# 实验报告：搭建Haproxy + Keepalived + Mariadb主从复制 + NFS集群

实验准备：

VIP:192.168.7.254

ha1:192.168.7.10 vmnet8

ha2:192.168.7.11 vmnet8

web1: 192.168.7.12 vmnet8

192.168.10.12 vmnet1

web2: 192.168.7.13 vmnet8

192.168.10.13 vmnet1

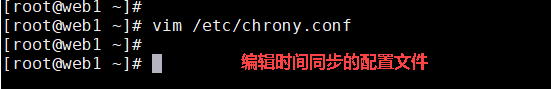
web3:192.168.7.14 vmnet8

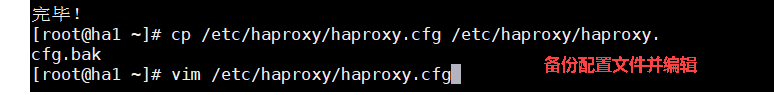
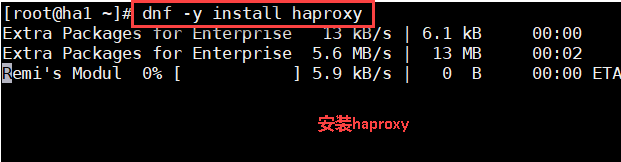
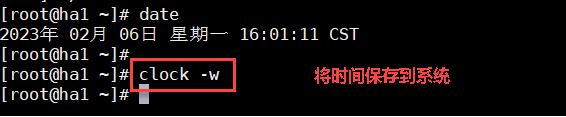
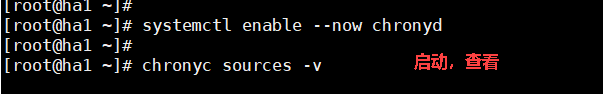
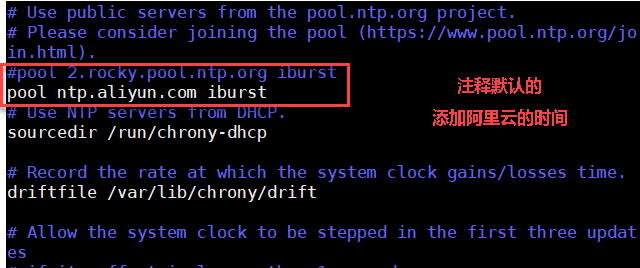
192.168.10.14 vmnet1

nfs: 192.168.10.18 vmnet1

192.168.7.18 vmnet8

Ha1和ha2同时操作：





haproxy做负载均衡，属于7层负载均衡

Haproxy配置文件通常分为哪几个部分？其作用分别是什么？

haproxy 的配置文件由两部分组成：全局设定和对代理的设定，共分为五段：global，defaults，frontend，backend，listen。

global：参数是进程级的，通常和操作系统（OS）相关。这些参数一般只设置一次，如果配置无误，就不需要再次配置进行修改

defaults：配置默认参数的，这些参数可以被利用配置到frontend，backend，listen组件

frontend：接收请求的前端虚拟节点，Frontend可以根据规则直接指定具体使用后端的      backend(可动态选择)。

backend：后端服务集群的配置，是真实的服务器，一个Backend对应一个或者多个实体服务器。

listen：Frontend和Backend的组合体。

配置文件内容如下：

global #全局属性

daemon #以daemon方式在后台运行

maxconn 256 #最大同时256连接

pidfile /root/haproxy/conf/haproxy.pid #指定保存HAProxy进程号的文件

defaults #默认参数

mode http #http模式

timeout connect 5000ms #连接server端超时5s

timeout client 50000ms #客户端响应超时50s

timeout server 50000ms #server端响应超时50s

frontend http-in #前端服务http-in

bind 192.168.6.128:9090 #监听9090端口

default\_backend servers #请求转发至名为"servers"的后端服务

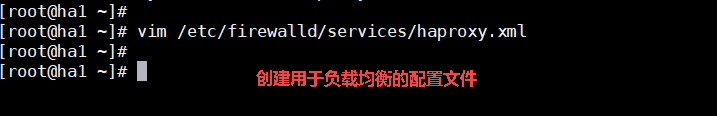
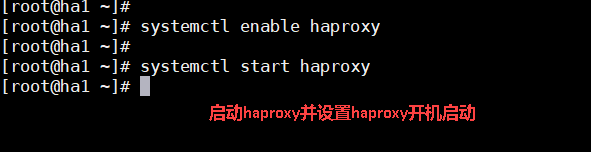
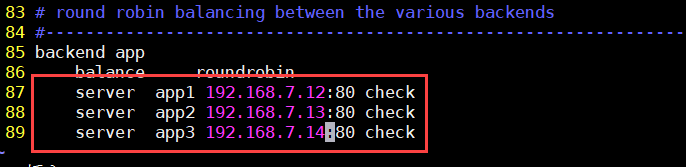
backend servers #后端服务servers

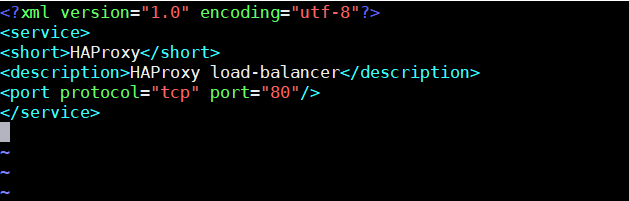
balance roundrobin #使用RR负载均衡算法

server server1 192.168.6.128:80 maxconn 32 #backend servers中只有一个后端服务，名字叫server1，启在本机的8000端口， HAProxy同时最多向这个服务发起32个连接

server server2 192.168.6.129:80 maxconn 32

server server3 192.168.6.130:80 maxconn 32





**<?xml version="1.0" encoding="utf-8"?>**

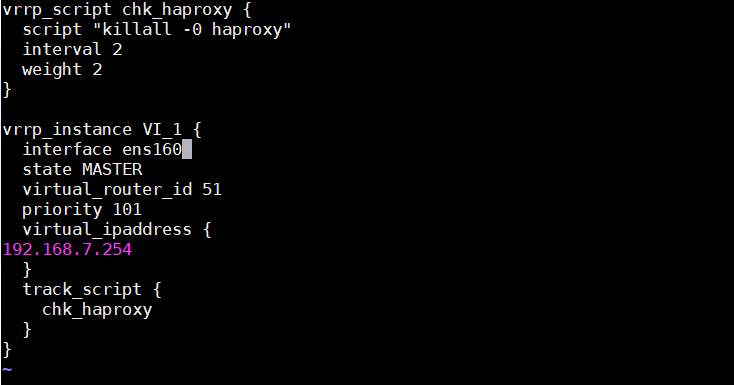
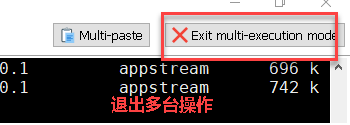
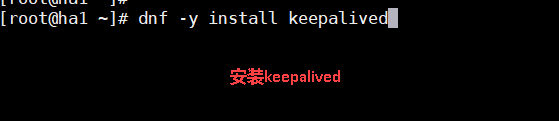
**<service>**

**<short>HAProxy</short>**

**<description>HAProxy load-balancer</description>**

**<port protocol="tcp" port="80"/>**

**</service>**



**vrrp\_script chk\_haproxy {**

**script "killall -0 haproxy" # check the haproxy process**

**interval 2 # every 2 seconds**

**weight 2 # add 2 points if OK**

**}**

**vrrp\_instance VI\_1 {**

**interface ens160 # interface to monitor**

**state MASTER # MASTER on haproxy1, BACKUP on haproxy2**

**virtual\_router\_id 51**

**priority 101 # 101 on haproxy1, 100 on haproxy2**

**virtual\_ipaddress {**

**192.168.7.254 # virtual ip address**

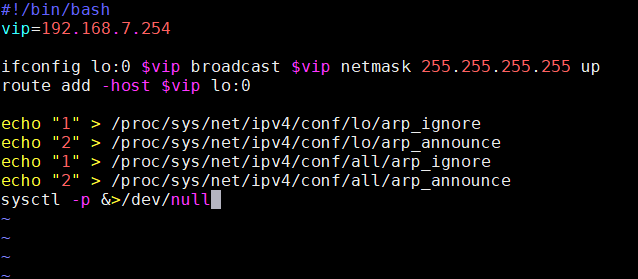
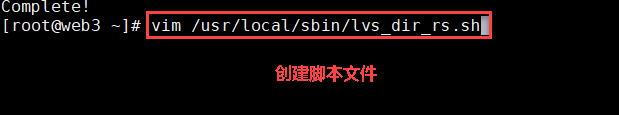
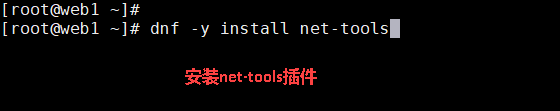
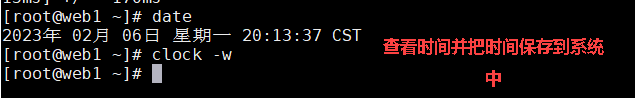
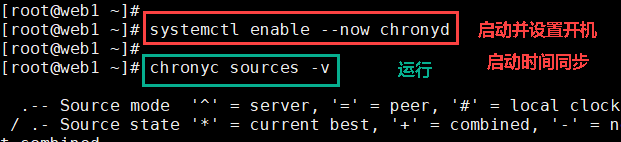
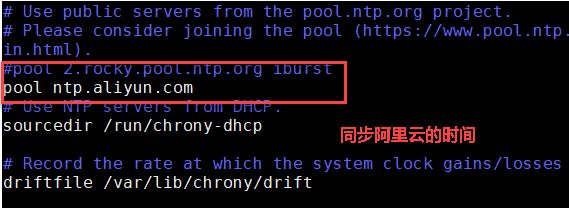
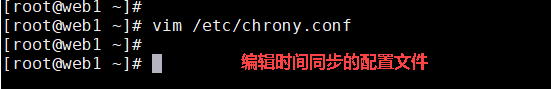
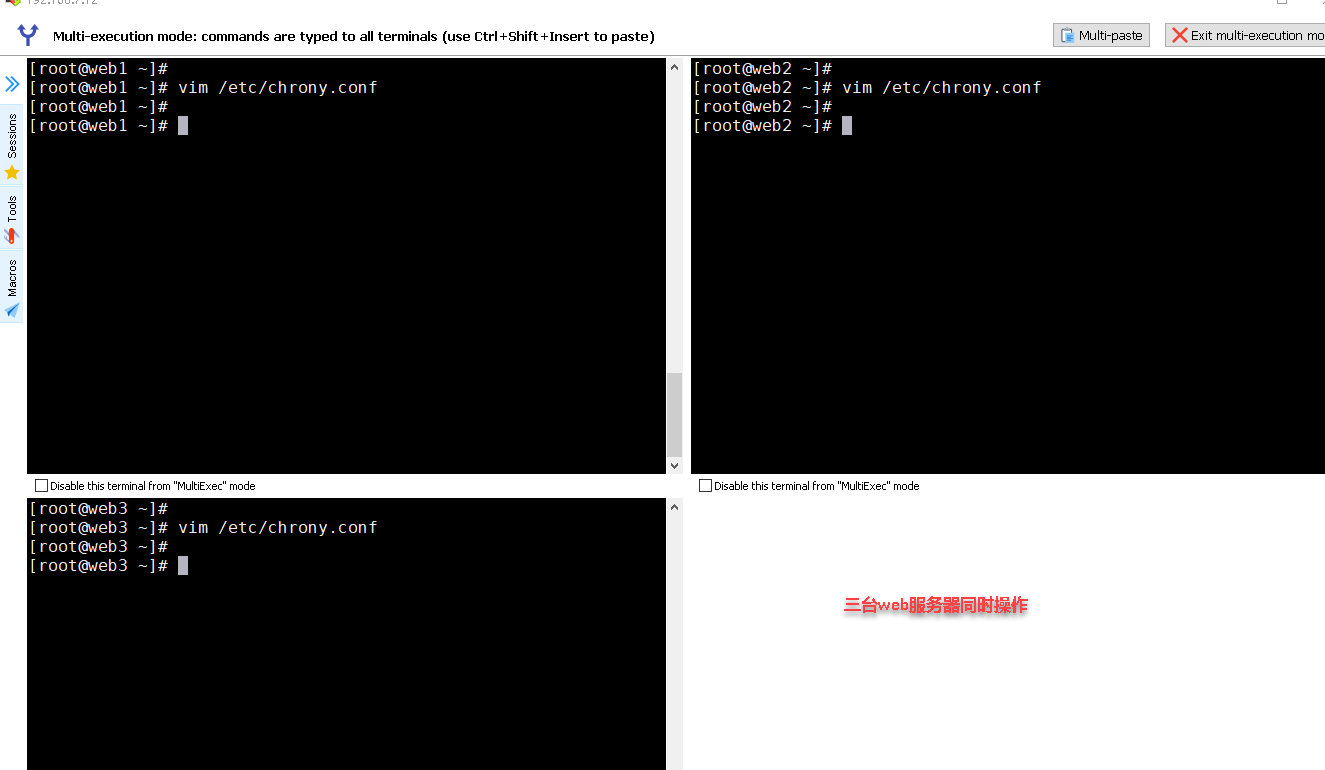
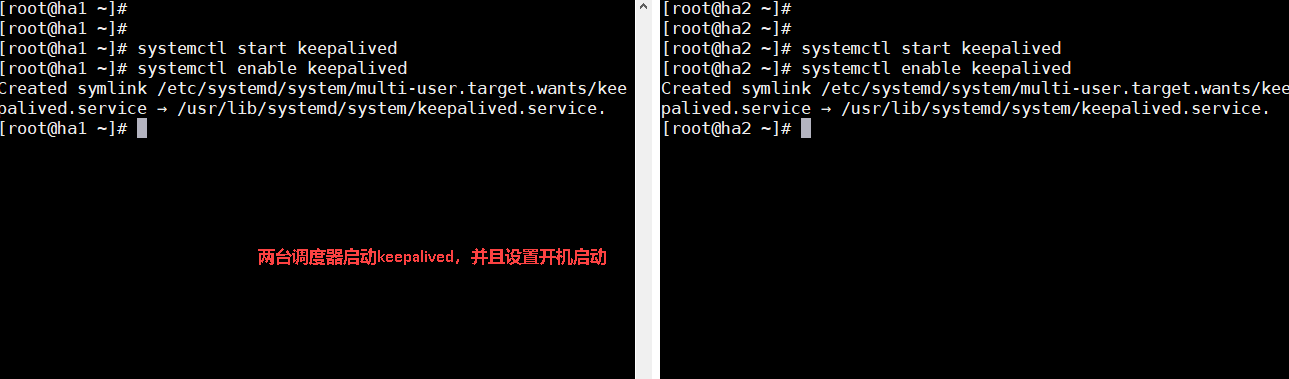
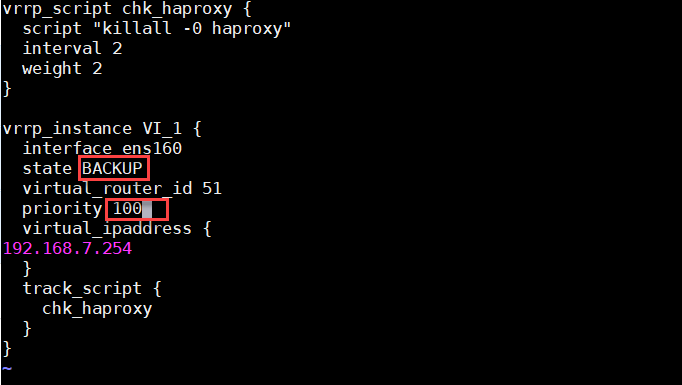
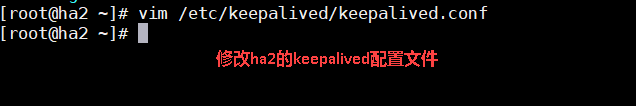
**}**

