

本文档仅作为自己学习记录使用。

本例环境架构： centos7.2-1511 测试环境，均关闭防火墙以及selinux

主机名	角色	ip
master	master	192.168.4.10
minion1	haproxy keepalived	192.168.4.11
minion2	haproxy keepalived	192.168.4.12
minion3	nginx	192.168.4.13
minion3	nginx	192.168.4.14
	VIP	192.168.4.16

该配置环境主要是配置haproxy + keepalived负载均衡的高可用，其中haproxy通过轮训的方式连接到后端实际的2台nginx服务器。

### salt安装

master节点： yum install epel-release -y yum install salt-master

minion节点： yum install epel-release -y yum install salt-minion

### salt基础配置

本例不作详细介绍。

master节点上 vi /etc/salt/master 修改如下：其他均默认。

```
file_roots:
```

base: - /srv/salt/base prod: - /srv/salt/prod

```
interface: 192.168.4.10
```

minion节点上 vi /etc/salt/minion 修改如下：其他均默认。

```
master: master
```

所有master minion节点启动服务后(systemctl start salt-minion/salt-master) 在master执行 salt-key -A 接受所有 minion节点的key。相关情况不做详细介绍。

所有节点/etc/hosts 均一致

```
[root@master ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.4.10 master
192.168.4.11 minion1
192.168.4.12 minion2
192.168.4.13 minion3
192.168.4.14 minion4
```

下面配置state等不作详细介绍，直接复制粘贴。

**所有配置均在master上，首先查看tree目录**

```
[root@master ~]# cd /srv/salt/
[root@master salt]# tree
.
├── base
│   ├── init
│   │   ├── audit.sls
│   │   ├── cron.sls
│   │   ├── dns.sls
│   │   ├── env_init.sls
│   │   ├── epel.sls
│   │   ├── files
│   │   │   ├── resolv.conf
│   │   │   └── sysctl.conf
│   │   ├── history.sls
│   │   ├── sysctl.sls
│   │   └── yum.sls
│   └── top.sls
├── prod
├── cluster
│   ├── files
│   │   ├── haproxy-outside.cfg
│   │   └── haproxy-outside-keepalived.cfg
│   ├── haproxy-outside-keepalived.sls
│   └── haproxy-outside.sls
├── haproxy
│   ├── files
│   │   └── haproxy-1.8.9.tar.gz
│   └── install_haproxy.sls
├── keepalived
│   ├── files
│   │   ├── keepalived
│   │   ├── keepalived-1.4.2.tar.gz
│   │   ├── keepalived.conf
│   │   └── keepalived.sysconfig
│   └── install_keepalived.sls
├── nginx
│   ├── files
│   │   ├── nginx-1.12.2.tar.gz
│   │   └── nginx.conf
```

```

|   |   |─ nginx.init
|   |   |─ pcre-8.41.tar.gz
|   |   └─ zlib-1.2.11.tar.gz
|   └─ nginx-install.sls
|   └─ nginx-service.sls
|   └─ nginx-user.sls
|   └─ pcre-install.sls
|   └─ zlib-install.sls
└─ pkg
   └─ pkg-init.sls

```

13 directories, 33 files

## 首先介绍base/init目录下的文件

[root@master init]# tree

```

.
├─ audit.sls
├─ cron.sls
├─ dns.sls
├─ env_init.sls
├─ epel.sls
├─ files
│   └─ resolv.conf
│   └─ sysctl.conf
├─ history.sls
├─ sysctl.sls
└─ yum.sls

```

1 directory, 10 files

该目录文件为所有节点配置初始化的一些配置，比方说统一dns，统一安装epel源 统一sysctl参数等等。其中env\_init.sls 是统一调配入口，这样只需要运行env\_init就可以自动运行其他所有配置文件。可自行增加编辑。

[root@master init]# cat env\_init.sls

include:

- init.audit
- init.cron
- init.dns
- init.epel
- init.history
- init.sysctl
- init.yum

[root@master init]# cat audit.sls

/etc/bashrc:

file.append:

- text:
  - export PROMPT\_COMMAND='{ msg=\$(history 1 | { read x y; echo \$y; });logger "[euid=\$(whoami)]":"\$(who am i):[`pwd`]"\$msg; }'

```
[root@master init]# cat cron.sls
ntpdate-install:
  pkg.installed:
    - name: ntpdate

set-crontab:
  cron.present:
    - name: /usr/sbin/ntpdate time1.aliyun.com >> /dev/null 2>&1
    - user: root
    - minute: "*2"
    - require:
      - pkg: ntpdate-install
```

```
[root@master init]# cat dns.sls
/etc/resolv.conf:
  file.managed:
    - source: salt://init/files/resolv.conf
    - user: root
    - group: root
    - mode: 644
```

```
[root@master init]# cat epel.sls
yum_epel:
  pkg.installed:
    - name: epel-release
    - unless: rpm -qa |grep epel-release
```

```
[root@master init]# cat history.sls
/etc/profile:
  file.append:
    - text:
      - export HISTTIMEFORMAT="%F %T `whoami` "
```

```
[root@master init]# cat sysctl.sls
/etc/sysctl.conf:
  file.managed:
    - source: salt://init/files/sysctl.conf
    - user: root
    - group: root
    - mode: 644
```

```
[root@master init]# cat yum.sls
yum_base:
  pkg.installed:
    - names:
      - gcc
      - gcc-c++
      - make
      - autoconf
      - net-tools
      - lrzsz
      - sysstat
```

- vim-enhanced
- openssh-clients
- lsof
- tree
- wget
- cmake

该目录下file目录

```
[root@master init]# tree files/
files/
├── resolv.conf
└── sysctl.conf

0 directories, 2 files

[root@master init]# cd files/
[root@master files]# ll
total 8
-rw-r--r-- 1 root root 53 Jun 6 11:37 resolv.conf
-rw-r--r-- 1 root root 449 Jun 6 11:57 sysctl.conf

[root@master files]# cat resolv.conf
# Generated by NetworkManager #根据实际情况填写
nameserver 192.168.0.1

[root@master files]# cat sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
#本例为空，测试环境不想调试内核参数，若实际应用中，请自行输入需要调整的内核参数
```

## 介绍prod目录

该目录为实际的安装包以及配置等目录。首先查看tree

每个目录均为一个需要安装的软件包以及其配置文件。cluster目录是后期在生成环境下结合不同环境配置haproxy和keepalived的配置文档，最后介绍。其他的目录比如nginx haproxy等都是安装配置。

```
[root@master prod]# tree
.
├── cluster
│   ├── files
│   │   ├── haproxy-outside.cfg
│   │   └── haproxy-outside-keepalived.cfg
```

```
|   ├── haproxy-outside-keepalived.sls
|   └── haproxy-outside.sls
|   └── haproxy
|       ├── files
|       │   └── haproxy-1.8.9.tar.gz
|       └── install_haproxy.sls
|   └── keepalived
|       ├── files
|       │   ├── keepalived
|       │   ├── keepalived-1.4.2.tar.gz
|       │   ├── keepalived.conf
|       │   └── keepalived.sysconfig
|       └── install_keepalived.sls
|   └── nginx
|       ├── files
|       │   ├── nginx-1.12.2.tar.gz
|       │   ├── nginx.conf
|       │   ├── nginx.init
|       │   ├── pcre-8.41.tar.gz
|       │   └── zlib-1.2.11.tar.gz
|       ├── nginx-install.sls
|       ├── nginx-service.sls
|       ├── nginx-user.sls
|       ├── pcre-install.sls
|       └── zlib-install.sls
|   └── pkg
|       └── pkg-init.sls
```

9 directories, 22 files

## 首先看pkg目录

这个目录是所有节点部署nginx haproxy keepalived等软件需要的依赖包

```
[root@master prod]# cd pkg/
[root@master pkg]# ll
total 4
-rw-r--r-- 1 root root 167 Jun  6 14:07 pkg-init.sls
[root@master pkg]# cat pkg-init.sls
pkg-init:
  pkg.installed:
    - names:
      - gcc
      - gcc-c++
      - glibc
      - make
      - autoconf
      - openssl
      - openssl-devel
      - automake
```

