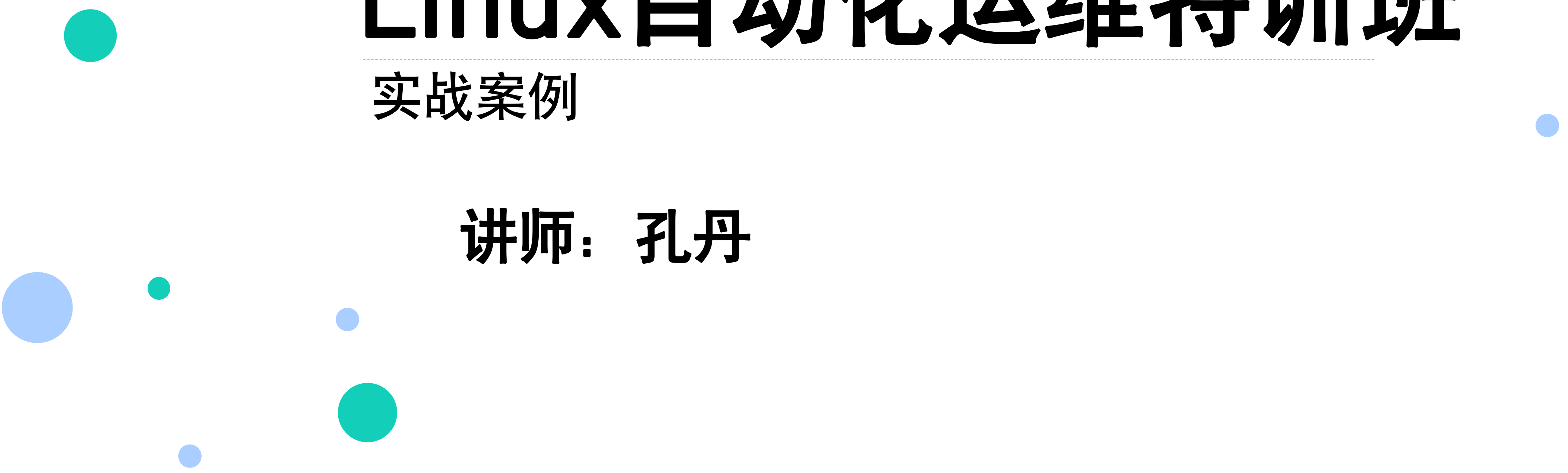




Linux自动化运维特训班

实战案例

讲师：孔丹



大纲

- 服务脚本
- 备份脚本
- 监控脚本
- 安全级别

服务脚本

□ 1、编写源码安装系统服务启动脚本

分别基于CentOS6和7开发源码安装后系统服务脚本。
要求脚本具备启动、停止等基本功能。

RHEL6:

```
#!/bin/sh
#
# nginx - this script starts and stops the nginx daemon
#
# chkconfig:   - 85 15
# description: Nginx is an HTTP(S) server, HTTP(S) reverse \
#               proxy and IMAP/POP3 proxy server
```

服务脚本放置路径: /etc/init.d

脚本增加执行权限: `chmod +x /etc/init.d/nginx`

添加成系统服务脚本: `chkconfig --add nginx`

设置开机自启动: `chkconfig --level 35 nginx on`

服务脚本

- 1、编写源码安装系统服务启动脚本
分别基于CentOS6和7开发源码安装后系统服务脚本。
要求脚本具备启动、停止等基本功能。

RHEL7:

systemd是RH7系列操作系统开始启用新的系统和服务管理器
systemd中引入了system units的概念,在units其中封装有关系统服务
(service),侦听套接字(socket),以及与init系统启动相关信息

unit文件保存位置

/usr/lib/systemd/system/		RPM包安装时分发的unit文件
/run/systemd/system/		systemd运行时创建的文件
/etc/systemd/system/		systemctl enable创建的unit文件

服务脚本

□ 案例1、编写源码安装系统服务启动脚本

分别基于CentOS6和7开发源码安装后系统服务脚本。

要求脚本具备启动、停止等基本功能。

RHEL7:

```
vim /usr/lib/systemd/system/nginx.service
```

```
[Unit]
```

```
Description=nginx - high performance web server
```

```
Documentation=http://nginx.org/en/docs/
```

```
After=network.target remote-fs.target nss-lookup.target
```

```
[Service]
```

```
Type=forking
```

```
PIDFile=/run/nginx.pid
```

```
ExecStartPre=/usr/sbin/nginx -t -c /etc/nginx/nginx.conf
```

```
ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf
```

```
ExecReload=/bin/kill -s HUP $MAINPID
```

```
ExecStop=/bin/kill -s QUIT $MAINPID
```

```
PrivateTmp=true
```

```
[Install]
```

```
WantedBy=multi-user.target
```

服务脚本

□ 案例1、编写源码安装系统服务启动脚本

分别基于CentOS6和7开发源码安装后系统服务脚本。

要求脚本具备启动、停止等基本功能。

RHEL7:

linux下出现nginx Failed to read PID from file /run/nginx.pid:

Invalid argument

systemd[1]: Failed to read PID from file /run/nginx.pid: Invalid argument

解决方法:

```
mkdir -p /etc/systemd/system/nginx.service.d
```

```
printf "[Service]\nExecStartPost=/bin/sleep 0.1\n" >
```

```
/etc/systemd/system/nginx.service.d/override.conf
```