**track\_script {**

**chk\_haproxy**

**}**

**}**



**#!/bin/bash**

**vip=192.168.7.254**

**ifconfig lo:0 $vip broadcast $vip netmask 255.255.255.255 up**

**route add -host $vip lo:0**

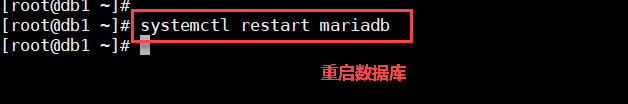
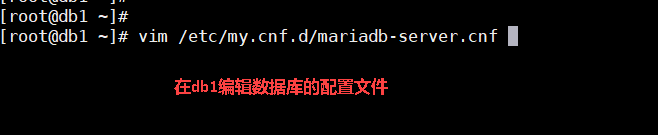
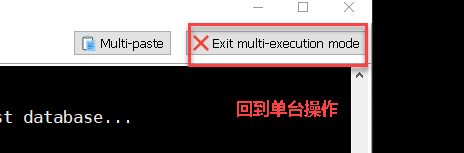
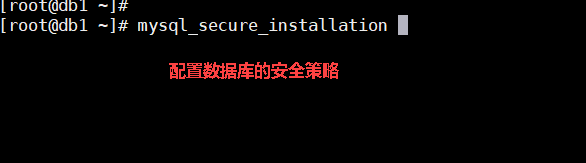
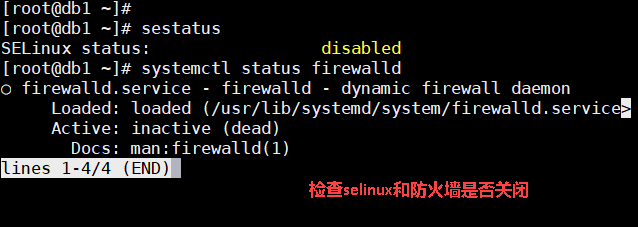
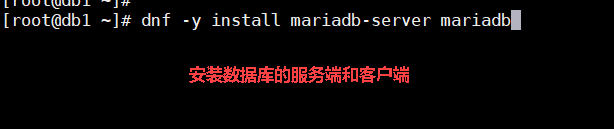
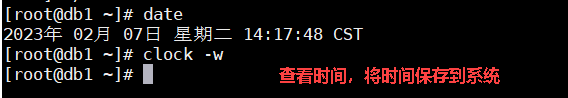
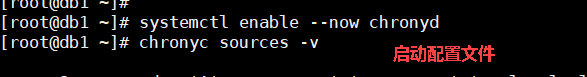
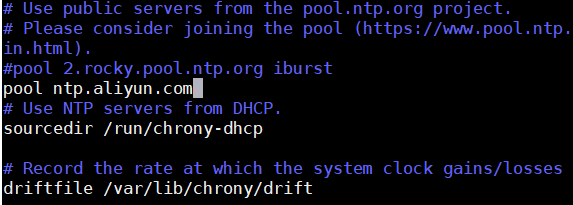
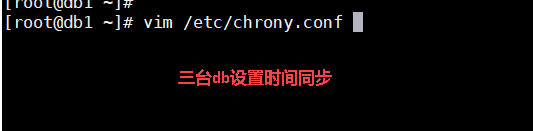
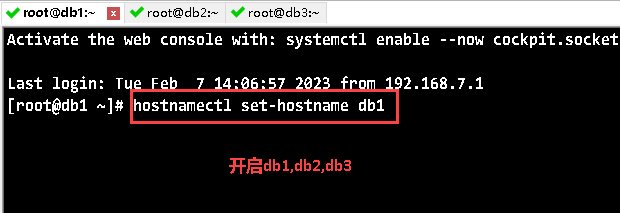
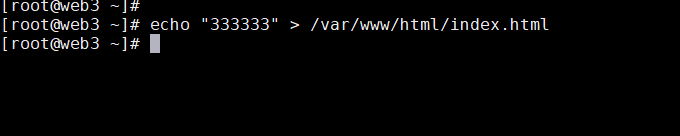
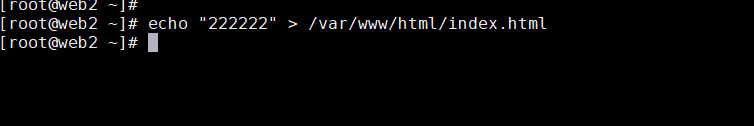
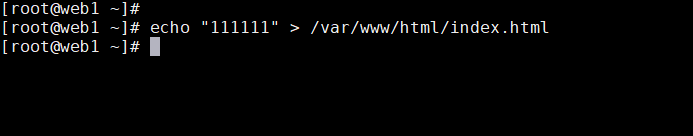
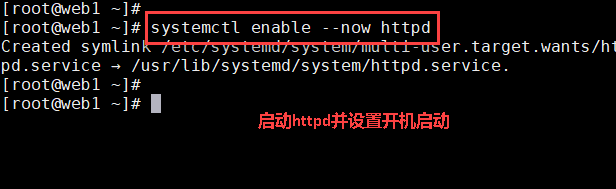
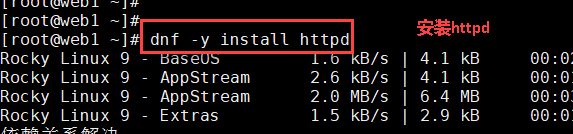
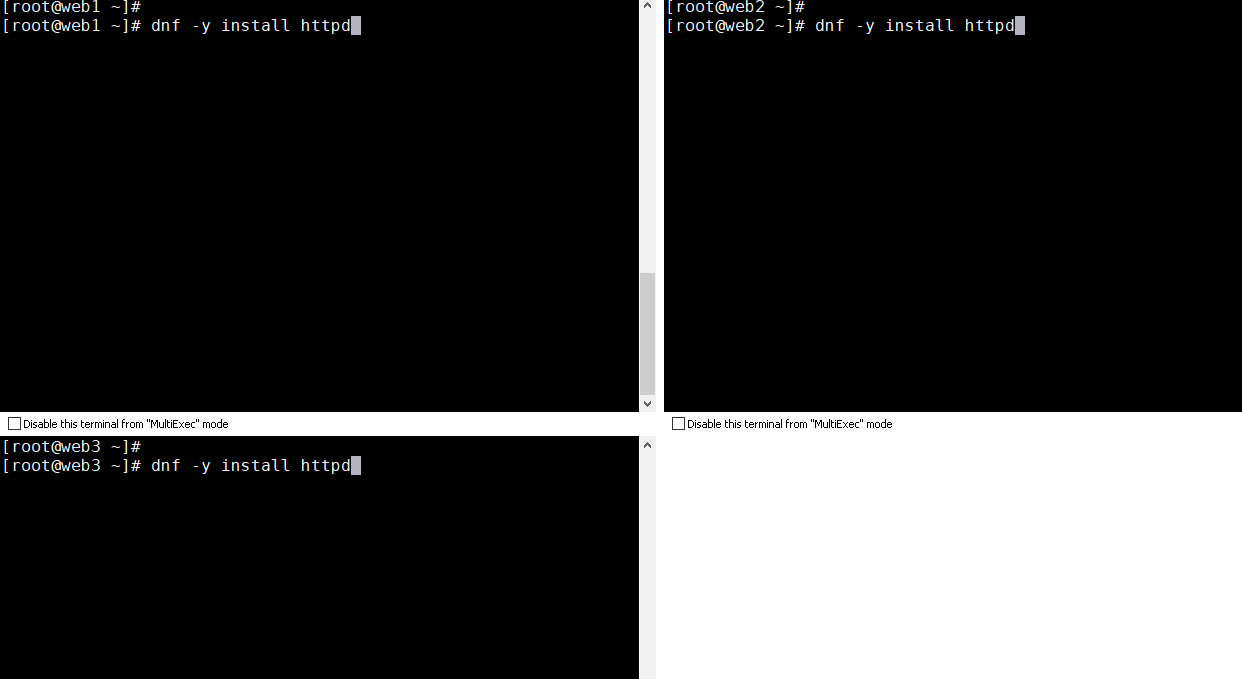
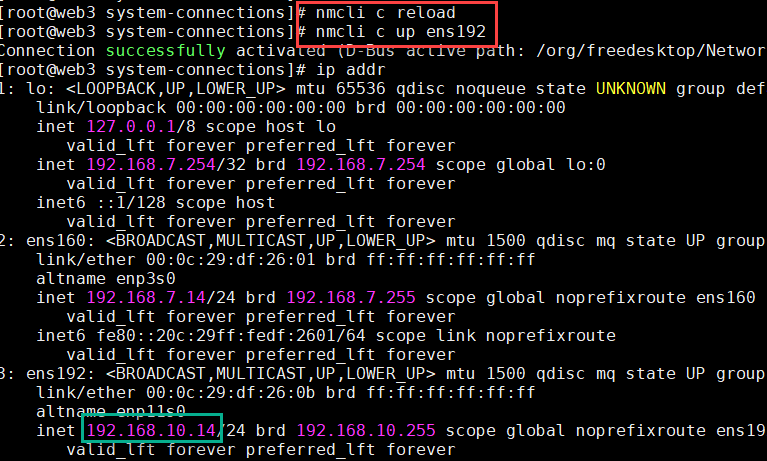
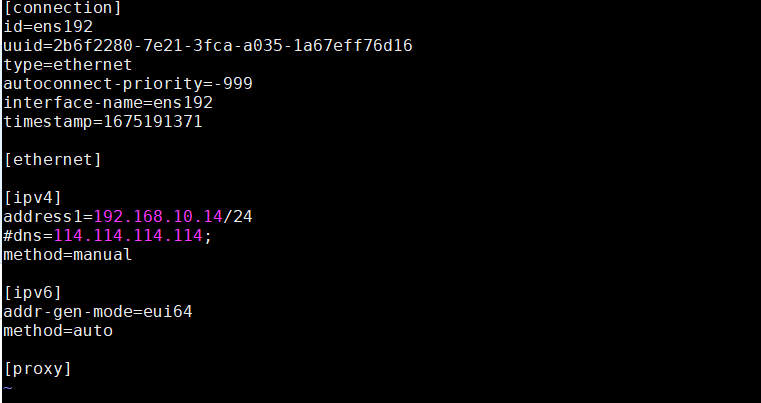
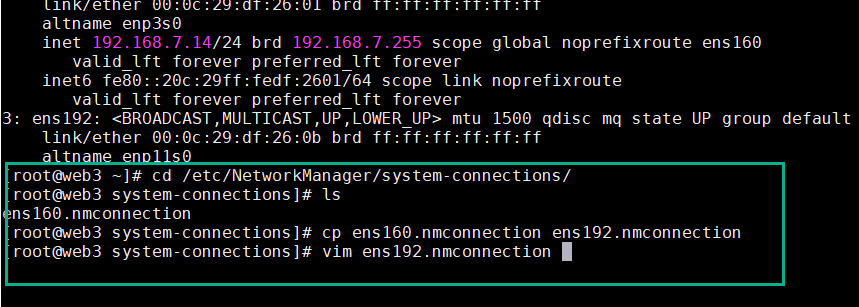
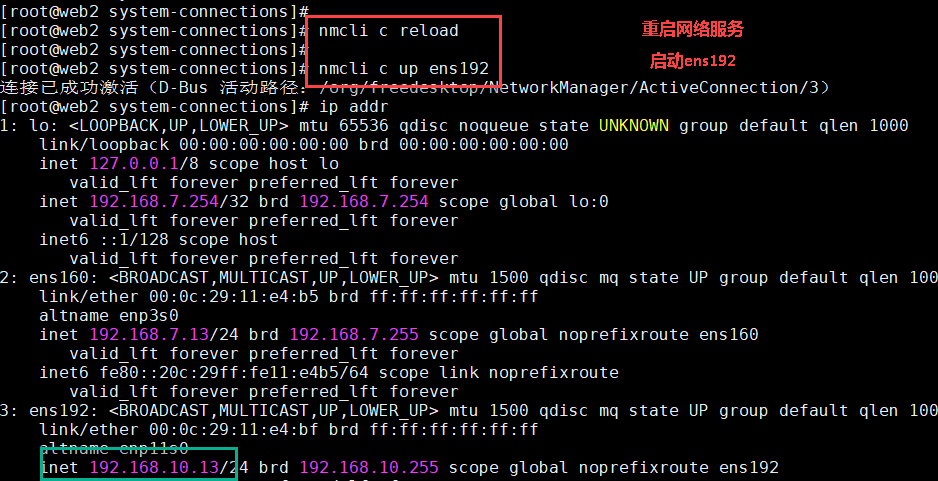
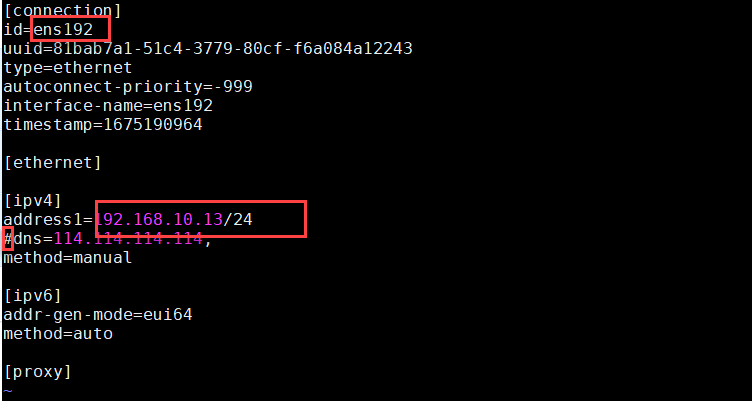
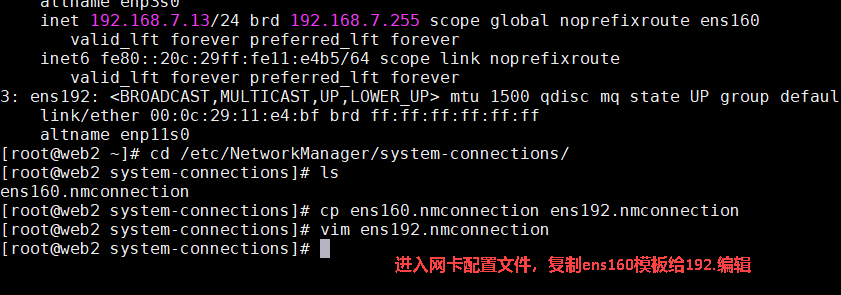
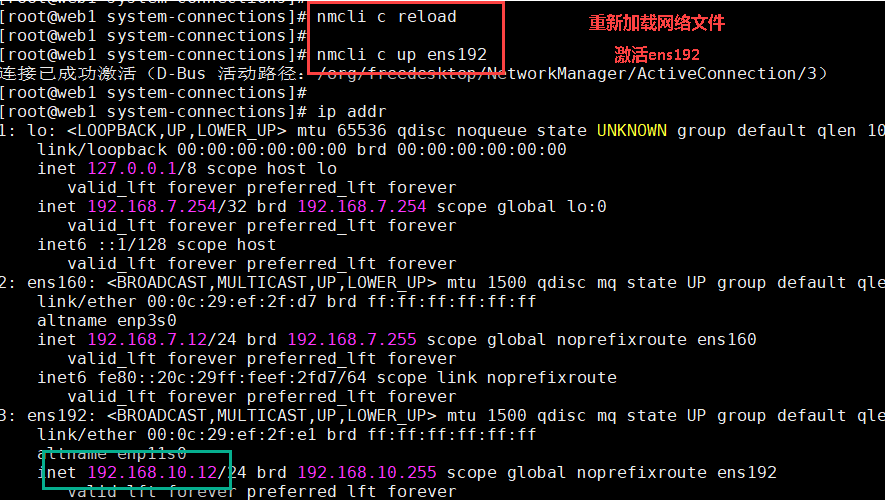
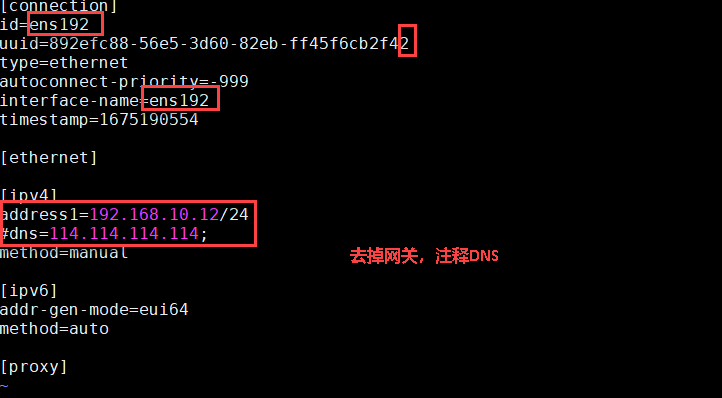
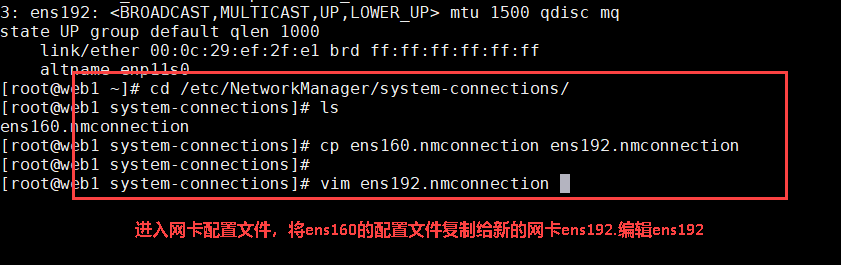
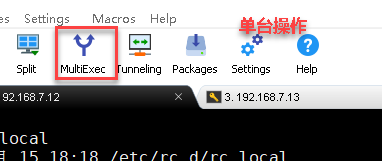
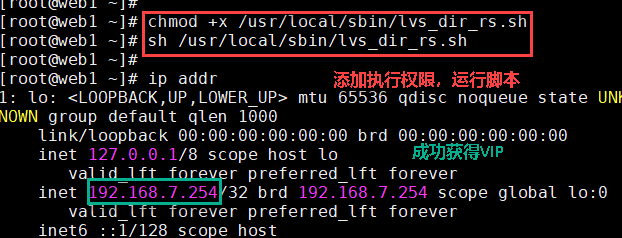
**echo "1" > /proc/sys/net/ipv4/conf/lo/arp\_ignore**

