# Ansible配置管理LVS

（by本末， mail: p\_junkai@163.com）

功能：管理配置LVS，包括NAT、DR模式，配置文件细化至某个vip和某个单独的real server节点

1. **配置说明**
2. ansible目录结构

|  |
| --- |
| lvs  ├── hosts # inventory 文件  ├── install\_lvs # 安装lvs的role  │   ├── tasks  │   │   └── main.yml  │   └── templates  │   └── global.conf # 全局配置文件模板  ├── lvs\_virtual\_server # 配置virtual server的role  │   ├── tasks  │   │   └── main.yml  │   └── templates  │   └── vip.conf # virtual server模板  ├── lvs\_realserver # 配置real server的role  │   ├── tasks  │   │   └── main.yml  │   └── templates # real server配置文件模板  │   └── real.conf  └── real\_config # 真实服务器配置的role  │ ├── tasks  │ │ └── main.yml  │ └── templates  │ └── lvs\_for\_realserver.sh # 后端真实服务器脚本模板  ├── lvs\_del\_virtual\_server # 移除virtual server的role  │   └── tasks  │   └── main.yml  ├── lvs\_del\_realserver # 移除real server的role  │   └── tasks  │   └── main.yml  └── lvs.yml # 主playbook |

Keepalived目录结构

|  |
| --- |
| /etc/keepalived/  ├── keepalived.conf # 全局配置文件  ├── realserver # real server配置  │   ├── ad\_http # real server节点  │   │   └── 172.199.18.103.conf  │   │   └── 172.199.18.104.conf  │   ├── ad\_rabbitmq # real server节点  │   │   └── 172.199.18.103.conf  │   │ └── 172.199.18.104.conf  │   └── login\_pass # real server 节点  │   ├── 172.199.18.107.conf  │   └── 172.199.18.108.conf  └── virtual\_server # virtual server配置  │ ├── ad\_http.conf  │ ├── ad\_rabbitmq.conf  │ └── login\_pass.conf  └── backup # 备份目录 |

1. 过程简述
2. 安装：install\_lvs(设置aliyum或者epel的源)
3. 配置过程：

**下发全局配置文件**(keepalived.conf)

│

│(include指向/etc/keepalived/virtual\_server/\*.conf)

│

**下发vip的配置**(根据vrrp\_instance\_name生成不同的配置文件)

│

│（include指向/etc/keepalived/realserver/$对应实例目录/\*.conf）

│

**Realservr的配置**(每个真实服务器生成一个conf,如：1.1.1.1.conf)

│

│(重启keepalived)

│

**keepalived生效**

1. 配置文件
2. lvs.yml（主playbook）

|  |
| --- |
| ---  - hosts: "{{hosts}}"  sudo: yes  vars:  # virtual server  vrrp\_instance\_name: ad\_http # 实例名  nopreempt: false # 设置抢占  wan\_interface: eth0 # 外网卡接口  virtual\_router\_id: 49 # virtual\_router\_id  vip\_address: 172.199.18.100 # vip地址  vip\_netmask: 24 # vip掩码  virtual\_port: 80 # vip需要转发口的端口  lb\_algo: rr # 轮询模式  lb\_kind: DR # 工作模式  # nat lan gateway  lan\_interface: eth1 # NAT模式的网卡名  lan\_gateway: 192.168.19.100 # NAT模式虚拟的网关IP  # add realserver  real\_port: 80 # 真实服务器的端口  realserver\_list: # 真实服务器的地址  - 172.199.18.103  - 172.199.18.104  roles:  # 各角色列表，并打上tags，方便执行时候调用不同role  - { role: install\_lvs, tags: install }  - { role: lvs\_virtual\_server, tags: add\_vip }  - { role: lvs\_realserver, tags: add\_node }  - { role: real\_config, tags: real\_config }  - { role: lvs\_del\_virtual\_server, tags: del\_vip }  - { role: lvs\_del\_realserver, tags: del\_node } |

1. hosts(inventory文件)

|  |
| --- |
| [local]  Manager ansible\_ssh\_host=172.199.18.99  [lvs]  # 在inentory中标记好lvs角色和权重，其他方法也可以  lvs-master ansible\_ssh\_host=172.199.18.101 lvs\_state=master priority=100  lvs-master ansible\_ssh\_host=172.199.18.102 lvs\_state=backup priority=99  [website]  website1 ansible\_ssh\_host=172.199.18.103  website2 ansible\_ssh\_host=172.199.18.104 |

1. install\_lvs/tasks/main.yml

|  |
| --- |
| - name: Install ipvsadm & keepalived  yum: name={{ item }} update\_cache=yes  with\_items:  - keepalived  - ipvsadm    - name: Copy global config file  template: >  src=global.conf  dest=/etc/keepalived/keepalived.conf  - name: Touch intances directory  file: >  path=/etc/keepalived/{{ item }}  state=directory  with\_items:  - virtual\_server  - realserver  - backup |