## 其次haproxy目录

```
[root@master haproxy]# pwd
/srv/salt/prod/haproxy
[root@master haproxy]# tree
.
├── files
│   └── haproxy-1.8.9.tar.gz
└── install_haproxy.sls
```

1 directory, 2 files

file目录下为haproxy安装源码包

```
[root@master haproxy]# cd files/
[root@master files]# ll
total 2012
-rw-r--r-- 1 root root 2057051 Jun  6 14:15 haproxy-1.8.9.tar.gz
```

安装配置文件

```
[root@master haproxy]# cat install_haproxy.sls
include:
  - pkg.pkg-init
```

haproxy-install:

file.managed:

- name: /usr/local/src/haproxy-1.8.9.tar.gz
- source: salt://haproxy/files/haproxy-1.8.9.tar.gz
- user: root
- group: root
- mode: 755

cmd.run:

- name: cd /usr/local/src && tar xf haproxy-1.8.9.tar.gz && cd haproxy-1.8.9 && make TARGET=linux2628 PREFIX=/usr/local/haproxy && make install PREFIX=/usr/local/haproxy && sed -i 's?BIN=/usr/sbin/\$BASENAME?BIN=/usr/local/haproxy/sbin/\$BASENAME?' /usr/local/src/haproxy-1.8.9/examples/haproxy.init && sed -i '/NETWORKING/c [[ \$NETWORKING = "no" ]]' && exit 0' /usr/local/src/haproxy-1.8.9/examples/haproxy.init && cp /usr/local/src/haproxy-1.8.9/examples/haproxy.init /etc/init.d/haproxy && chmod +x /etc/init.d/haproxy
- unless: test -d /usr/local/haproxy
- require:
  - pkg: pkg-init
  - file: haproxy-install

haproxy\_chkconfig:

cmd.run:

- name: chkconfig --add haproxy && chkconfig --level 2345 haproxy on
- unless: chkconfig --list |grep haproxy
- require:
  - file: haproxy-install

haproxy-config-dir:

file.directory:

- name: /etc/haproxy
- user: root

```
- group: root
- mode: 755

net.ipv4.ip_nonlocal_bind:
  cmd.run:
    - name: echo "net.ipv4.ip_nonlocal_bind=1" >> /etc/sysctl.conf && sysctl -p
    - unless: cat /etc/sysctl.conf | grep net.ipv4.ip_nonlocal_bind
    - require:
      - file: haproxy-install
```

## keepalived目录

```
[root@master prod]# cd keepalived/
[root@master keepalived]# ll
total 4
drwxr-xr-x 2 root root 102 Jun  7 11:03 files
-rw-r--r-- 1 root root 1452 Jun  7 11:18 install_keepalived.sls
```

### 首先查看files目录

```
[root@master files]# ll
total 736
-rwxr-xr-x 1 root root 1335 Jun  7 11:01 keepalived
-rw-r--r-- 1 root root 738096 Feb 26 00:48 keepalived-1.4.2.tar.gz
-rw-r--r-- 1 root root 3550 Jun  7 11:02 keepalived.conf
-rw-r--r-- 1 root root 667 Jun  7 11:02 keepalived.sysconfig
```