**echo "2" > /proc/sys/net/ipv4/conf/lo/arp\_announce**

**echo "1" > /proc/sys/net/ipv4/conf/all/arp\_ignore**

**echo "2" > /proc/sys/net/ipv4/conf/all/arp\_announce**

**sysctl -p &>/dev/null**



MariaDB [(none)]> SHOW MASTER STATUS;

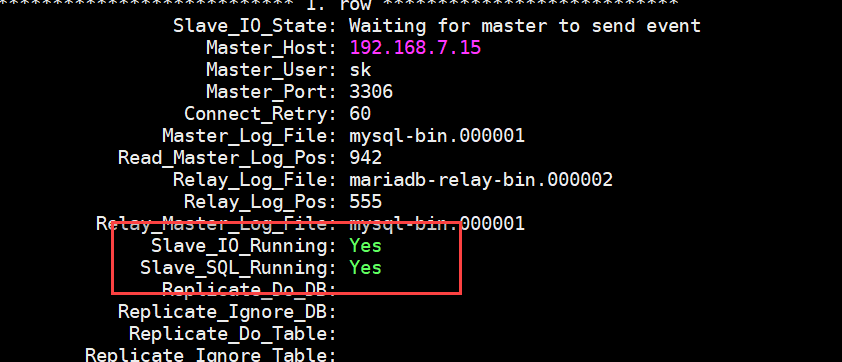
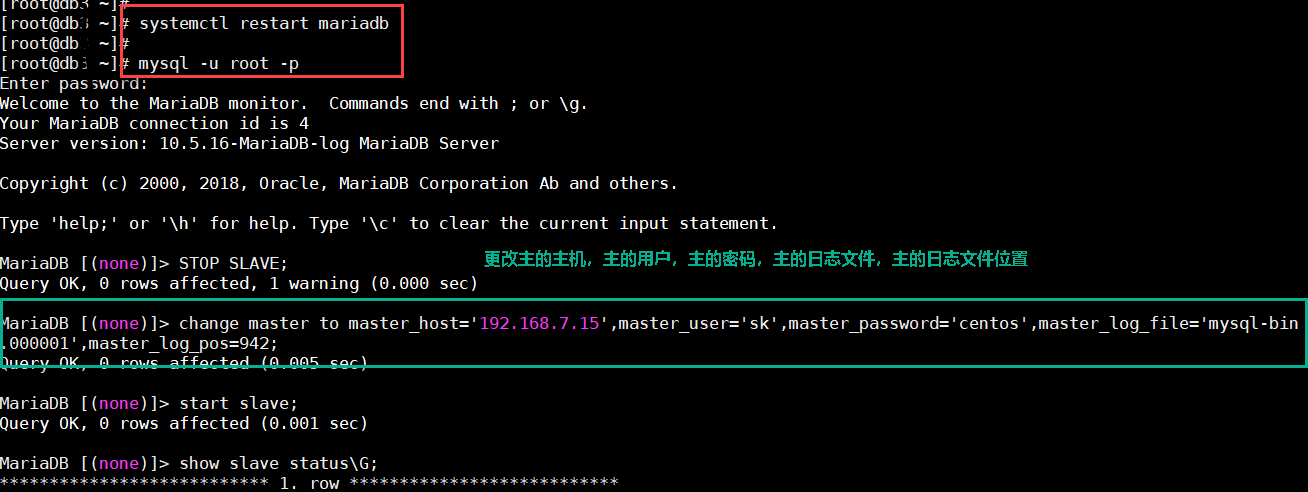
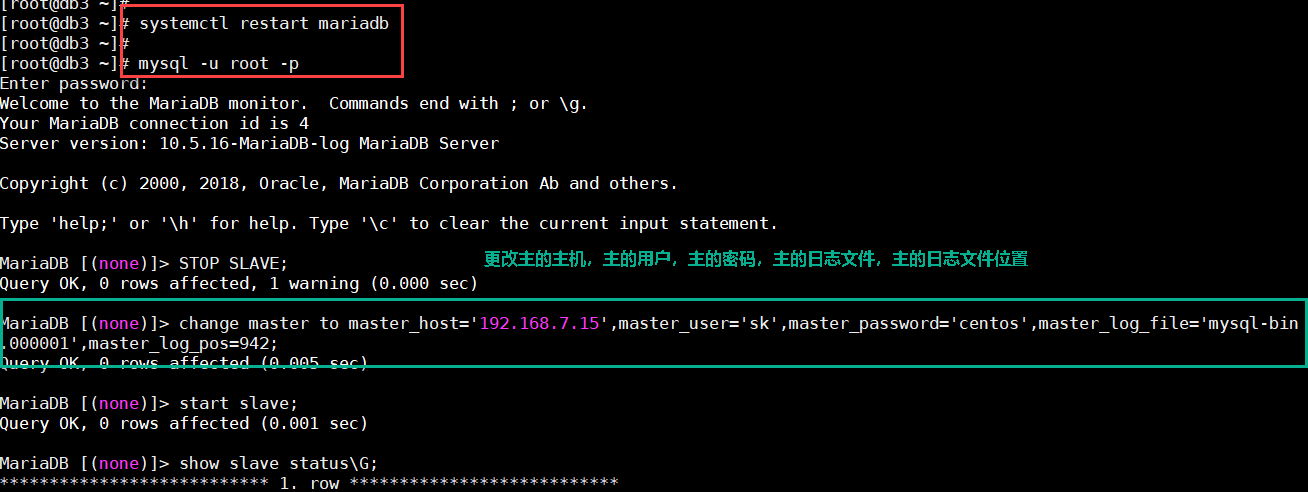
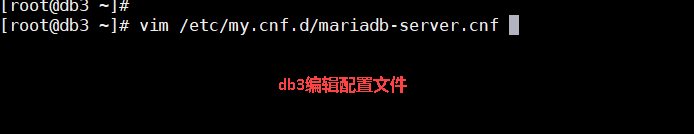
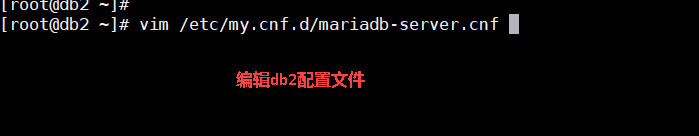
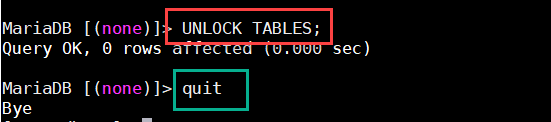
+------------------+----------+--------------+------------------+

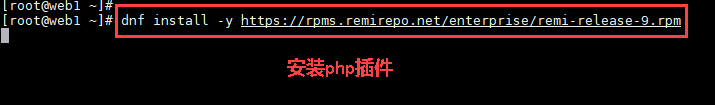
| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+------------------+----------+--------------+------------------+

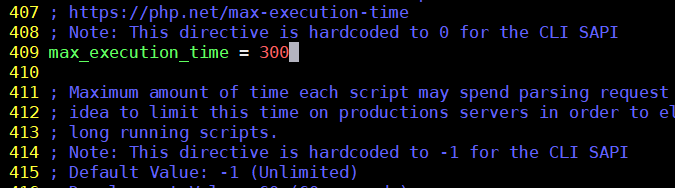
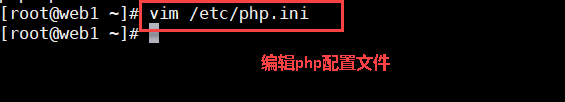
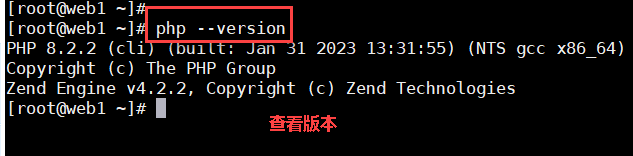
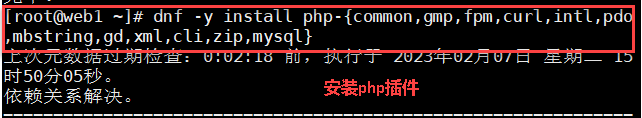
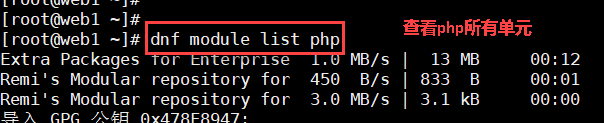
| mysql-bin.000001 | 514 | | |

+------------------+----------+--------------+------------------+





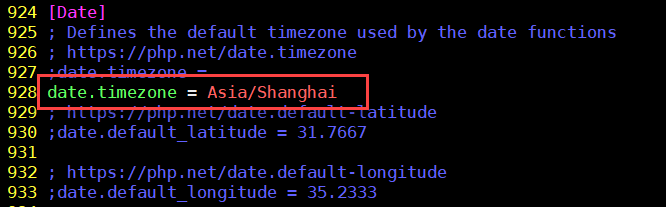
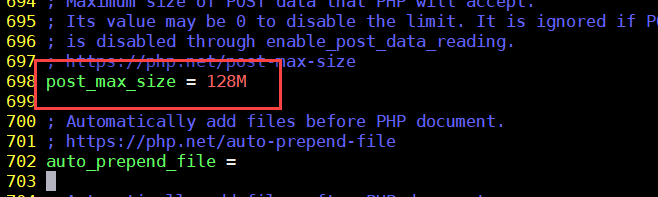
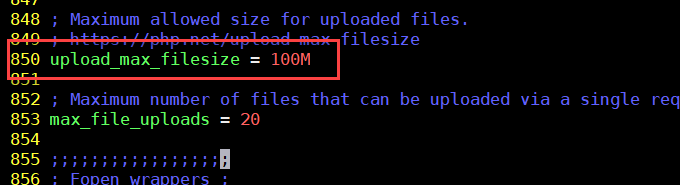
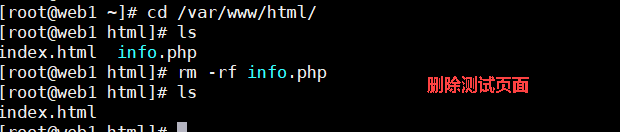
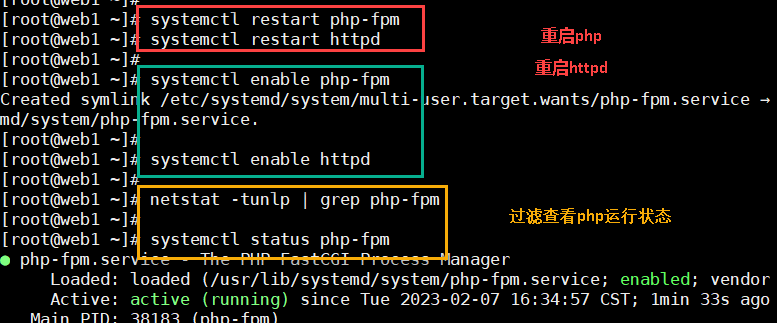
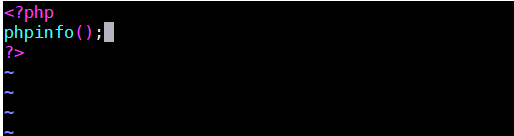
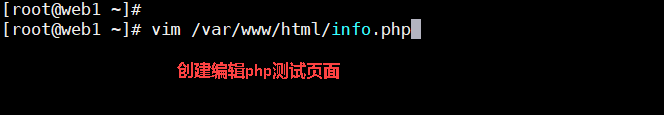
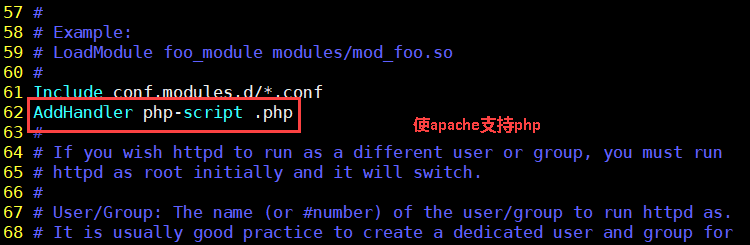
**dnf install -y https://rpms.remirepo.net/enterprise/remi-release-9.rpm**