Install\_lvs/templates/global.conf

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {  {% if lvs\_state == "master" %}  router\_id LVS\_MASTER  {% else %}  router\_id LVS\_BACKUP  {% endif %}  }  include virtual\_server/\*.conf |

1. lvs\_virtual\_server/tasks/main.yml

|  |
| --- |
| - name: Add vip config file to keepalived  template: >  src=vip.conf dest=/etc/keepalived/virtual\_server/{{ vrrp\_instance\_name }}.conf  - name: Set ip\_forward  sysctl: >  name=net.ipv4.ip\_forward  value=1  state=present  reload=yes  when: lb\_kind == "NAT"  - name: Restart keepalived  shell: /etc/init.d/keepalived restart |

lvs\_virtual\_server/templates/vip.conf

|  |
| --- |
| ######## {{ vrrp\_instance\_name }} virtual address ###########  vrrp\_instance {{ vrrp\_instance\_name }} {  {% if lvs\_state == "master" %}  state MASTER  {% else %}  state BACKUP  {% endif %}  {% if lvs\_state != "master" %}  {% if nopreempt %}  nopreempt  {% endif %}  {% endif %}  interface {{ wan\_interface }}  virtual\_router\_id {{ virtual\_router\_id }}  priority {{ priority }}  advert\_int 1  authentication {  auth\_type pass  auth\_pass 7032  }  virtual\_ipaddress {  {{ vip\_address }}/{{ vip\_netmask }}  }  }  {% if lb\_kind == "NAT" %}  ############ {{ vrrp\_instance\_name }} lan gateway #############  vrrp\_instance {{ vrrp\_instance\_name }}\_lan\_gateway {  {% if lvs\_state == "master" %}  state MASTER  {% else %}  state BACKUP  {% endif %}  {% if lvs\_state != "master" %}  {% if nopreempt %}  nopreempt  {% endif %}  {% endif %}  interface {{ lan\_interface }}  virtual\_router\_id {{ virtual\_router\_id + 1 }}  priority {{ priority }}  advert\_int 1  authentication {  auth\_type PASS  auth\_pass 7032  }  virtual\_ipaddress {  {{ lan\_gateway }}/24  }  }  {% endif %}  ############ {{ vrrp\_instance\_name }} real server #############  virtual\_server {{ vip\_address }} {{ virtual\_port }} {  delay\_loop 10  lb\_algo {{ lb\_algo }}  lb\_kind {{ lb\_kind }}  persistence\_timeout 15  protocol TCP  include /etc/keepalived/realserver/{{ vrrp\_instance\_name }}/\*.conf  } |

1. lvs\_realserver/tasks/main.yml

|  |
| --- |
| - name: Create realserver node directory  file: >  path=/etc/keepalived/realserver/{{ vrrp\_instance\_name }} state=directory  - name: Push realserver config  template: >  src=real.conf dest=/etc/keepalived/realserver/{{ vrrp\_instance\_name }}/{{ item }}.conf  with\_items: realserver\_list  - name: Restart keepalived  shell: /etc/init.d/keepalived restart |

lvs\_realserver/templates/real.conf

|  |
| --- |
| real\_server {{ item }} {{ real\_port }} {  weight 1  TCP\_CHECK {  connect\_timeout 10  nb\_get\_retry 3  delay\_before\_retry 3  connect\_port {{ real\_port }}  }  } |

1. real\_config/tasks/main.yml(该role在真实服务器上执行)

|  |
| --- |
| - name: Copy lvs node scripts  template: >  src=lvs\_for\_realserver.sh  dest=/etc/init.d/lvs-realserver  mode=0755  - name: Run node lvs scripts  shell: /etc/init.d/lvs-realserver start |

Real\_config/templates/lvs\_for\_realserver.sh(当然我写了很多废话)

