operation day 06

================================================

**为b网站虚拟主机添加日志**

**tomcat只为默认的虚拟主机添加了日志功能，如果新建的虚拟主机需要该功能就**

**按下列方式配置：**

打开配置文件，在b网站的配置中添加日志功能<Valve ..... />(该配置从默认的

虚拟主机中复制即可，最后一页)，修改prefix日志名字，suffix日志后缀

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"

prefix=" web\_b\_access\_log" suffix=".log"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>

之后保存退出，重启tomcat服务，然后访问几次b网站

ls logs //查看tomcat目录下的日志目录，可以看到b网站的日志

------------------------------------------------------------------------------

**使用Varnish加速Web**

**有时网站服务器搭建好了客户会因为距离较远而访问效果不好，出现这种情况就可以用varnish工具在距离客户比较近的地区搭建缓存服务器，然后客户访问缓存服务器即可，缓存服务器会从原始站点获得数据并缓存，随着被访问与被缓存的数据越来越多，客户的访问速度就可以加快了，但由于这种方式不是一般企业可以做到，通常有需求时可以去购买CDN （内容分发网络）服务。**

**以下实验仅用来感受该服务效果，了解CDN服务的理念即可**

在web1主机开启httpd服务

yum -y install httpd

killall nginx

systemctl start httpd

systemctl stop firewalld

在proxy主机测试

[root@proxy ~]# curl 192.168.2.100

[root@proxy ~]#cd /root/lnmp\_soft

[root@proxy lnmp\_soft]# tar -xf varnish-5.2.1.tar.gz

[root@proxy lnmp\_soft]# cd varnish-5.2.1/

yum -y install gcc readline-devel pcre-devel python-docutils

./configure

make

make install

useradd varnish -s /sbin/nologin //创建varnish所需用户

cp etc/example.vcl /usr/local/etc/default.vcl //拷贝配置文件

vim /usr/local/etc/default.vcl //修改配置文件

backend default {

.host = "192.168.2.100"; //原始站点服务器的ip

.port = "80"; //原始站点服务器的端口号

}

killall nginx //如果nginx还在运行就关掉

varnishd -f /usr/local/etc/default.vcl //开启varnish服务

[root@proxy varnish-5.2.1]# curl 192.168.2.5 //测试

httpd-web1~~ //看到web1的页面则成功

**---------------------------------------------------------------**

**程序(文件或其他数据)的版本管理工具**

**svn 集中式 使用该服务时，客户端要与服务器保持**

**在线状态，数据统一保存在服务器端**

**git 分布式 使用该服务时，仅仅传输数据时需要在线**

**，数据保存在git服务器与git客户端**

配置git服务端，这里使用web1

[root@web1 ~]# yum -y install git

[root@web1 ~]# mkdir /var/lib/git //创建git存储数据的目录

[root@web1 ~]# git init --bare /var/lib/git/project //创建仓库，仓

库是用来存储数据的目录

配置git客户端，这里使用web2

[root@web2 ~]# yum -y install git

[root@web2 ~]# git clone 192.168.2.100:/var/lib/git/project //克隆

服务端的仓库

[root@web2 ~]# cd project //进入仓库

[root@web2 project]# echo "web2-01" > web2-01.txt //创建测试文件