然后:

```
systemctl daemon-reload
```

```
systemctl restart nginx.service
```

备份脚本

□ 1、数据库分库分表备份

分库备份:

```
#!/bin/bash
# define var
bak_user=root
bak_passwd=123456
bak_path=/server/dbbak
bak_cmd="-u$bak_user -p$bak_passwd"
exclude_db="Database|information_schema|mysql|performance_schema|test"
db_name=`mysql $bak_cmd -e "show databases;" | egrep -v
$exclude_db`

# main program
for db in `echo $db_name`
do
    [ -d $bak_path ] || mkdir -p $bak_path
    mysqldump $bak_cmd -B $db |gzip> $bak_path/${db}_`date
    +%Y%m%d`.sql.gz
done
```


备份脚本

❑ 1、数据库分库分表备份

分库备份:

```
# define var
bak_user=root
bak_passwd=123456
bak_path=/server/dbbak
bak_cmd="-u$bak_user -p$bak_passwd"
exclude_db="Database|information_schema|mysql|performance_schema|test"
db_name=`mysql $bak_cmd -e "show databases;" | egrep -v $exclude_db`

# main program
for db in `echo $db_name`
do
    [ -d ${bak_path}/${db} ] || mkdir -p ${bak_path}/${db}
    table_name=`mysql $bak_cmd -e "use $db;show tables;" | grep -v "Tables_in"`
    for table in `echo ${table_name}`
    do
        mysqldump $bak_cmd $db $table |gzip >
        $bak_path/${db}/${db}_${table}.`date +%Y%m%d`.sql.gz
    done
done
```


备份脚本

❑ 1、数据库分库分表备份

分表备份:

```
# define var
bak_user=root
bak_passwd=123456
bak_path=/server/dbbak
bak_cmd="-u$bak_user -p$bak_passwd"
exclude_db="Database|information_schema|mysql|performance_schema|test"
db_name=`mysql $bak_cmd -e "show databases;" | egrep -v $exclude_db`

# main program
for db in `echo $db_name`
do
    [ -d ${bak_path}/${db} ] || mkdir -p ${bak_path}/${db}
    table_name=`mysql $bak_cmd -e "use $db;show tables;" | grep -v
    "Tables_in"`
    for table in `echo ${table_name}`
    do
        mysqldump $bak_cmd $db $table |gzip >
        $bak_path/${db}/${db}_${table}.`date +%Y%m%d`.sql.gz
    done
done
```

备份脚本

□ 2、全网备份:

应用场景: 备份公司web服务器数据, 日志以及系统配置信息。

脚本说明: 本地使用tar备份, 备份完成时使用md5sum 生成标志以便备份服务器上检查备份是否成功, 备份结果用rsync推送到备份服务器(也可使用ftp方式上传至ftp服务器), 备份服务器检查备份是否成功并发送邮件通知管理员。

备份本地保留一周, 服务器保留一月数据。

分析: 需要备份内容 (1-4为配置文件, 5-6为web服务器数据及日志)

- 1、定时任务服务的配置文件/var/spool/cron/root
- 2、开机自启动配置文件/etc/rc.local
- 3、日常脚本的目录
- 4、防火墙iptables的配置文件/etc/sysconfig/iptables
- 5、web服务器数据, 假定为/var/www/html
- 6、日志, 假定为/var/log/httpd

首先, 配置好rsync服务器和客户端, 并测试可以使用;

其次, 本地tar打包备份;

再次, 使用rsync推送到服务器;

最后, 服务器端检查并邮件告警;

测试各个阶段都没问题, 设置定时任务。

备份脚本

□ 3、基于Innobackupex的MySQL备份脚本 具体要求：

- 1、周日全备
- 2、周一至周六增量备份
- 3、备份使用backup用户

grant SELECT,RELOAD,SHOW DATABASES,LOCK
TABLES,SUPER,REPLICATION CLIENT on *.* to
backup@'localhost' identified by '123456';

- 4、结合计划任务，备份时间为每天02:00:00

监控脚本

□ Web服务器监控

应用场景：监控web服务器状态，异常时邮件报警。

脚本说明：通过wget（也可以用curl）监控服务器状态，如果不能正常访问，ping检测网络，网络正常通知管理员检查服务，ping不通邮件通知管理员。

服务器列表使用数组，服务器状态函数使用返回值判断服务器是否异常。

安全脚本

自动化禁止恶意IP访问

应用场景：防止恶意IP尝试ssh登录。

脚本说明：将密码输入错误超过4次的IP地址通过iptables防火墙阻止访问。

分析：

1) 首先，需要知道ssh远程访问记录在哪个文件中/var/log/secure

2) 其次，模拟远程访问输错密码，查看日志文件

Dec 26 11:34:53 agent1 sshd[3060]: Failed password for root
from 192.168.211.1 port 2075 ssh2

3) 再次，通过日志可以看到关键信息“Failed password”表示密码错误

有可能是手误输错，所以需要设定几次错误为恶意试探密码，建议设置为4

另一个关键信息是需要将密码错误的IP地址提取出来
对提取出来的IP地址进行统计次数

4) 最后，需要明确怎么在脚本中通过iptables策略设置阻止恶意IP访问
策略添加到哪里合适
防火墙配置文件等

总结

- 服务脚本
- 备份脚本
- 监控脚本
- 安全脚本

作业

- 1、源码安装nginx，编写系统服务脚本。
- 2、写一个脚本判断你的Linux服务器里是否开启web服务，如果开启了请判断跑的是什么服务，是httpd还是nginx又或是其他？
- 3、写一个shell脚本，通过curl -I返回的状态码来判断所访问的网站是否正常，比如:当状态码为200|301|302时，才算正常



谢谢观看

更多好课，请关注[万门大学APP](#)