|  |
| --- |
| #!/bin/bash  ### BEGIN INIT INFO  # Provides: lvs-realserver  # Default-Start: 2 3 4 5  # Default-Stop: 0 1 6  ### END INIT INFO  . /etc/rc.d/init.d/functions  {% if lb\_kind == "NAT" %}  # route table: /etc/iproute2/rt\_tables  LAN\_IP={{ lvs\_lan\_address }}  LAN\_GW={{ lan\_gateway }}  LAN\_SCOPE="{{ lvs\_lan\_address | regex\_replace('(^\d+\.\d+\.\d+)\.\d+$','\\1.0') }}/24"  ROUTE\_TABLE=lvs  LAN\_INT={{ lan\_interface }}  # add route to lvs  startRealserver() {  [ -n "`grep lvs /etc/iproute2/rt\_tables`" ] || echo "50 lvs" >> /etc/iproute2/rt\_tables  [ -n "`ip addr show $LAN\_INT | grep -o $LAN\_IP`" ] || ip addr add $LAN\_IP/24 brd + dev $LAN\_INT  [ -z "`ip rule list | grep -o $LAN\_IP`" ] || ip rule del from $LAN\_IP  ip route flush table $ROUTE\_TABLE >/dev/null 2>&1  ip route add table $ROUTE\_TABLE $LAN\_SCOPE dev $LAN\_INT:$ROUTE\_TABLE proto kernel scope link src $LAN\_IP >/dev/null 2>&1  ip route add table $ROUTE\_TABLE default via $LAN\_GW dev $LAN\_INT:$ROUTE\_TABLE src $LAN\_IP >/dev/null 2>&1  ip rule add from $LAN\_IP table $ROUTE\_TABLE prio 500 >/dev/null 2>&1  echo -e "\n\e[0;33mRealServer start...\e[0m\n"  }  # del route  stopRealserver() {  [ -z "`grep lvs /etc/iproute2/rt\_tables`" ] || sed -i "s/50 lvs//" /etc/iproute2/rt\_tables  [ -z "`ip addr show $LAN\_INT | grep -o $LAN\_IP`" ] || ip addr del $LAN\_IP/24 brd + dev $LAN\_INT  [ -z "`ip rule list | grep -o $LAN\_IP`" ] || ip rule del from $LAN\_IP  echo -e "\n\e[0;31mRealServer stop...\e[0m\n"  }  # show status  realStatus() {  showVip=`ip addr show $LAN\_INT | grep -Po “(?<=inet )$LAN\_IP”`  showRule=`ip rule list | grep lvs | head -n 1`  showTable=`grep "lvs" /etc/iproute2/rt\_tables`  showRoute=`ip route show | grep $LAN\_IP`  if [ ! "$showVip" -o ! "$showRule" -o ! "$showTable" -o ! "$showRoute" ];then  echo -e "\n\e[031mRealServer is not running...\e[0m"  else  echo -e "\n\e[032mRealServer is running...\e[0m"  fi  echo -e "  LVS lb\_kind:\t\t\e[0;33mNAT\e[0m  RealServer VIP:\t\t\e[0;33m$showVip\e[0m  RealServer Table:\t\e[0;33m$showTable\e[0m  RealServer Rule:\t\e[0;33m$showRule\e[0m  RealServer Route:\t\e[0;33m$showRoute\e[0m\n"  }  case "$1" in  start)  startRealserver  ;;  stop)  stopRealserver  ;;  status)  realStatus  ;;  \*) echo -e "\n\e[0;31mError:\e[0m Usage: $0 {start|status|stop}\n"  exit 1  ;;  esac  exit 0  {% else %}  VIP={{ vip\_address }}  startRealserver() {  #ifconfig lo:0 $VIP netmask 255.255.255.255 broadcast $VIP  ip addr add $VIP/32 broadcast $VIP dev lo  /sbin/route add -host $VIP dev lo  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 2>&1  echo -e "\n\e[0;33mRealServer Start...\e[0m\n"  }    stopRealserver() {  #ifconfig lo:0 down  route del $VIP >/dev/null 2>&1  ip addr del $VIP/32 broadcast $VIP dev lo  echo "0" >/proc/sys/net/ipv4/conf/lo/arp\_ignore  echo "0" >/proc/sys/net/ipv4/conf/lo/arp\_announce  echo "0" >/proc/sys/net/ipv4/conf/all/arp\_ignore  echo "0" >/proc/sys/net/ipv4/conf/all/arp\_announce  echo -e "\n\e[0;31mRealServer Stop...\e[0m\n"  }  realStatus() {  # Status of LVS-DR real server.  showVip=`ip addr show lo | grep -Po "(?<=inet )$VIP"`  showRoute=`ip route show | grep $VIP`  if [ ! "$showVip" -o ! "$showRoute" ];then  echo -e "\n\e[0;32mRealServer is not running...\e[0m"  else  echo -e "\n\e[0;33mRealServer is Running...\e[0m"  fi  echo -e "  LVS lb\_kind:\t\t\e[0;33mDR\e[0m  RealServer Vip:\t\t\e[0;33m$showVip\e[0m  RealServer Route:\t\e[0;33m$showRoute\e[0m\n  ARP\_Response:  \t\e[0;33m/proc/sys/net/ipv4/conf/lo/arp\_ignore = `cat /proc/sys/net/ipv4/conf/lo/arp\_ignore`\e[0m  \t\e[0;33m/proc/sys/net/ipv4/conf/lo/arp\_announce = `cat /proc/sys/net/ipv4/conf/lo/arp\_announce`\e[0m  \t\e[0;33m/proc/sys/net/ipv4/conf/all/arp\_ignore = `cat /proc/sys/net/ipv4/conf/all/arp\_ignore`\e[0m  \t\e[0;33m/proc/sys/net/ipv4/conf/all/arp\_announce = `cat /proc/sys/net/ipv4/conf/all/arp\_announce`\e[0m  "  }  case "$1" in  start)  startRealserver  ;;  stop)  stopRealserver  ;;  status)  realStatus  ;;  \*)  # Invalid entry.  echo -e "\n\e[0;31mERROR:\e[0m Usage: $0 {start|status|stop}\n"  exit 1  ;;  esac  exit 0  {% endif %} |

