

NSD Project1 DAY04

1. [案例1：部署Git版本控制系统](#)
2. [案例2：优化Web服务器](#)

1 案例1：部署Git版本控制系统

1.1 问题

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

1.2 方案

生产环境应该有一台独立的Git服务器，这里为了节约主机资源，我们使用数据库主机同时做完Git服务器，如图-1所示。

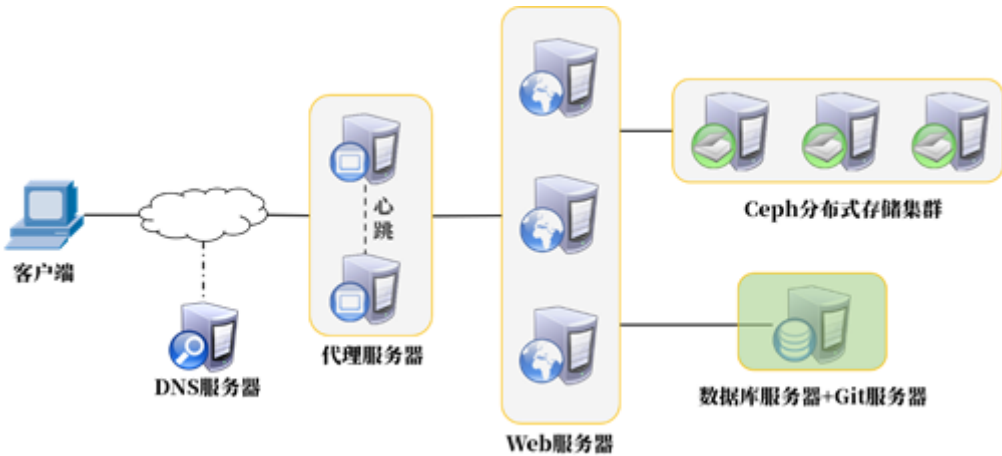


图-1

主机配置如表-1所示。

表-1

主机角色	主机名称	IP 地址
数据库服务器	database	eth1(192.168.2.21/24)

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署SSH协议的版本控制服务器

- 1) 安装软件包，创建空仓库。

[Top](#)

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

```
02. [root@database ~]# git init --bare /var/git/wordpress.git
```

2) 登陆web1服务器克隆git仓库，上传网站代码到git服务器。

```
01. [root@web1 var]# git config --global push.default simple
02. [root@web1 var]# git config --global user.email you@example.com
03. [root@web1 var]# git config --global user.name "Your Name"
04.
05. [root@web1 var]# cd /var/
06. [root@web1 var]# git clone root@192.168.2.21:/var/git/wordpress.git
07. [root@web1 var]# cd /var/wordpress
08. [root@web1 wordpress]# cp -a /usr/local/nginx/html/* ./
09.
10. [root@web1 wordpress]# git add .
11. [root@web1 wordpress]# git commit -m "wordpress code"
12. [root@web1 wordpress]# git push
13. root@192.168.2.21's password:<输入192.168.2.21主机root的密码>
```

步骤二：部署Git协议的版本控制服务器

1) 安装软件包（192.168.2.21操作）

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

2) 修改配置文件，启动Git服务

```
01. [root@database ~]# vim /usr/lib/systemd/system/git@.service
02. 修改前内容如下：
03. ExecStart=-/usr/libexec/git-core/git-daemon --base-path=/var/lib/git
04. 修改后内容如下：
05. ExecStart=-/usr/libexec/git-core/git-daemon --base-path=/var/git --ex
06.
07. [root@database ~]# systemctl start git.socket
08. [root@database ~]# systemctl status git.socket
```

[Top](#)

3)客户端测试（使用web2做完客户端主机，192.168.2.12）

在web2执行clone等同于是把代码又备份了一份。

```
01. [root@web2 ~]# cd /var/  
02. [root@web2 var]# git clone git://192.168.2.21/wordpress.git
```

步骤三：部署HTTP协议的版本控制服务器

1) 安装软件包（192.168.2.21操作）

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

2) 修改配置文件

```
01. [root@database ~]# vim /etc/gitweb.conf  
02. $projectroot = "/var/git"; #添加一行
```

3) 启动服务

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

4) 客户端验证

```
01. [root@room9pc01 ~]# firefox http://192.168.2.21/git
```

访问网页可以查看到wordpress仓库，点击tree菜单后可以看到如图-2所示的代码。

[Top](#)



图-2

2 案例2：优化Web服务器

2.1 问题

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

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：自定义404错误页面

1) 优化前测试（客户端访问一个不存在的页面）。

```
01. [root@room9pc01 ~]# firefox http://www.lab.com/git
```

2) 修改Nginx配置文件，自定义错误页面

```
01. [root@web1 ~]# vim /usr/local/nginx/conf/nginx.conf
02. error_page 404 /404.html; //自定义错误页面
03. [root@web2 ~]# vim /usr/local/nginx/conf/nginx.conf
04. error_page 404 /404.html; //自定义错误页面
05. [root@web3 ~]# vim /usr/local/nginx/conf/nginx.conf
06. error_page 404 /404.html; //自定义错误页面
```

[Top](#)

3) 重启nginx

```
01. [root@web1 ~]# /usr/local/nginx/sbin/nginx -s reload
02. [root@web2 ~]# /usr/local/nginx/sbin/nginx -s reload
03. [root@web3 ~]# /usr/local/nginx/sbin/nginx -s reload
```

步骤二：升级nginx版本，开启status模块

1) 配置、编译新的nginx (web1、web2、web3做相同操作，下面以web1为例)

```
01. [root@web1 ~]# tar -xf nginx-1.15.8.tar.gz
02. [root@web1 ~]# cd nginx-1.15.8
03. [root@web1 ~]# ./configure \
04. --with-http_ssl_module \
05. --with-http_stub_status_module
06. [root@web1 ~]# make
```

2) 备份老版本nginx，更新新版本nginx

```
01. [root@web1 ~]# mv /usr/local/nginx/sbin/nginx{,.bak}
02. [root@web1 ~]# cp objs/nginx /usr/local/nginx/sbin/
```

3) 修改配置文件

```
01. [root@web1 ~]# vim /usr/local/nginx/conf/nginx.conf
02. ... ..
03. location /status {
04.             stub_status on;
05.             allow 192.168.2.0/24;
06.             deny all;
07.         }
08. ... ..
```

4) 重启服务

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

[Top](#)

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

步骤三：编写日志切割脚本

1) 编写脚本（以web1为例）

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

2) 创建计划任务

```
01. [root@web1 ~]# crontab -e
02. 03 03 * * 5 /usr/local/nginx/logbak.sh
```

[Top](#)