keepalived文件为keepalived的service启动服务文件，在/etc/init.d/目录下，keepalived.conf 为其基础配置文件，keepalived.sysconfig为启动文件需要的配置文件。

```
[root@master files]# cat keepalived
#!/bin/sh
#
# Startup script for the Keepalived daemon
#
# processname: keepalived
# pidfile: /var/run/keepalived.pid
# config: /etc/keepalived/keepalived.conf
# chkconfig: - 21 79
# description: Start and stop Keepalived

# Source function library
. /etc/rc.d/init.d/functions

# Source configuration file (we set KEEPALIVED_OPTIONS there)
. /etc/sysconfig/keepalived

RETVAL=0

prog="keepalived"
```



```

start() {
    echo -n $"Starting $prog: "
    daemon /usr/local/keepalived/sbin/keepalived ${KEEPALIVED_OPTIONS}
##上面参数是修改之后的，默认的为/sbin/keepalived ${KEEPALIVED_OPTIONS}
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$prog
}

stop() {
    echo -n $"Stopping $prog: "
    killproc keepalived
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog
}

reload() {
    echo -n $"Reloading $prog: "
    killproc keepalived -1
    RETVAL=$?
    echo
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    reload)
        reload
        ;;
    restart)
        stop
        start
        ;;
    condrestart)
        if [ -f /var/lock/subsys/$prog ]; then
            stop
            start
        fi
        ;;
    status)
        status keepalived
        RETVAL=$?
        ;;
    *)
        echo "Usage: $0 {start|stop|reload|restart|condrestart|status}"
        RETVAL=1

```

```
esac
```

```
exit $RETVAL
```

```
[root@master files]# cat keepalived.conf
```

##该文件为默认文件，放这里是为了启动过程中有个初始默认文件，后期结合实际生产环境会被修改的，在cluster目录中介绍。

! Configuration File for keepalived

```
global_defs {
    notification_email {
        acassen@firewall.loc
        failover@firewall.loc
        sysadmin@firewall.loc
    }
    notification_email_from Alexandre.Cassen@firewall.loc
    smtp_server 192.168.200.1
    smtp_connect_timeout 30
    router_id LVS_DEVEL
    vrrp_skip_check_adv_addr
    vrrp_strict
    vrrp_garp_interval 0
    vrrp_gna_interval 0
}
```

```
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.200.16
        192.168.200.17
        192.168.200.18
    }
}
```

```
virtual_server 192.168.200.100 443 {
    delay_loop 6
    lb_algo rr
    lb_kind NAT
    persistence_timeout 50
    protocol TCP

    real_server 192.168.201.100 443 {
        weight 1
        SSL_GET {
```

```

        url {
            path /
            digest ff20ad2481f97b1754ef3e12ecd3a9cc
        }
        url {
            path /mrtg/
            digest 9b3a0c85a887a256d6939da88aabd8cd
        }
        connect_timeout 3
        retry 3
        delay_before_retry 3
    }
}

```

```

virtual_server 10.10.10.2 1358 {
    delay_loop 6
    lb_algo rr
    lb_kind NAT
    persistence_timeout 50
    protocol TCP

```

```

    sorry_server 192.168.200.200 1358

```

```

    real_server 192.168.200.2 1358 {
        weight 1
        HTTP_GET {
            url {
                path /testurl/test.jsp
                digest 640205b7b0fc66c1ea91c463fac6334d
            }
            url {
                path /testurl2/test.jsp
                digest 640205b7b0fc66c1ea91c463fac6334d
            }
            url {
                path /testurl3/test.jsp
                digest 640205b7b0fc66c1ea91c463fac6334d
            }
            connect_timeout 3
            retry 3
            delay_before_retry 3
        }
    }
}

```

```

    real_server 192.168.200.3 1358 {
        weight 1
        HTTP_GET {
            url {
                path /testurl/test.jsp
                digest 640205b7b0fc66c1ea91c463fac6334c
            }
            url {

```

```

        path /testurl2/test.jsp
        digest 640205b7b0fc66c1ea91c463fac6334c
    }
    connect_timeout 3
    retry 3
    delay_before_retry 3
}
}

virtual_server 10.10.10.3 1358 {
    delay_loop 3
    lb_algo rr
    lb_kind NAT
    persistence_timeout 50
    protocol TCP

    real_server 192.168.200.4 1358 {
        weight 1
        HTTP_GET {
            url {
                path /testurl/test.jsp
                digest 640205b7b0fc66c1ea91c463fac6334d
            }
            url {
                path /testurl2/test.jsp
                digest 640205b7b0fc66c1ea91c463fac6334d
            }
            url {
                path /testurl3/test.jsp
                digest 640205b7b0fc66c1ea91c463fac6334d
            }
            connect_timeout 3
            retry 3
            delay_before_retry 3
        }
    }
}

real_server 192.168.200.5 1358 {
    weight 1
    HTTP_GET {
        url {
            path /testurl/test.jsp
            digest 640205b7b0fc66c1ea91c463fac6334d
        }
        url {
            path /testurl2/test.jsp
            digest 640205b7b0fc66c1ea91c463fac6334d
        }
        url {
            path /testurl3/test.jsp
            digest 640205b7b0fc66c1ea91c463fac6334d
        }
    }
}

