

项目一

NSD PROJECT1

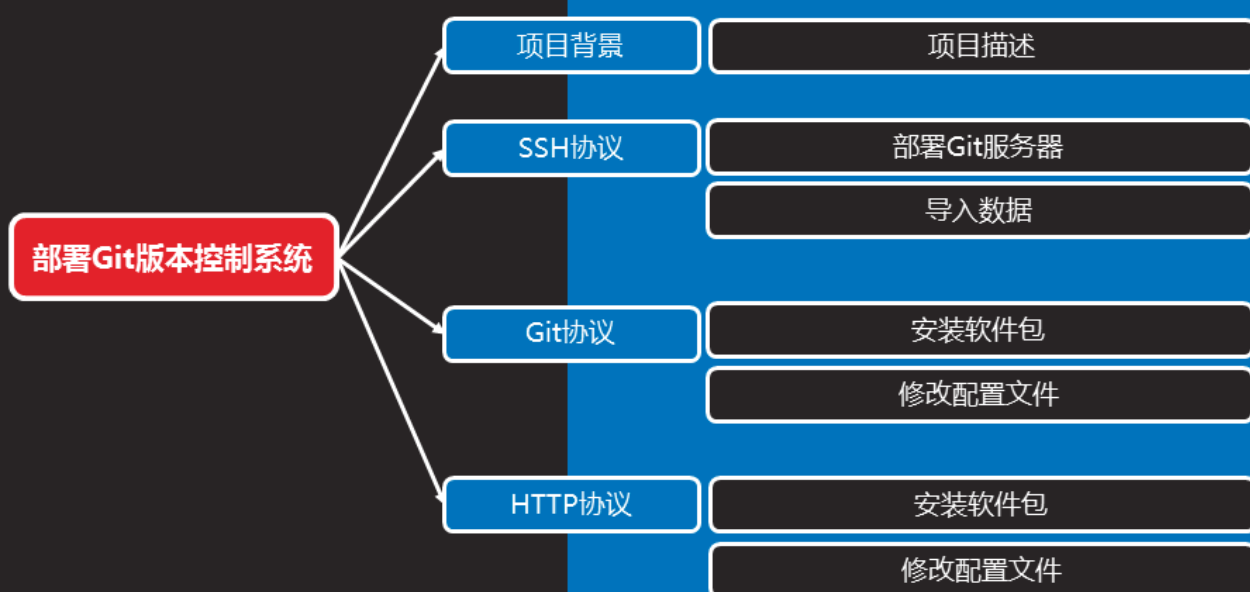
DAY04

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	部署Git版本控制系统
	10:30 ~ 11:20	
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	优化Web服务器
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



部署Git版本控制系统



项目背景

项目描述

知识讲解

我们的代码目前没有使用版本控制系统，后期迭代更新版本多了之后，对比代码的管理会是一种挑战。



小锤

为了更好地进行代码版本控制，我们需要将现有所有代码迁移到版本控制系统，你看看多久可以完成这个任务。



大宇

版本控制系统还是比较容易的，差不多一两天就可以完成代表的迁移。



项目描述（续1）

知识讲解



大字

目前主流的版本控制系统有：SVN、Git。
选择哪一款软件呢？



项目描述（续2）

知识讲解

产品	优点	缺点
SVN	逻辑结构简单 采用一对多的集中式管理方式 目录级别权限控制	必须联网 分支不灵活 不支持分布式
Git	支持分布式 速度快、可以灵活创建分支 可以离线工作 解决冲突容易	暂时没想到



SSH协议

部署Git服务器

- 安装git软件包

```
[root@database ~]# yum -y install git
```

- 创建空仓库

```
[root@database ~]# mkdir /var/git/
```

```
[root@database ~]# git init --bare /var/git/wordpress.git #创建空仓库
```

导入数据

知识讲解

- 导入数据

```
[root@web1 var]# git config --global push.default simple
[root@web1 var]# git config --global user.email you@example.com
[root@web1 var]# git config --global user.name "Your Name"

[root@web1 var]# cd /var/
[root@web1 var]# git clone root@192.168.2.21:/var/git/wordpress.git
[root@web1 var]# cd /var/wordpress
[root@web1 wordpress]# cp -a /usr/local/nginx/html/* ./

[root@web1 wordpress]# git add .
[root@web1 wordpress]# git commit -m "wordpress code"
[root@web1 wordpress]# git push
```



Git协议

安装软件包

- 安装git-daemon

```
[root@database ~]# yum -y install git-daemon
```

知识讲解



修改配置文件

- 修改git@.service配置

```
[root@database ~]# vim /usr/lib/systemd/system/git@.service
```

修改前内容如下：

```
ExecStart=-/usr/libexec/git-core/git-daemon --base-path=/var/lib/git --export-all --user-path=public_git --syslog --inetd -verbose
```

修改后内容如下：

```
ExecStart=-/usr/libexec/git-core/git-daemon --base-path=/var/git --export-all -  
-user-path=public_git --syslog --inetd -verbose
```

```
[root@database ~]# systemctl start git.socket
```

```
[root@database ~]# systemctl status git.socket
```