max\_execution\_time程序执行时间限制

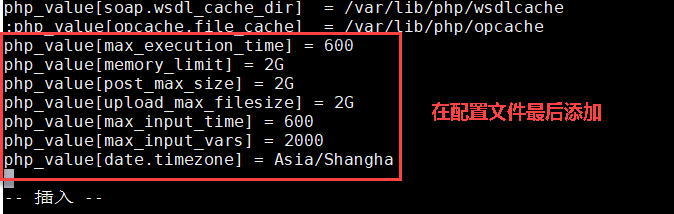
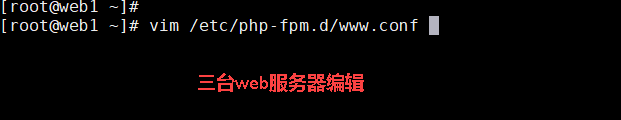
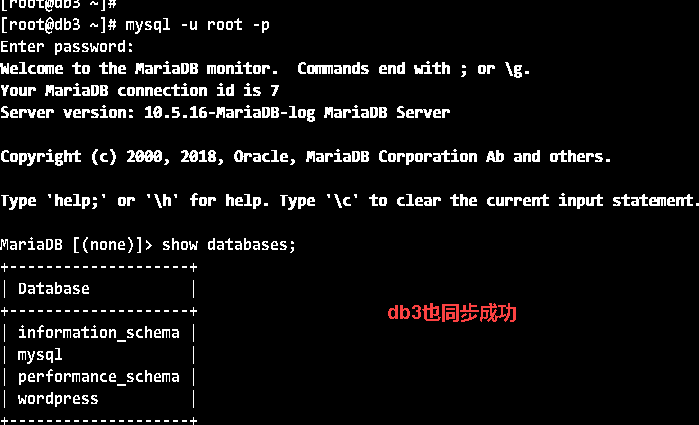
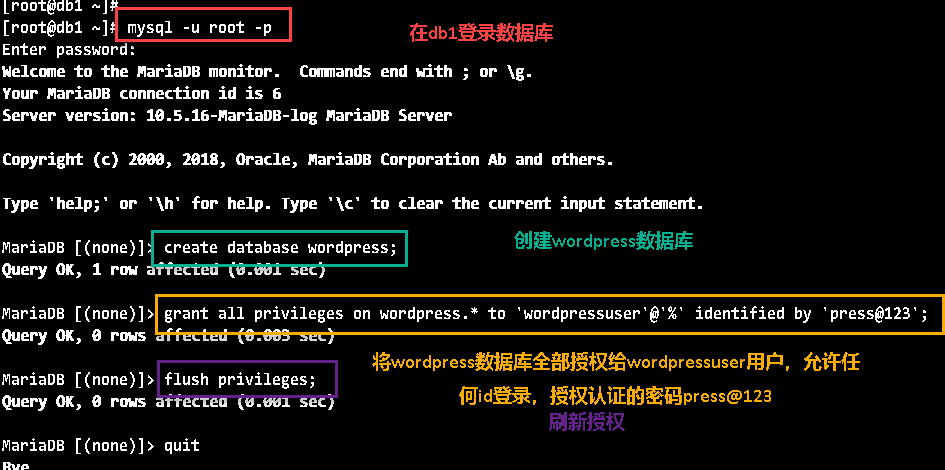
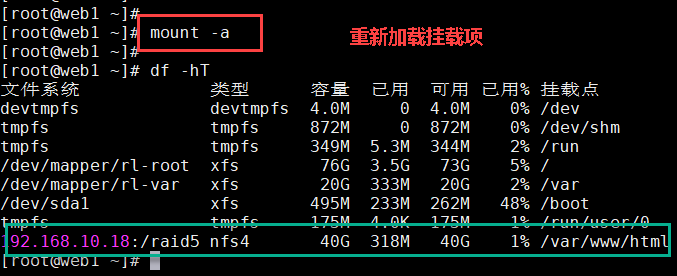
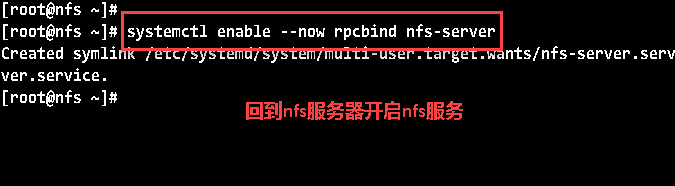
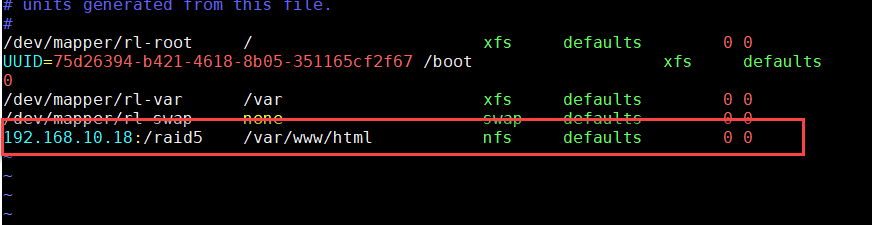
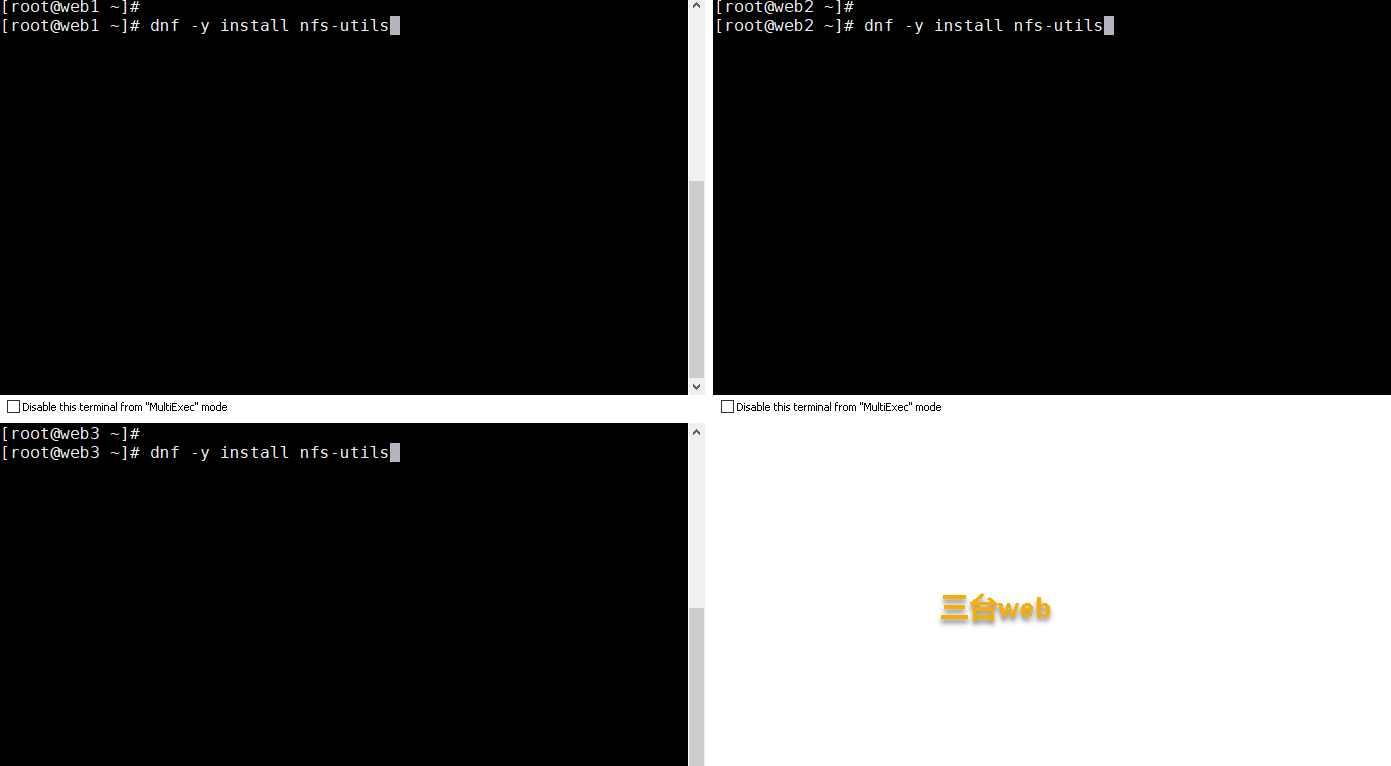
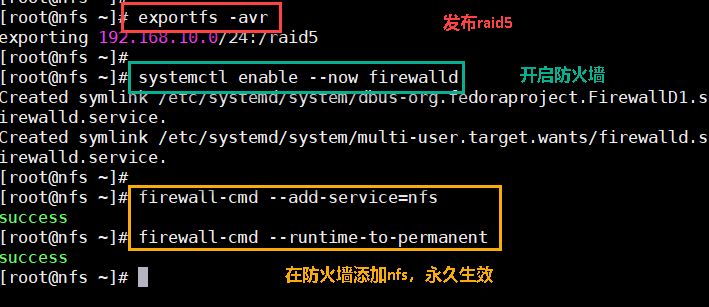
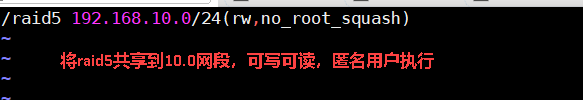
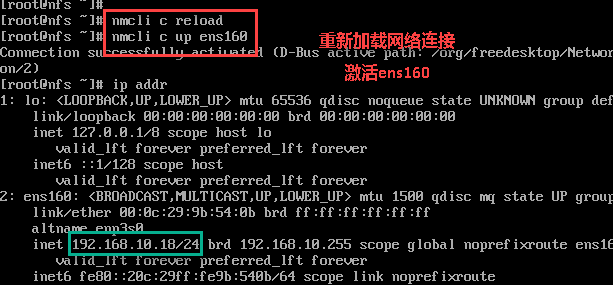
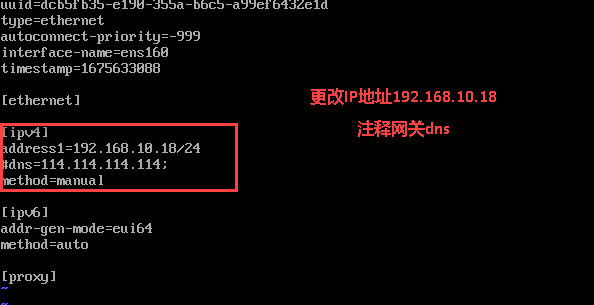
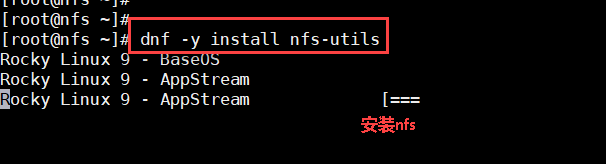
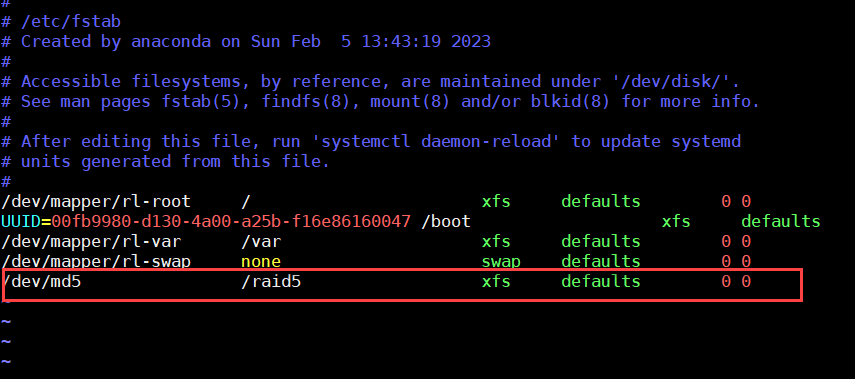
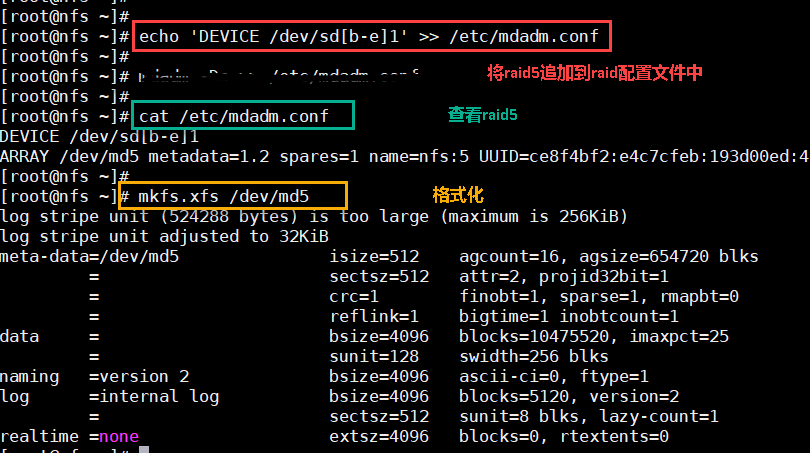
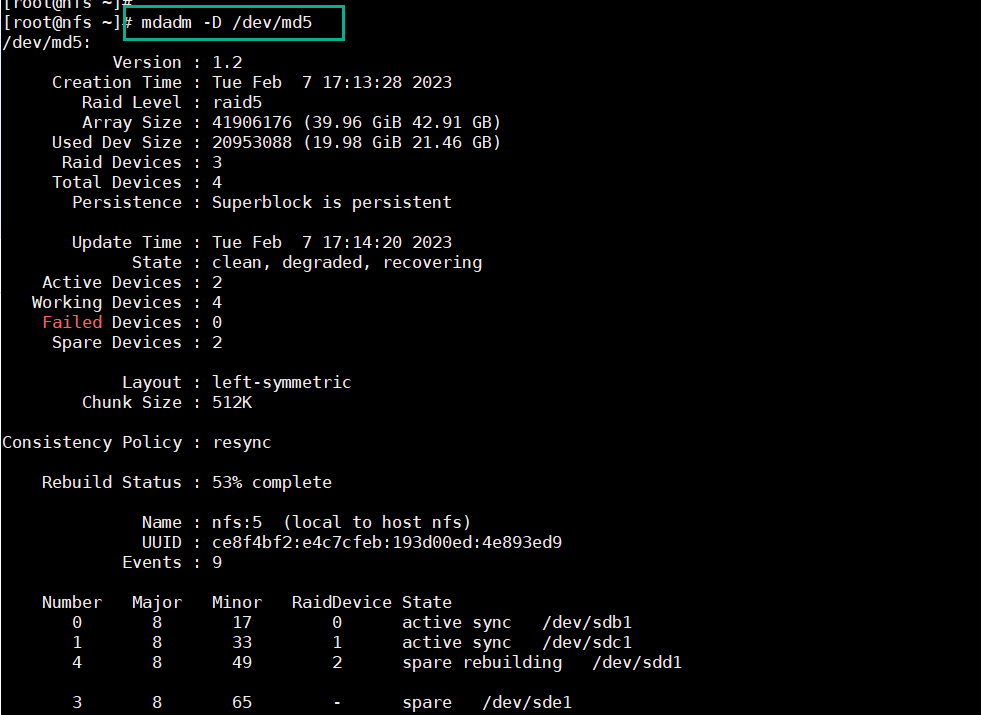
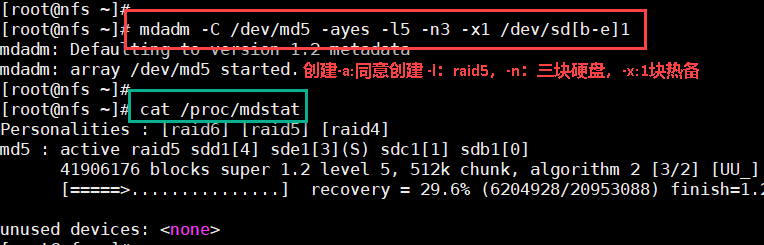
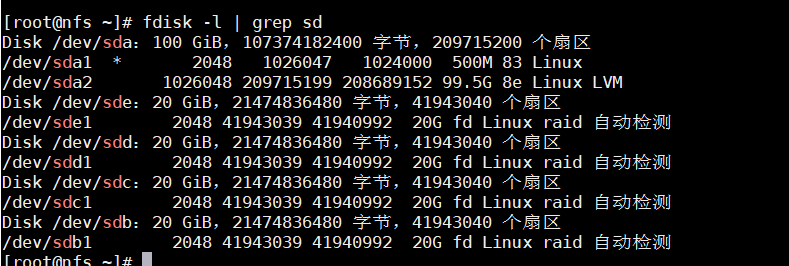
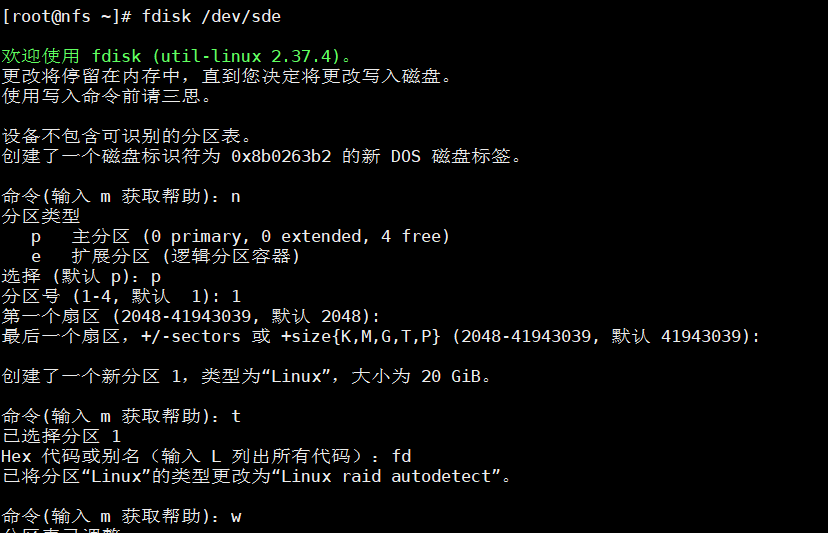
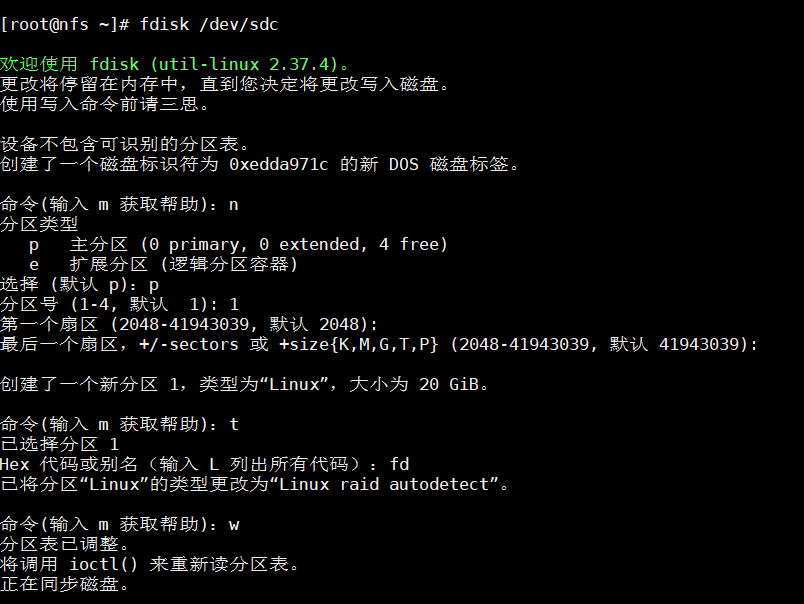
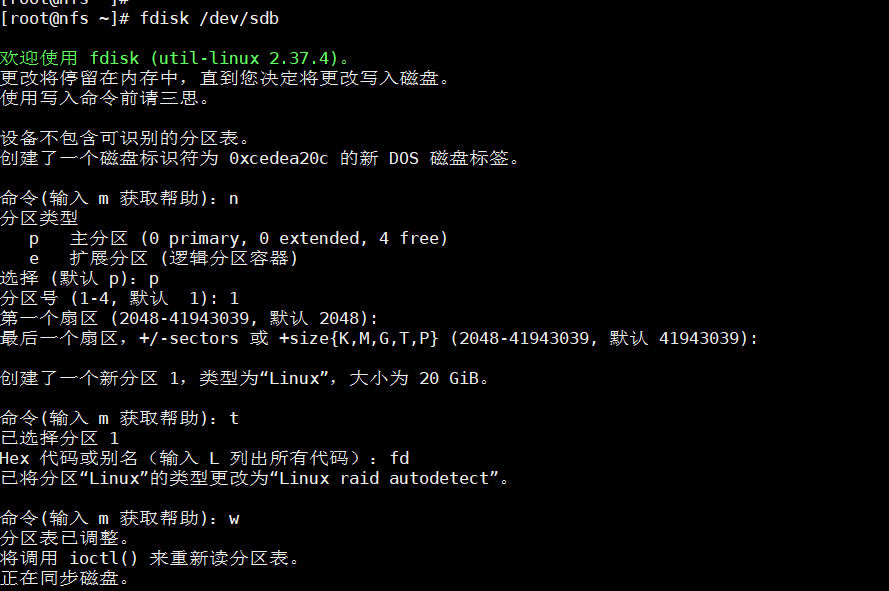
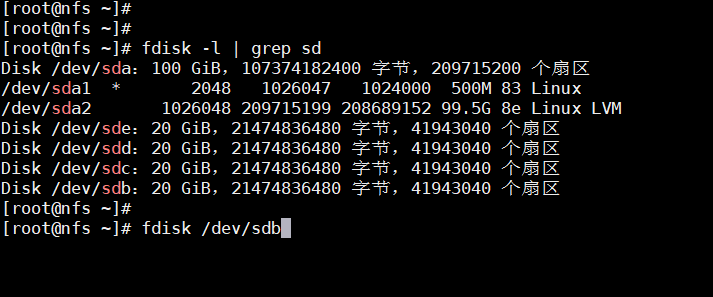
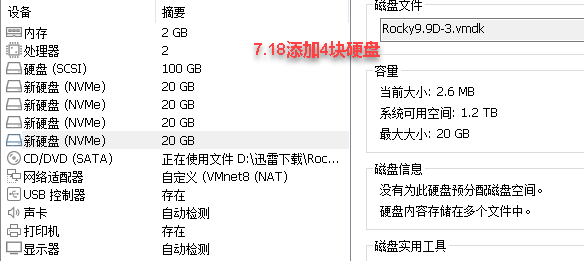
post\_max\_size最大POST数据限制