1. lvs\_del\_virtual\_server/tasks/main.yml

|  |
| --- |
| - name: Create virttual server backup directory  file: >  path=/etc/keepalived/backup/virtual\_server  state=directory  - name: Remove virtual server  shell: mv /etc/keepalived/virtual\_server/{{ vrrp\_instance\_name }}.conf /etc/keepalived/backup/virtual\_server/  - name: Restart keepalived  shell: /etc/init.d/keepalived restart |

1. lvs\_del\_realserver/tasks/main.yml

|  |
| --- |
| - name: Create realserver backup directory  file: >  path=/etc/keepalived/backup/realserver/{{ vrrp\_instance\_name }}  state=directory  - name: Remove realserver node  shell: >  mv /etc/keepalived/realserver/{{ vrrp\_instance\_name }}/{{ item }}.conf /etc/keepalived/backup/realserver/{{ vrrp\_instance\_name }}/{{ item }}.conf  with\_items: realserver\_list  - name: Restart keepalived  shell: /etc/init.d/keepalived restart |

1. **命令行执行**

|  |
| --- |
| # 配置2台lvs  ansible-playbook lvs.yml -e "hosts=lvs" -t install,add\_vip,add\_node |

|  |
| --- |
| # 配置后端真实服务器  ansible-playbook lvs.yml -e "hosts=website" -t real\_config |

|  |
| --- |
| # 移除某个vip节点  ansible-playbook lvs.yml -e "hosts=website" -t del\_vip |

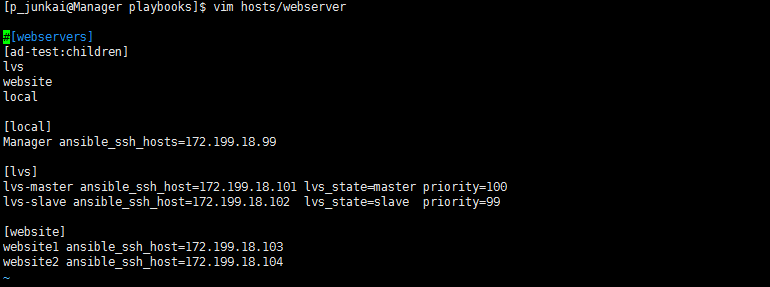
|  |
| --- |
| # 移除某个real server节点  ansible-playbook lvs.yml -e "hosts=website" -t del\_node |

