~]# systemctl start mariadb //起服务
```