upload\_max\_filesize最大上传文件大小限制

83

NFS 192.168.10.18

安装和配置RAID5



**php\_value[max\_execution\_time] = 600**

**php\_value[memory\_limit] = 2G**

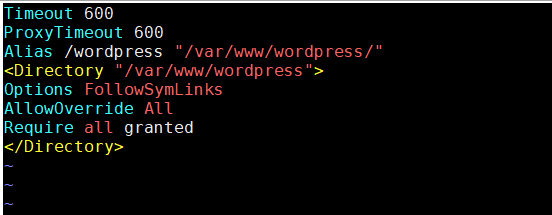
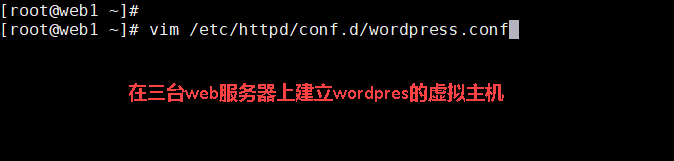
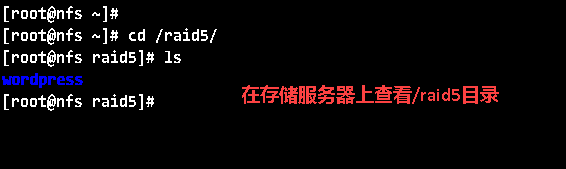
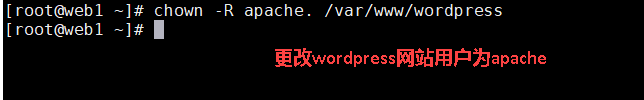
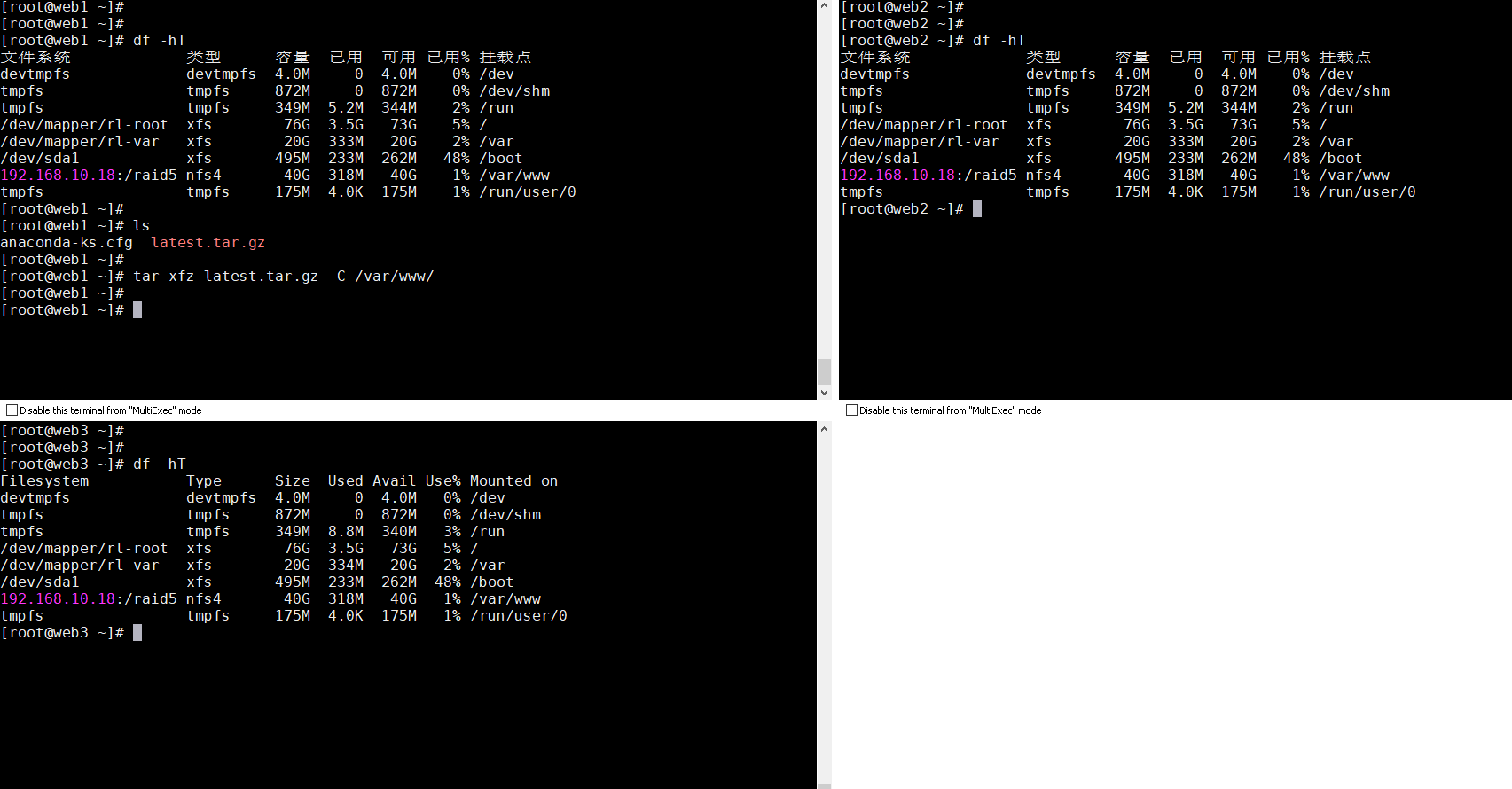
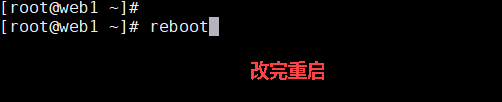
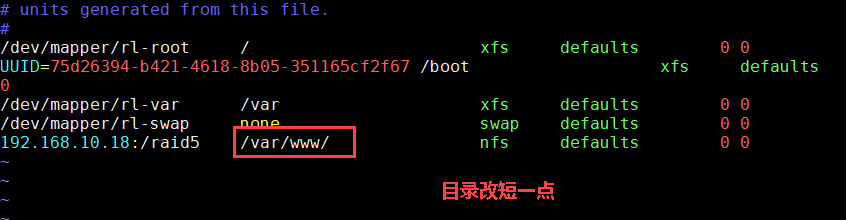
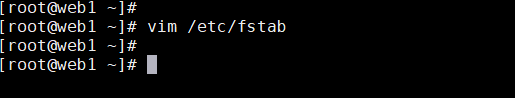
**php\_value[post\_max\_size] = 2G**

**php\_value[upload\_max\_filesize] = 2G**

**php\_value[max\_input\_time] = 600**

**php\_value[max\_input\_vars] = 2000**

**php\_value[date.timezone] = Asia/Shanghai**



**Timeout 600**

**ProxyTimeout 600**

**Alias /wordpress "/var/www/wordpress/"**

**<Directory "/var/www/wordpress">**

**Options FollowSymLinks**

**AllowOverride All**

**Require all granted**

**</Directory>**

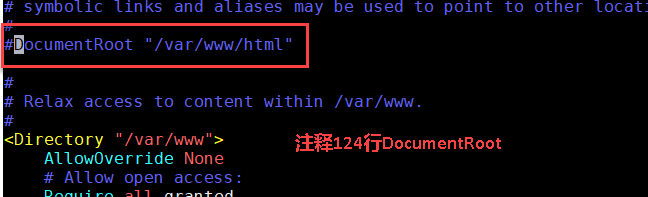
**Options FollowSymLinks**

禁止 Apache 显示该目录结构。用户就不会看到该目录下的文件和子目录列表了。

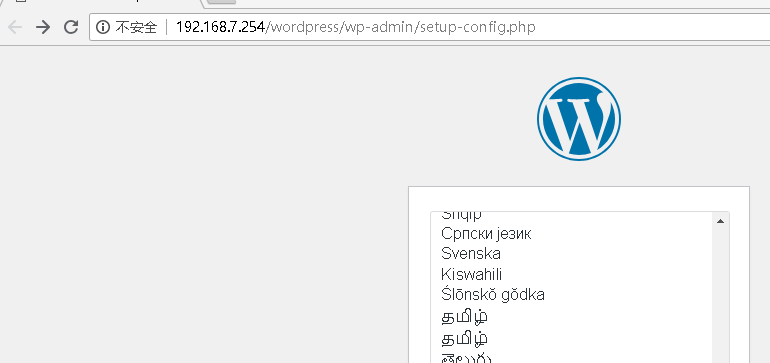
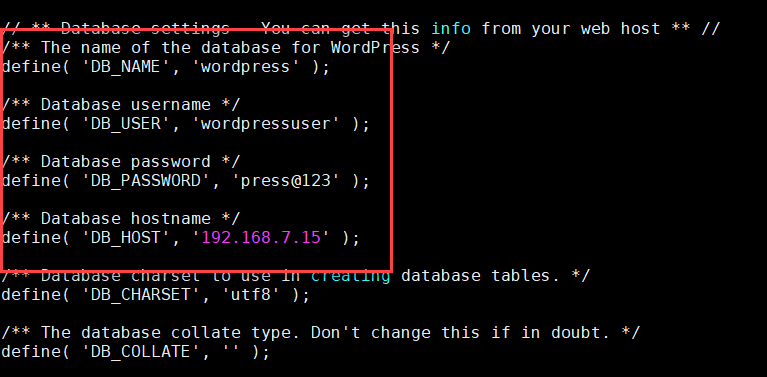
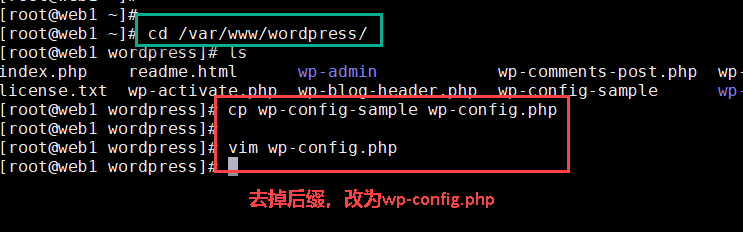
**AllowOverride All**

我们通常利用Apache的rewrite模块对URL进行重写，rewrite规则会写在 .htaccess 文件里。但要使 apache 能够正常的读取.htaccess 文件的内容，就必须对.htaccess 所在目录进行配置。