知识讲解



HTTP协议

安装软件包

- 安装git-daemon

```
[root@database ~]# yum -y install httpd gitweb
```


修改配置文件

知识讲解

- 修改gitweb.conf配置

```
[root@database ~]# vim /etc/gitweb.conf  
$projectroot = "/var/git";
```

#添加一行

- 启动服务

```
[root@database ~]# systemctl start httpd
```



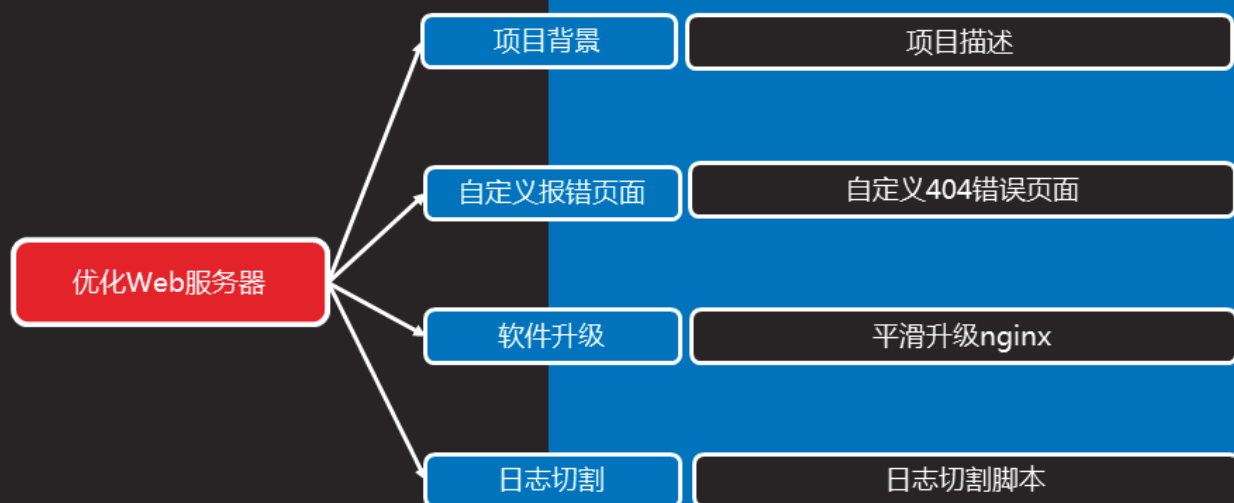
案例1：部署Git版本控制系统

课堂练习

- 部署Git版本控制系统，管理网站代码，实现如下效果：
 - 基于SSH协议的服务器
 - 基于Git协议的服务器
 - 基于HTTP协议的服务器
 - 上传代码到版本仓库



优化Web服务器



项目背景

项目描述

知识讲解

现在我们的博客系统已经非常健壮了，
但还是有进一步优化的空间。



小锤

大字，你看看目前我们这套系统还可以
做哪些优化工作？



大字

好的，我思考思考。



项目描述（续1）

知识讲解

优化分为性能优化、功能优化、用户体验优化。

对，就从这三个方面入手吧。



大字



自定义报错页面

自定义404错误页面

- 修改配置 (以web1为例)

```
[root@web1 ~]# vim /usr/local/nginx/conf/nginx.conf
error_page 404 /404.html; //自定义错误页面
```

知识讲解

- 重启服务 (以web1为例)

```
[root@web1 ~]# /usr/local/nginx/sbin/nginx -s reload
```



软件升级

平滑升级nginx

- 升级nginx

知识讲解

```
[root@web1 ~]# tar -xf nginx-1.15.8.tar.gz
[root@web1 ~]# cd nginx-1.15.8
[root@web1 ~]# ./configure \
--with-http_ssl_module \
--with-http_stub_status_module
[root@web1 ~]# make
[root@web1 ~]# mv /usr/local/nginx/sbin/nginx{,.bak}
[root@web1 ~]# cp objs/nginx /usr/local/nginx/sbin/
```



平滑升级nginx (续1)

知识讲解

- 开启status功能

```
[root@web1 ~]# vim /usr/local/nginx/conf/nginx.conf
```

```
... ..
```

```
location /status {  
    stub_status on;  
    allow 192.168.2.0/24;  
    deny all;  
}
```

```
... ..
```

- 重启服务器

```
[root@web1 ~]# killall nginx
```

```
[root@web1 ~]# /usr/local/nginx/sbin/nginx
```



日志切割

日志切割脚本

知识讲解

- 编写脚本

```
[root@web1 ~]# vim /usr/local/nginx/logbak.sh
#!/bin/bash
date=`date +%Y%m%d`
logpath=/usr/local/nginx/logs
mv $logpath/access.log $logpath/access-$date.log
mv $logpath/error.log $logpath/error-$date.log
kill -USR1 $(cat $logpath/nginx.pid)
```

- 编写计划任务

```
[root@web1 ~]# crontab -e
03 03 * * 5 /usr/local/nginx/logbak.sh
```



案例2：优化Web服务器

课堂练习

- 优化Web服务器，实现如下效果：
 - 自定义网站404错误页面
 - 升级nginx至1.15.8版本，开启status模块
 - 编写日志切割脚本，实现每周五备份日志



总结和答疑