1. **测试**

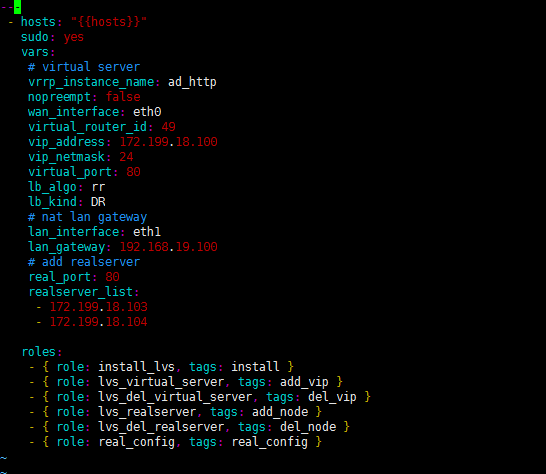
**系统环境**：CentOS 6.2 x86\_64

**Ansible**：1.9.2

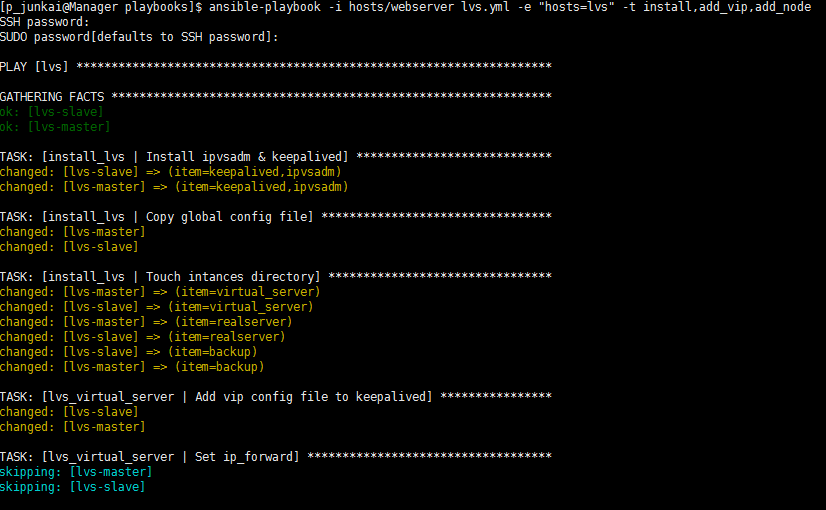
**inventory**：

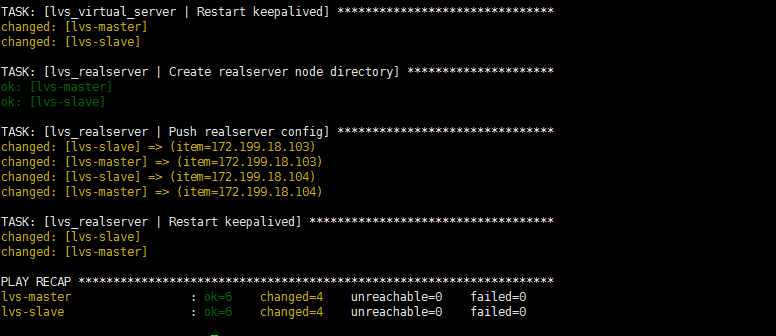


**Playbook**:

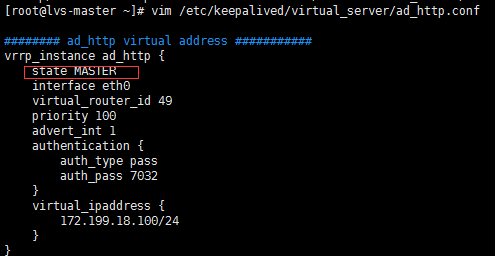


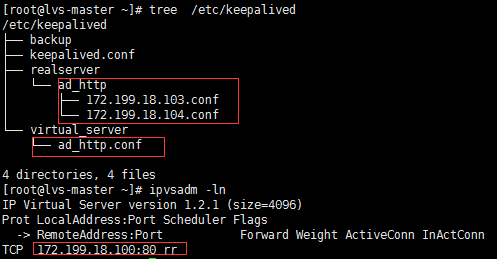
在2台lvs上执行：



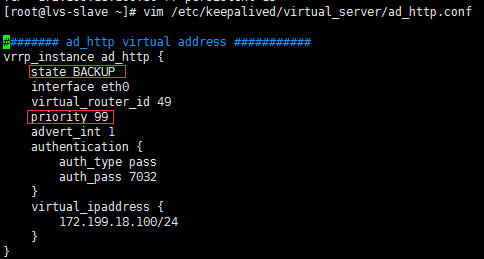


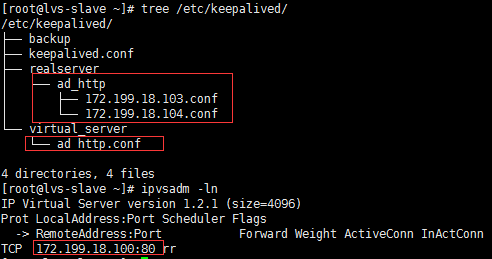
Lvs-master上查看配置





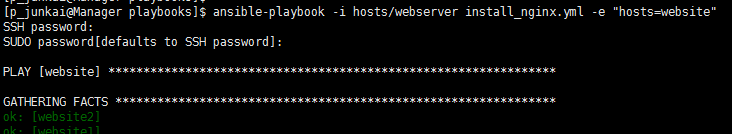
lvs-slave上查看配置



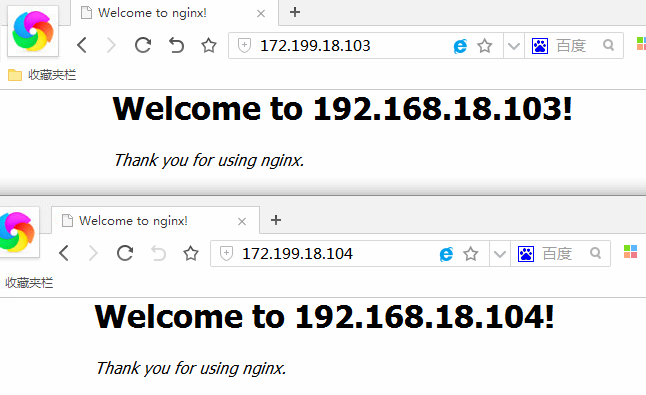