从安全性考虑，根目录的AllowOverride属性一般都配置成不允许任何Override

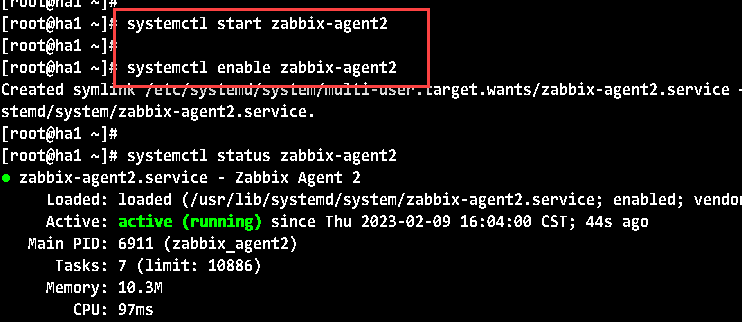
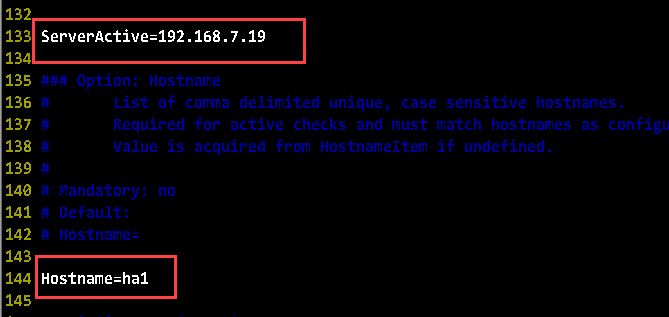
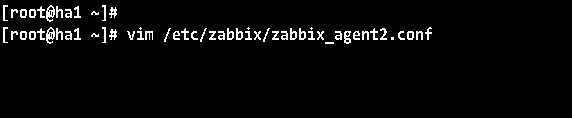
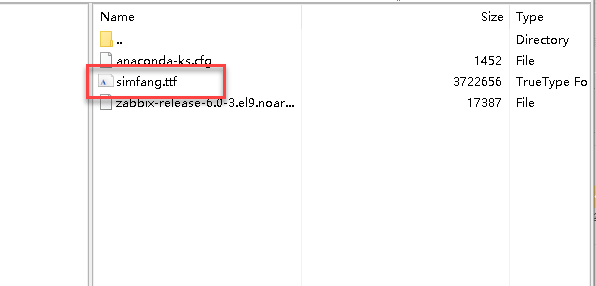
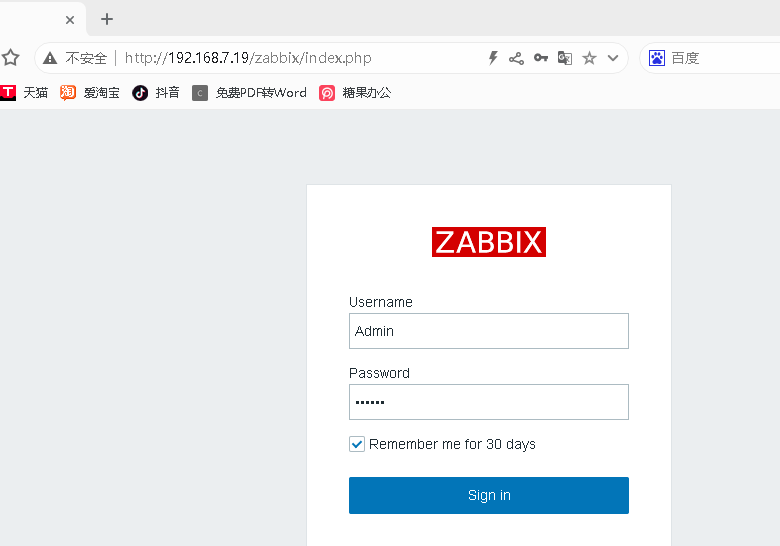
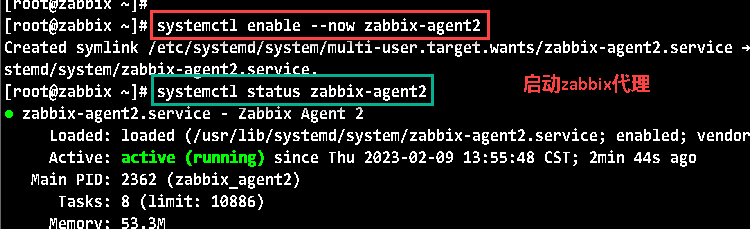
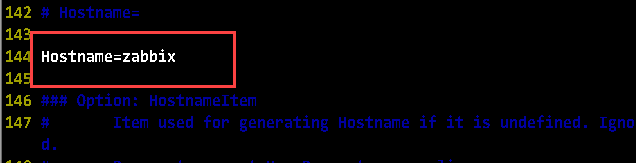
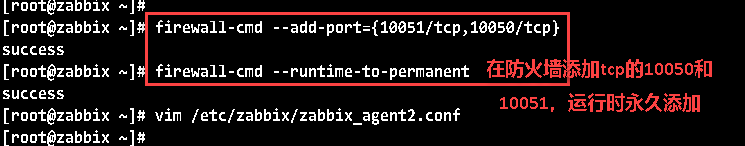
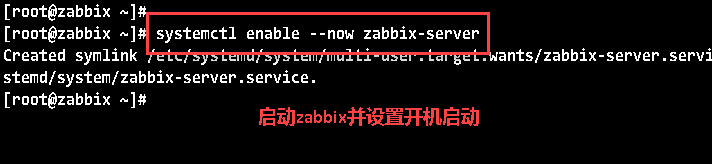
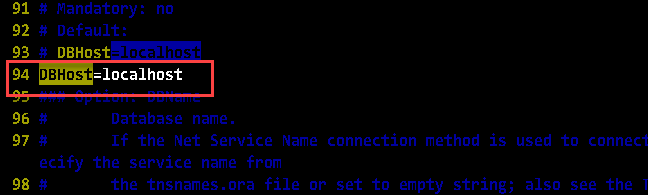
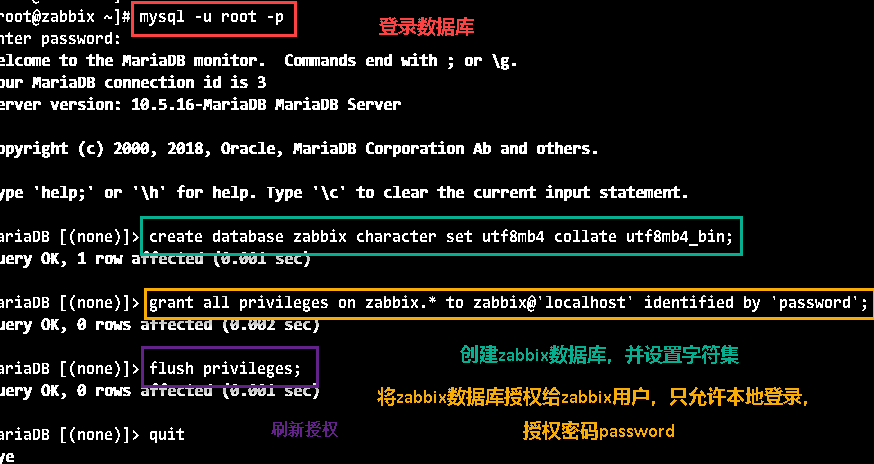
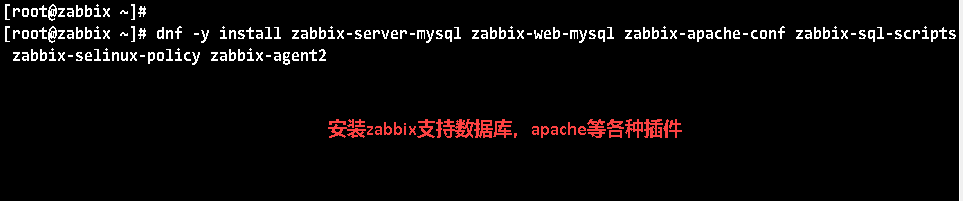
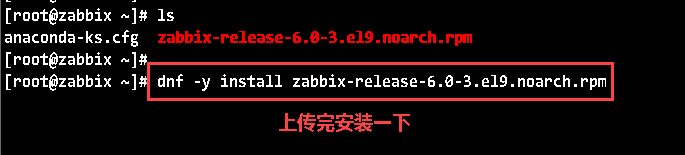
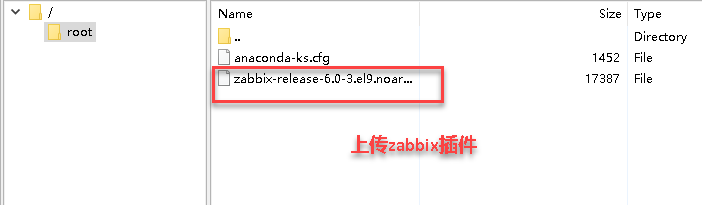
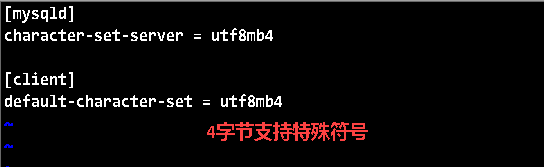
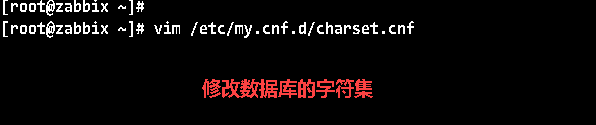
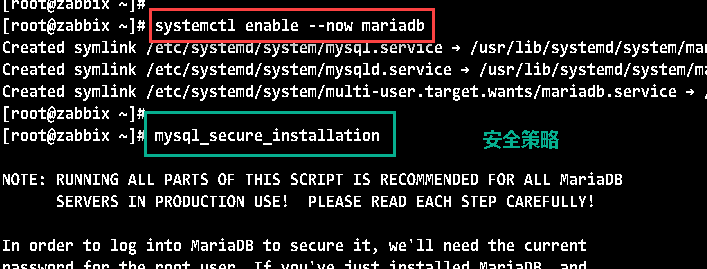
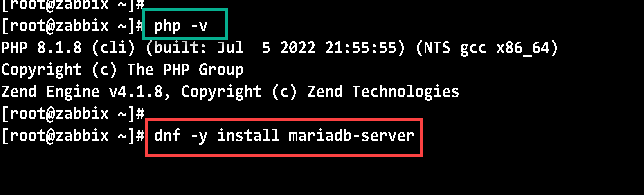
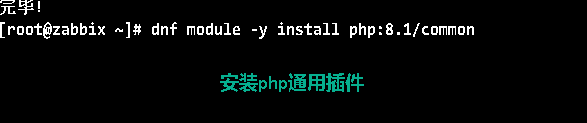
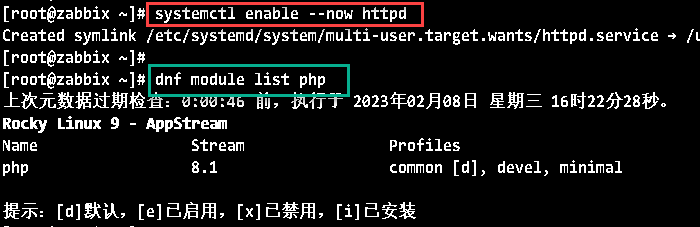
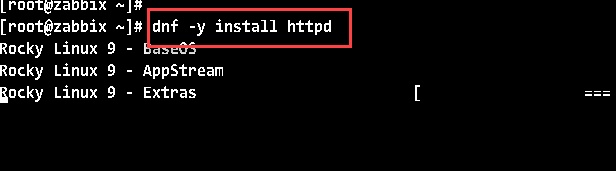
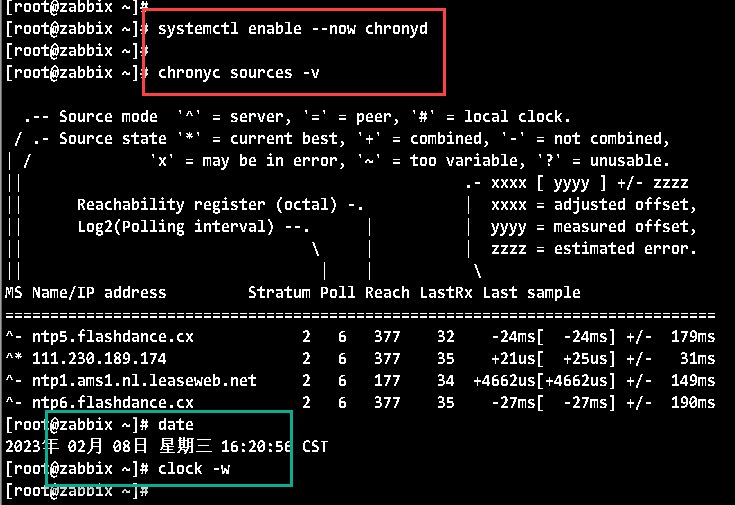
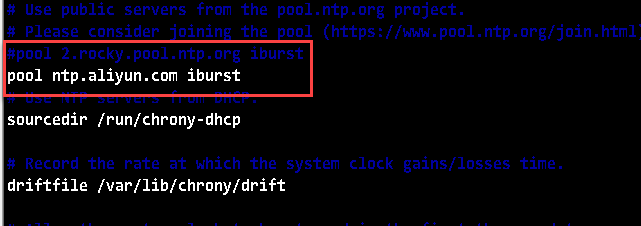
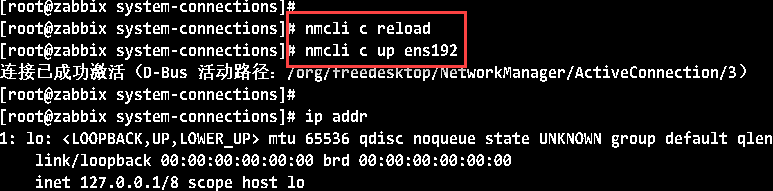
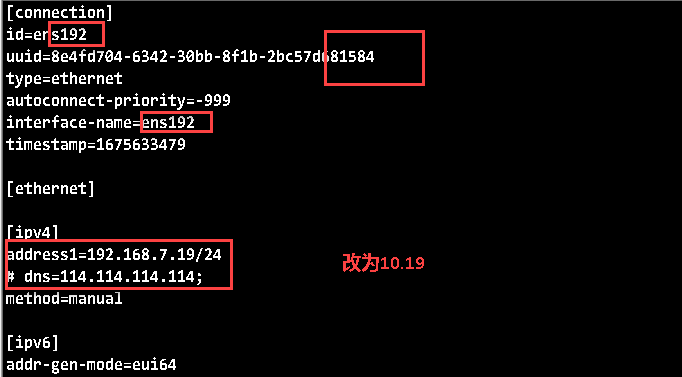
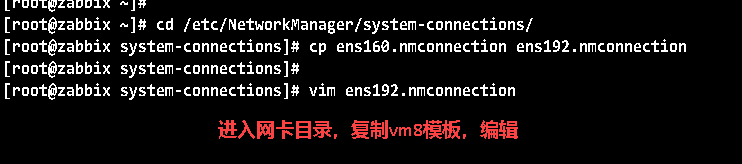
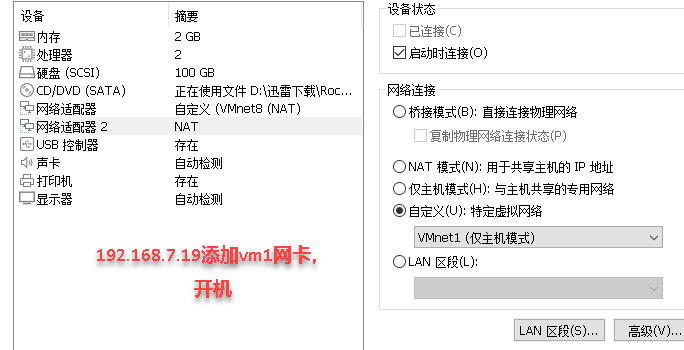


**Web1操作：**



Zabbix server 192.168.7.19

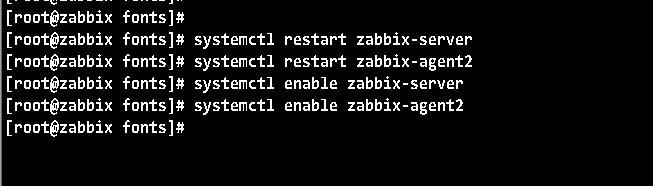
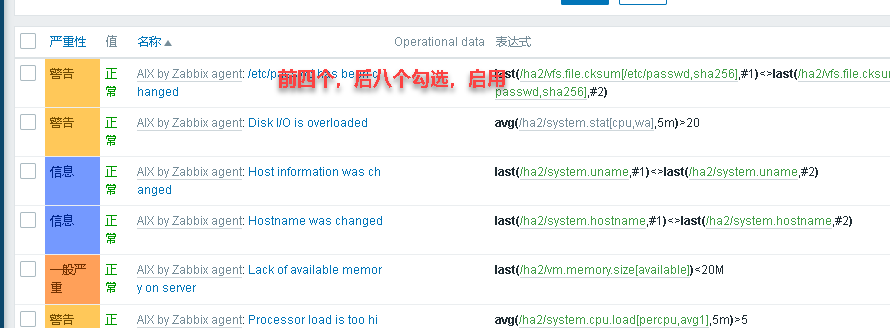
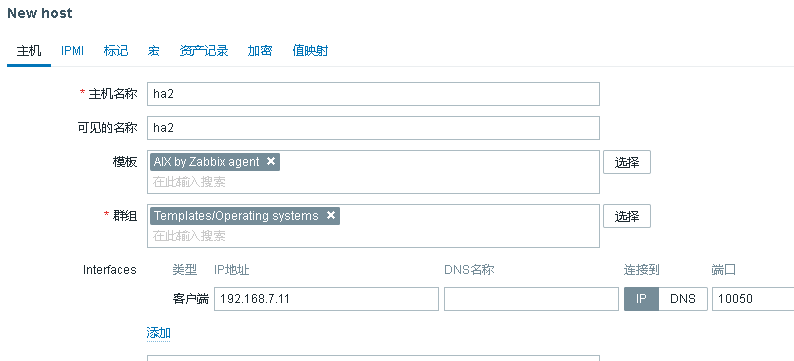
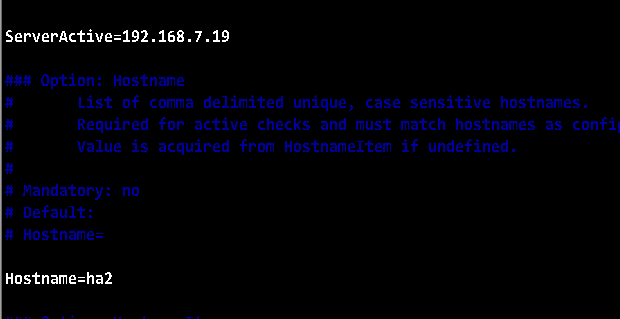
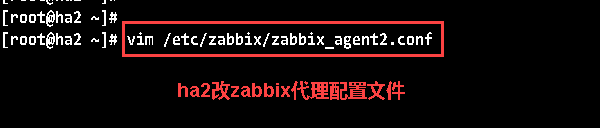
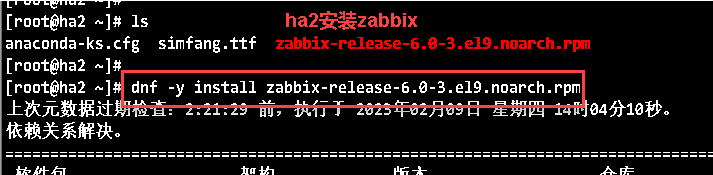
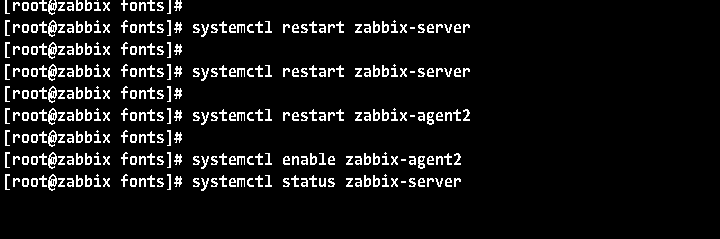
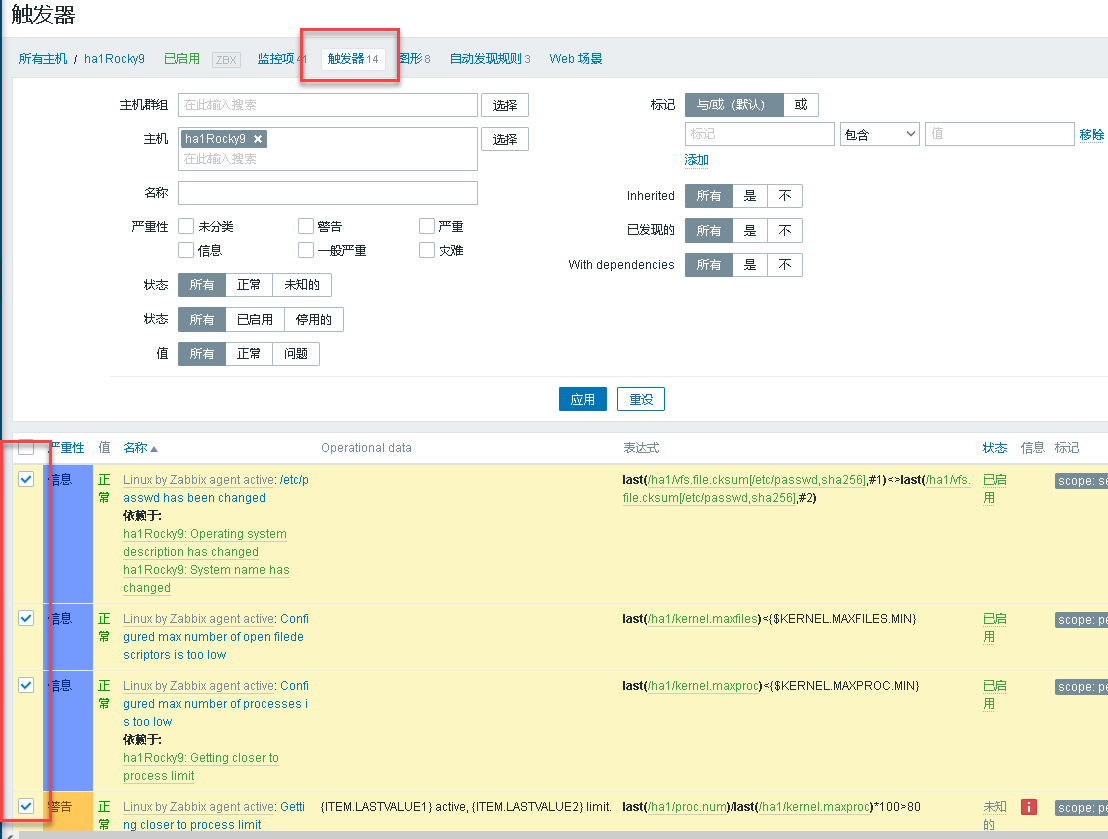
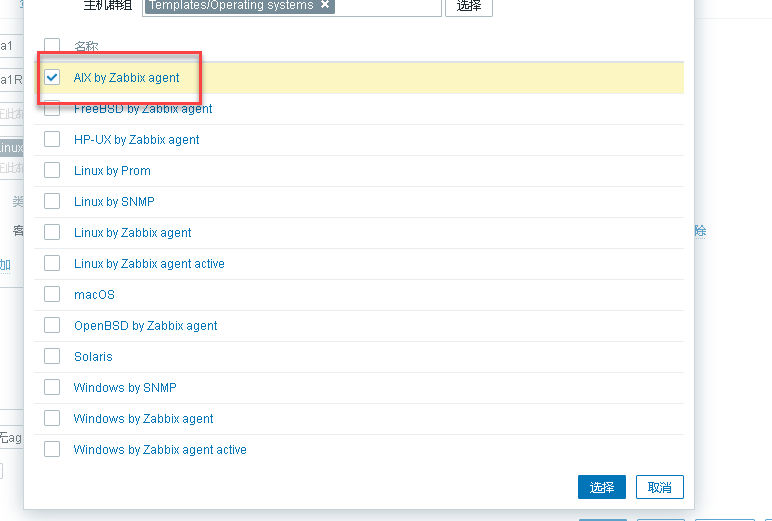
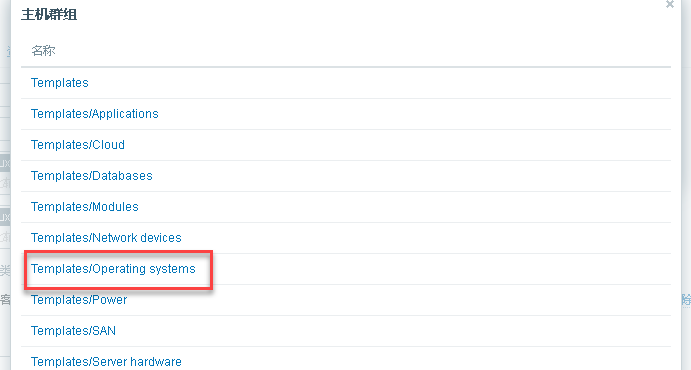
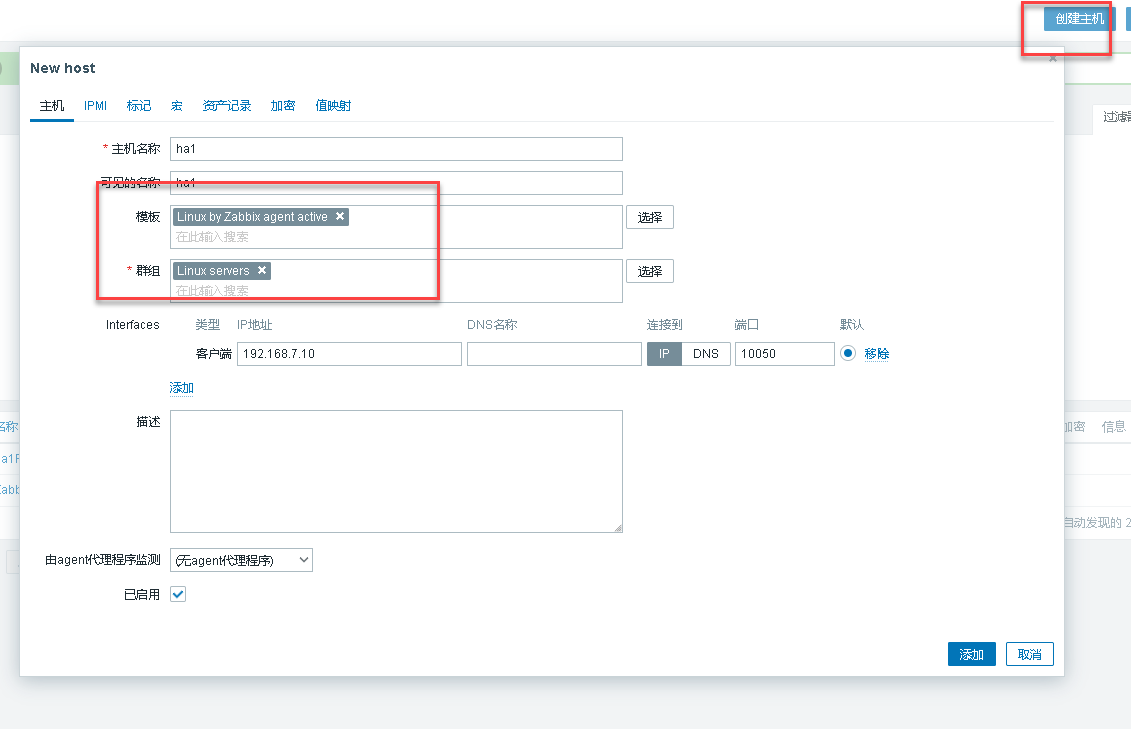
## 实验：在Rocky Linux 9安装和配置Zabbix6.0











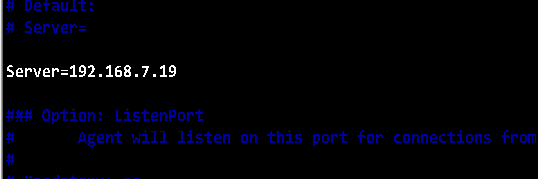
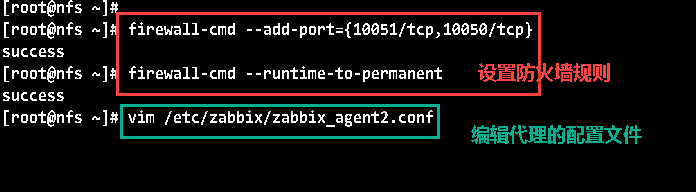
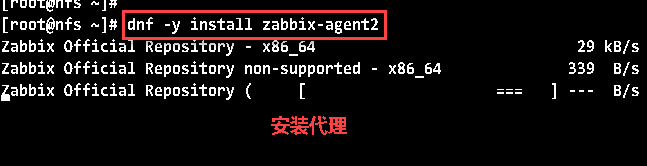
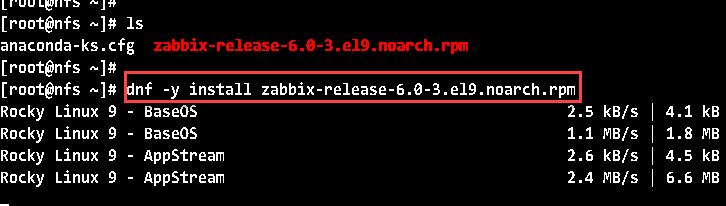
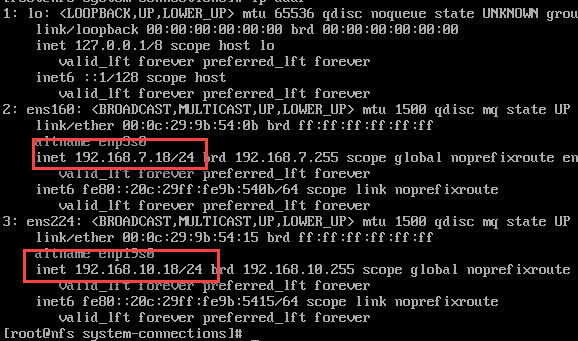
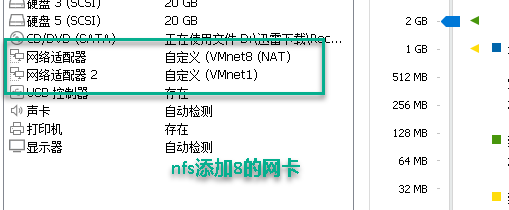
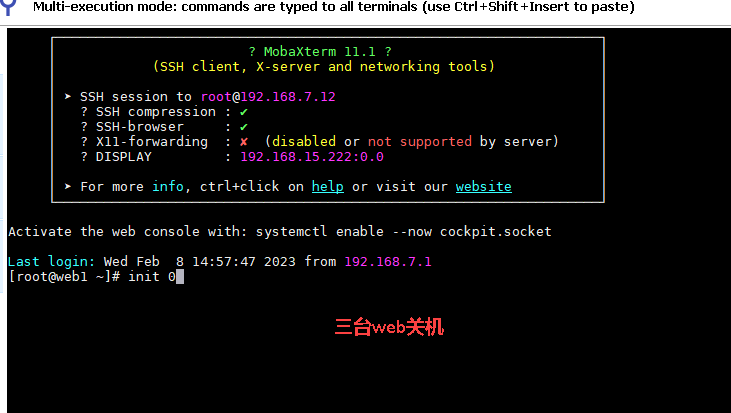
**2.监控Linux主机**

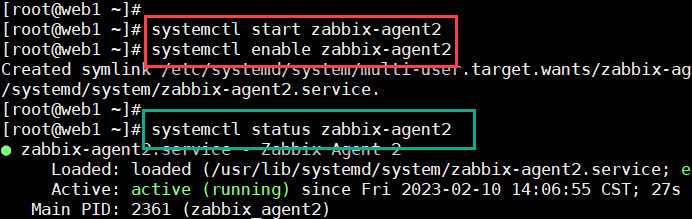
监控NFS存储的服务器需要2块网卡

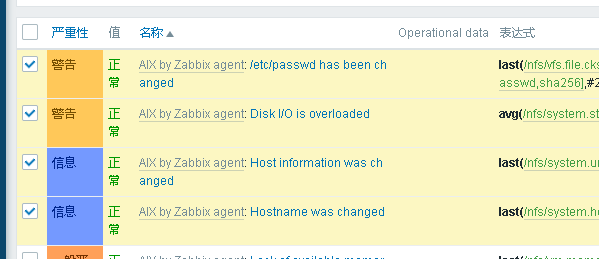
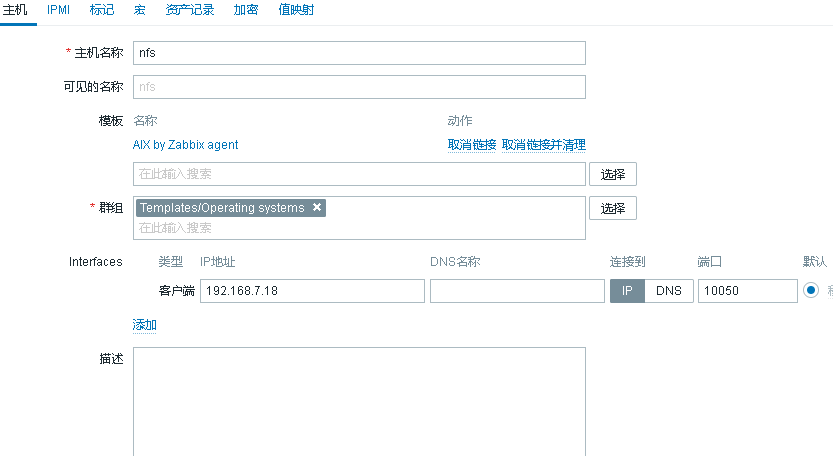
vmnet8 192.168.7.18 监控用的