**后端服务器配置：**

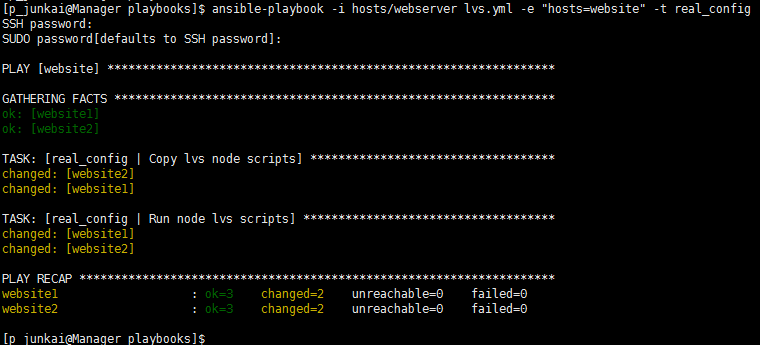
在这之前，我们先安装nginx用于测试(192是内网地址)：



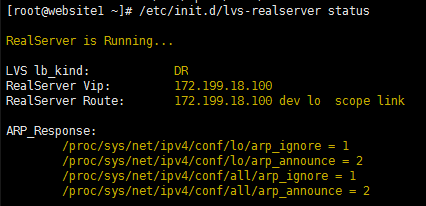
查看默认页面（192为对应内网地址）：



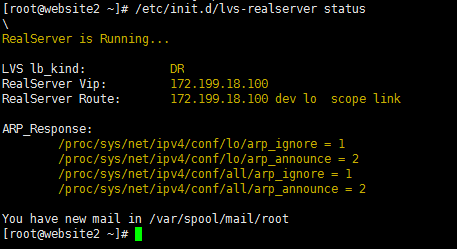
下发后端服务器脚本：



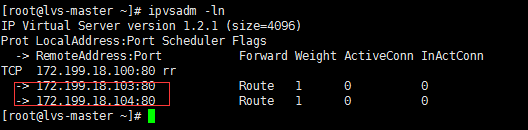
website1:



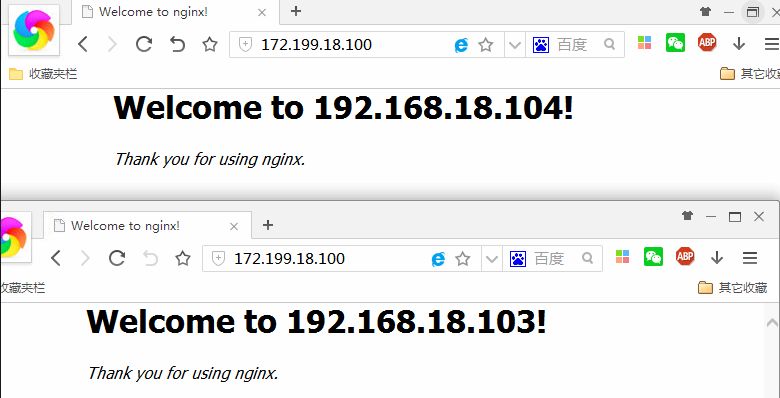
website2:



lvs-master上：

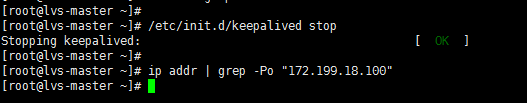


访问vip测试(可以先取消persistent\_timeout)：

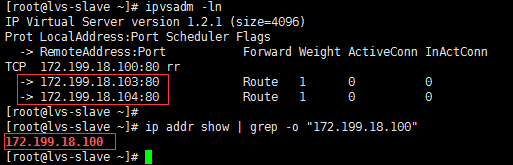


**测试故障转移**

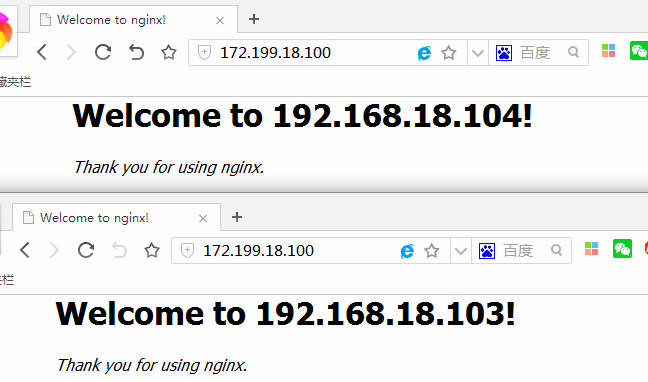
lvs-master:



lvs-slave:

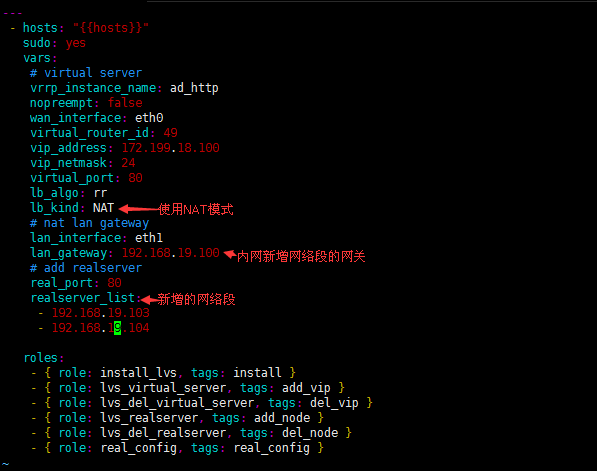


访问vip：



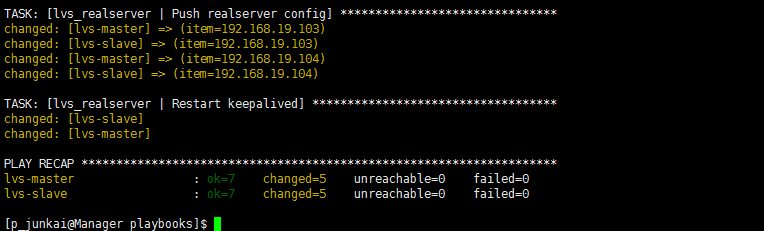
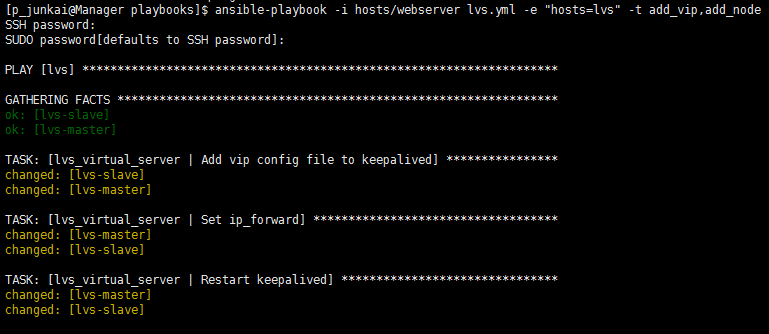
**测试NAT模式**：

playbook:

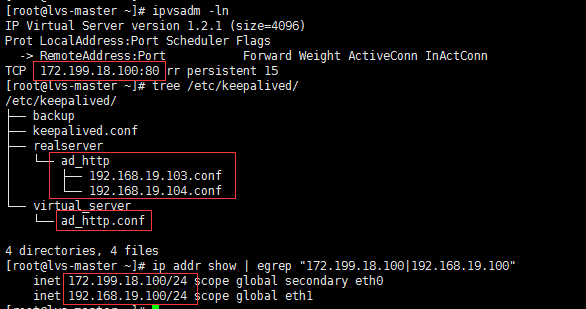


**执行：**

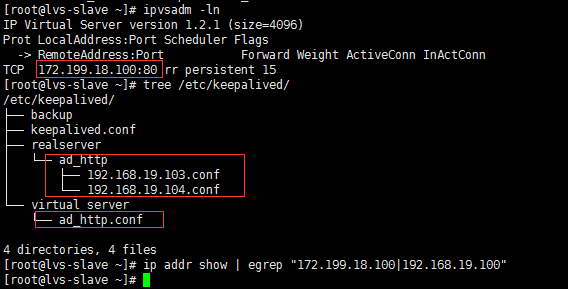
在2台lvs上：



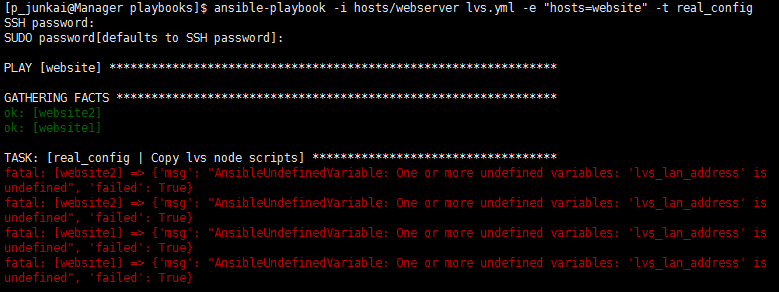
lvs-master:



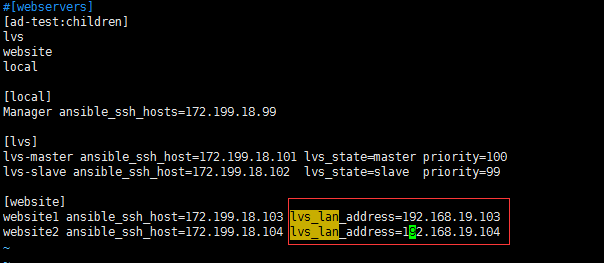
lvs-slave:



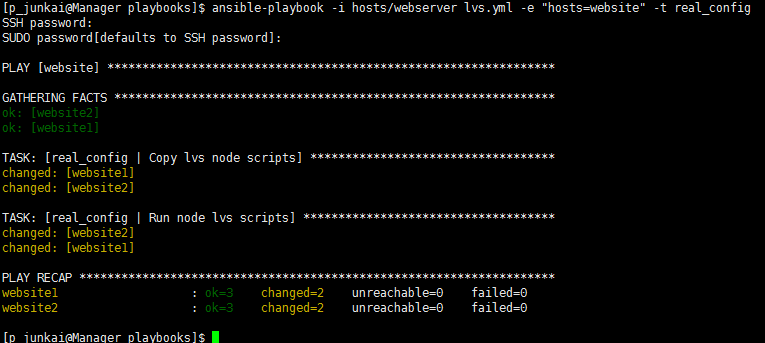
**后端服务器配置：**



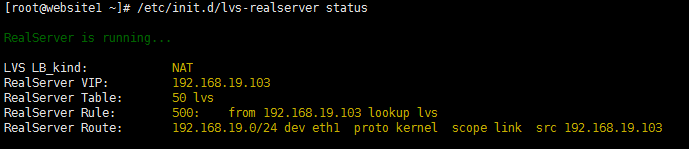
执行报错，发现lvs\_lan\_address没有定义，因为这个不同段的地址是需要手动设置的，你也可以改下脚本一台一台执行，或者在系统初始化的时候在客户机上定义facts,当然你有其他办法当然都可以，下面我们在inventory中定义：



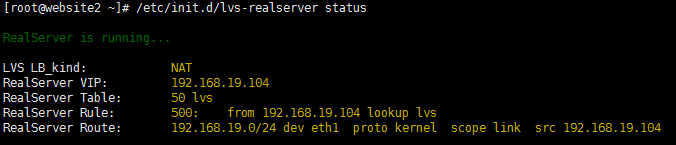
再次执行：



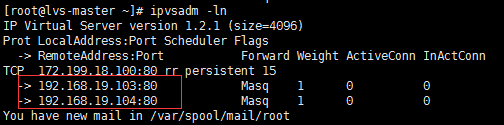
在website1上：



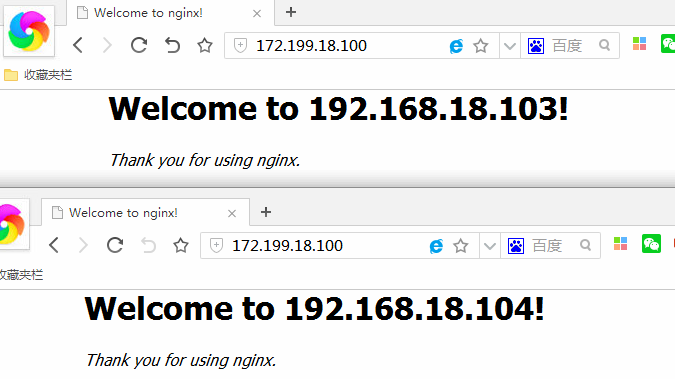
在website2上：



lvs-master上：



访问vip:

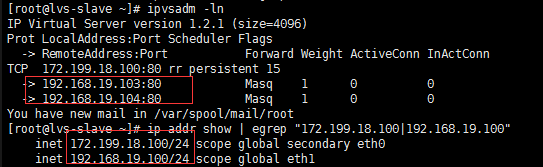


**测试故障转移：**

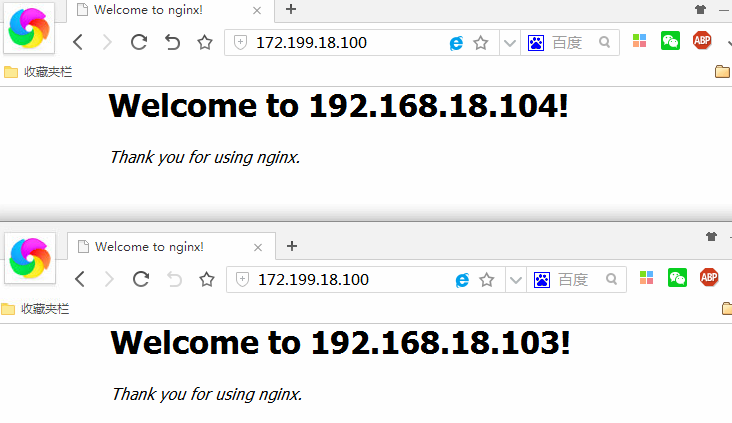
lvs-master上：



lvs-slave上：



访问测试：



**移除某个vip节点或者real server节点(略…)**