vnmet1 192.168.10.18 存储用的

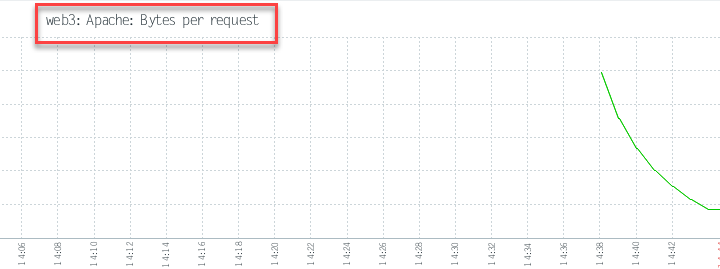
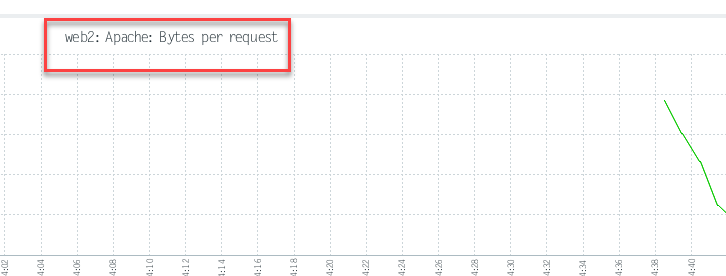
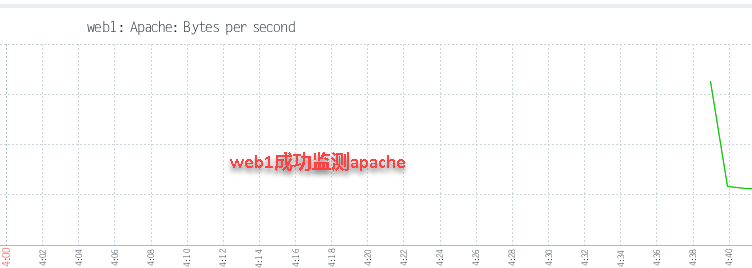
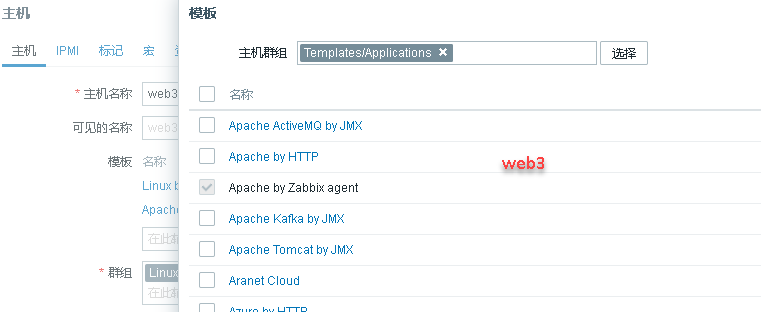
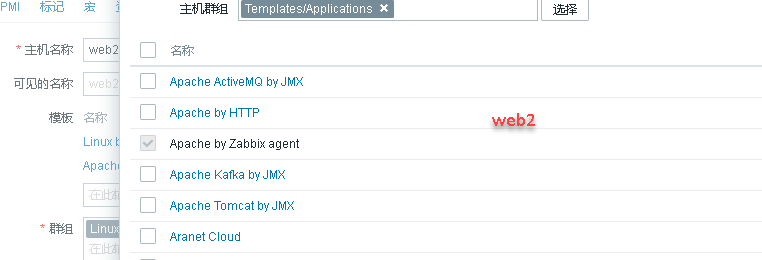
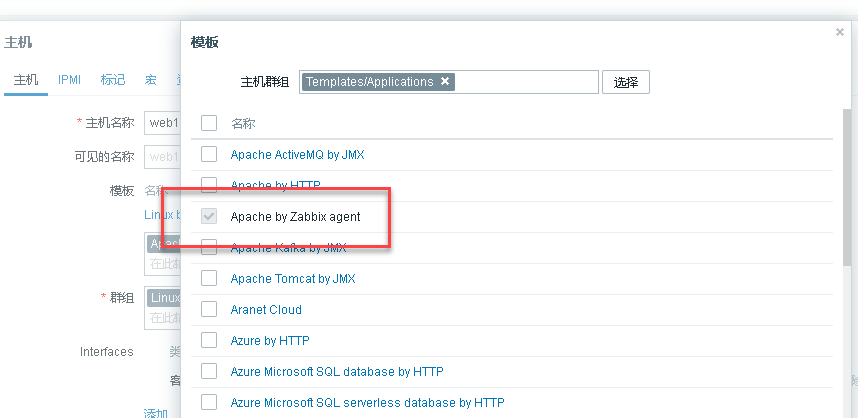
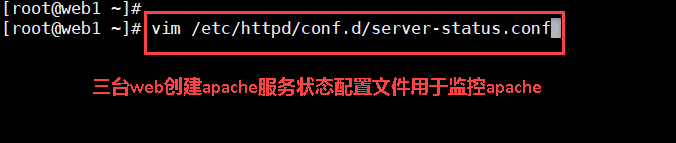
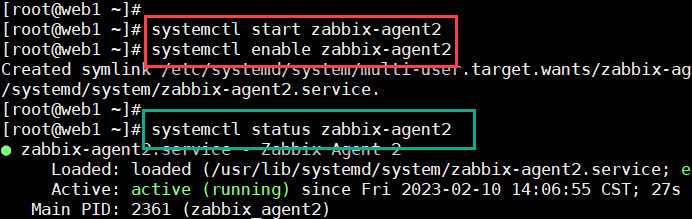
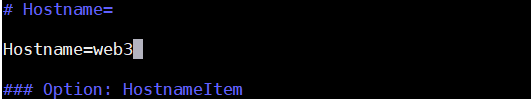
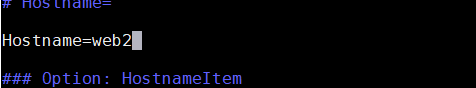
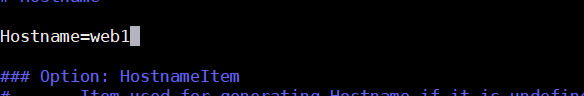
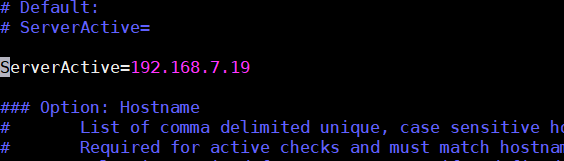
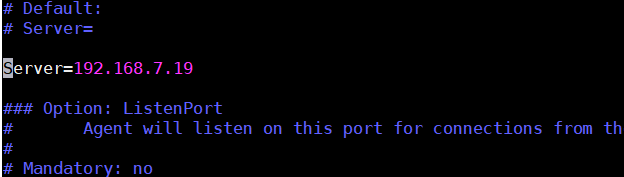
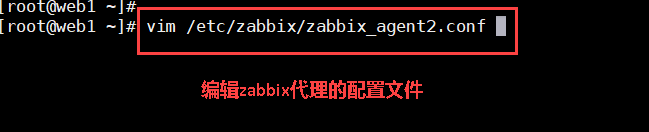
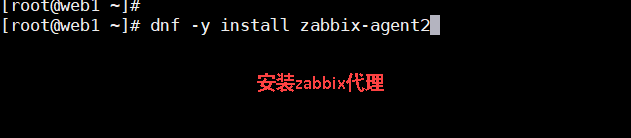
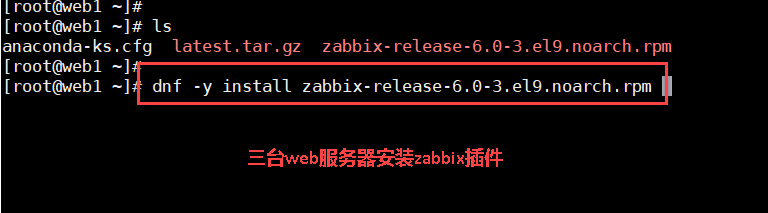
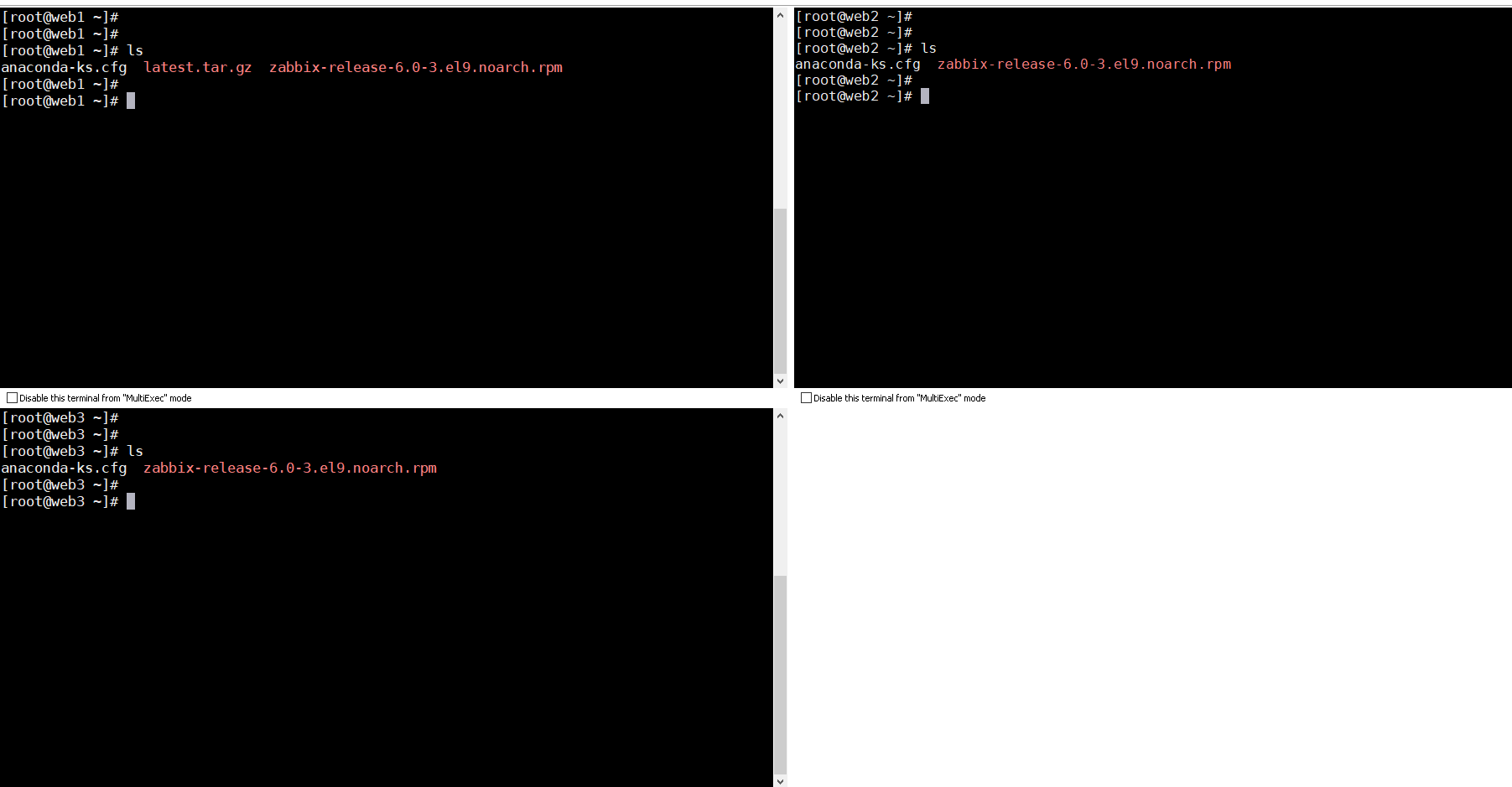
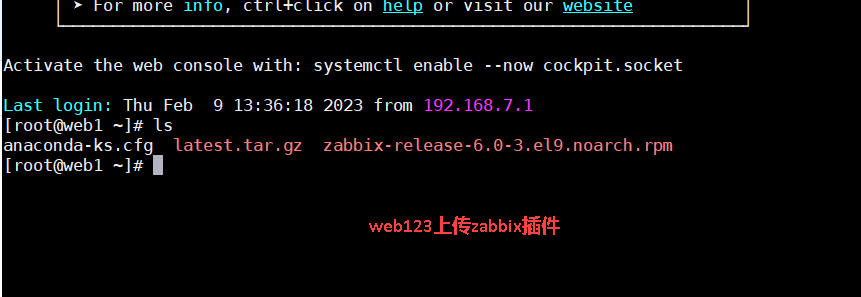
Zabbix Server只需要一块网卡 vmnet8





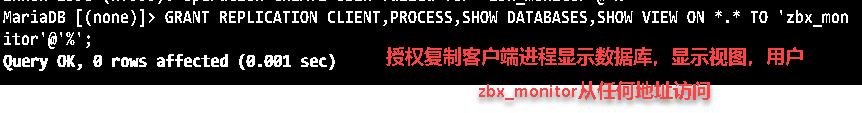
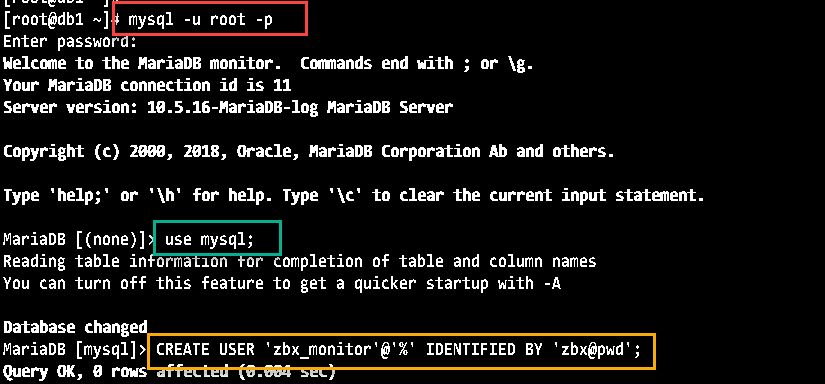
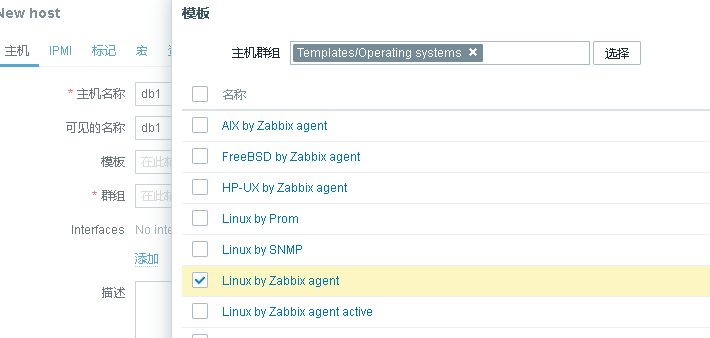
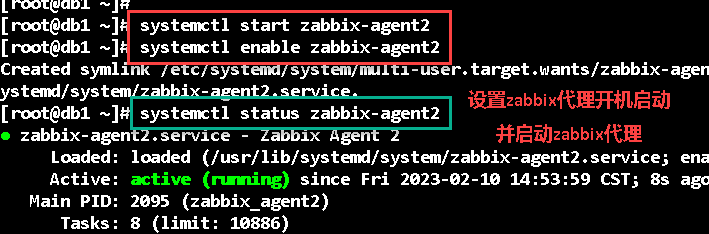
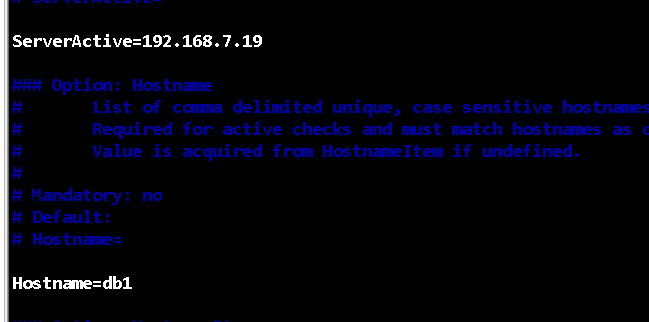
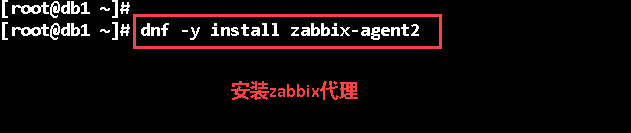
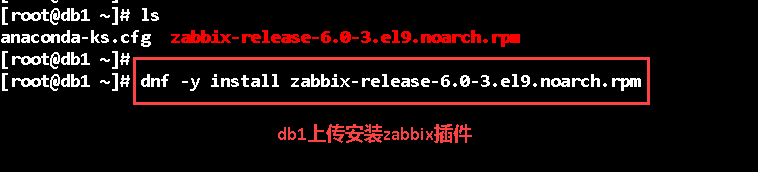


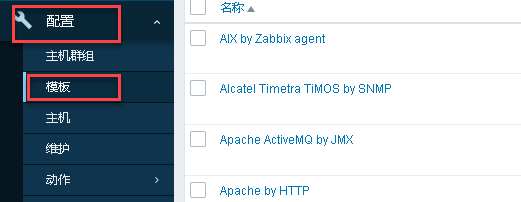
**3.监控apache**

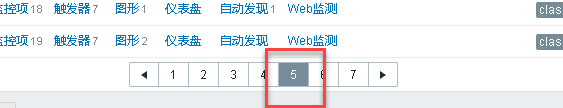


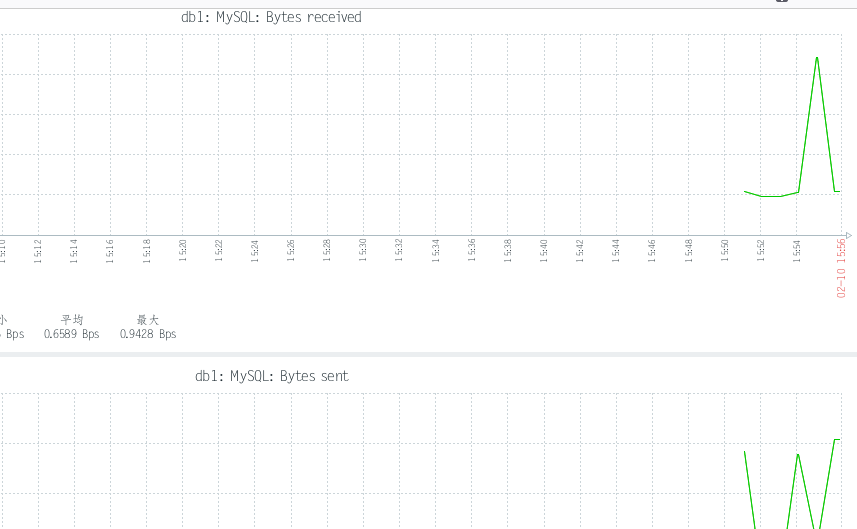
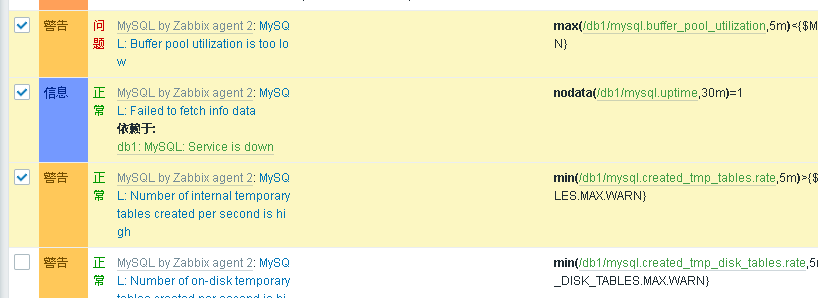
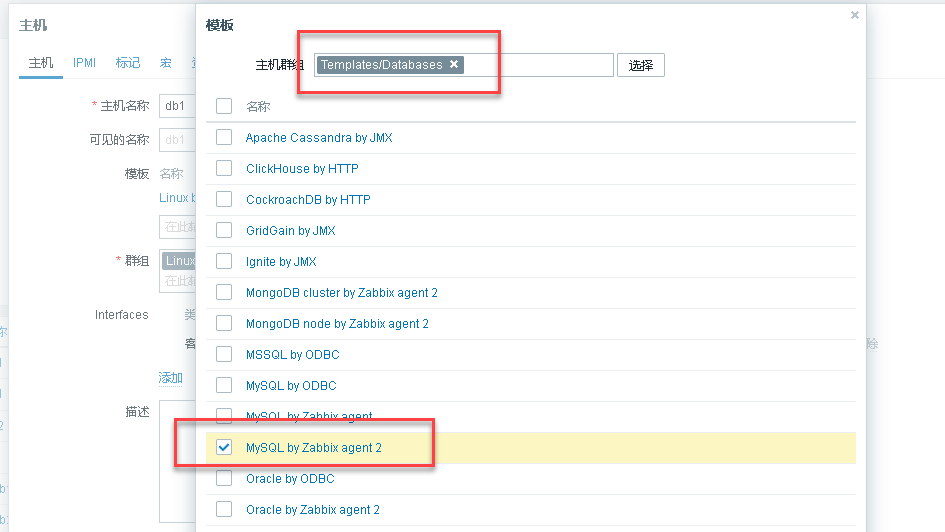
**3.监控数据库mariadb**

192.168.7.15









**1.监控Windows Server 2019**