```

```

        connect_timeout 3
        retry 3
        delay_before_retry 3
    }
}
}

```

```

[root@master files]# cat keepalived.sysconfig
#默认文件, 在解压之后的安装包里面
# Options for keepalived. See `keepalived --help' output and keepalived(8) and
# keepalived.conf(5) man pages for a list of all options. Here are the most
# common ones :
#
# --vrrp                -P    Only run with VRRP subsystem.
# --check                -C    Only run with Health-checker subsystem.
# --dont-release-vrrp    -V    Dont remove VRRP VIPs & VROUTEs on daemon stop.
# --dont-release-ipvs    -I    Dont remove IPVS topology on daemon stop.
# --dump-conf            -d    Dump the configuration data.
# --log-detail           -D    Detailed log messages.
# --log-facility         -S    0-7 Set local syslog facility (default=LOG_DAEMON)
#
KEEPALIVED_OPTIONS="-D"

```

## 查看keepalived的salt配置文档

```

[root@master keepalived]# ll
total 4
drwxr-xr-x 2 root root 102 Jun  7 11:03 files
-rw-r--r-- 1 root root 1452 Jun  7 11:18 install_keepalived.sls

[root@master keepalived]# cat install_keepalived.sls
include:
  - pkg.pkg-init
dependency_package_install:
  pkg.installed:
    - names:
      - libnl3-devel
      - libnfnetlink-devel

keepalived-install:
  file.managed:
    - name: /usr/local/src/keepalived-1.4.2.tar.gz
    - source: salt://keepalived/files/keepalived-1.4.2.tar.gz
    - user: root
    - group: root
    - mode: 755
  cmd.run:
    - name: cd /usr/local/src && tar -xf keepalived-1.4.2.tar.gz && cd keepalived-1.4.2 &&
      ./configure --prefix=/usr/local/keepalived && make && make install
    - unless: test -d /usr/local/keepalived

```

```

- require:
  - pkg: pkg-init
  - pkg: dependency_package_install
  - file: keepalived-install

keepalived-init:
  file.managed:
    - name: /etc/init.d/keepalived
    - source: salt://keepalived/files/keepalived
    - user: root
    - group: root
    - mode: 755
  cmd.run:
    - name: chkconfig --add keepalived && chkconfig --level 2345 keepalived on
    - unless: chkconfig --list | grep keepalived
    - require:
      - file: keepalived-init

/etc/sysconfig/keepalived:
  file.managed:
    - source: salt://keepalived/files/keepalived.sysconfig
    - user: root
    - group: root
    - mode: 644

/etc/keepalived:
  file.directory:
    - user: root
    - group: root
    - mode: 755

/etc/keepalived/keepalived.conf:
  file.managed:
    - source: salt://keepalived/files/keepalived.conf
    - user: root
    - group: root
    - mode: 644
    - require:
      - file: /etc/keepalived

```

## nginx目录

```

[root@master nginx]# tree
.
├── files
│   ├── nginx-1.12.2.tar.gz
│   ├── nginx.conf
│   ├── nginx.init
│   ├── pcre-8.41.tar.gz
│   └── zlib-1.2.11.tar.gz
├── nginx-install.sls
└── nginx-service.sls

```

```
|— nginx-user.sls
|— pcre-install.sls
└─ zlib-install.sls
```

1 directory, 10 files

file目录中为nginx的源码包以及需要的依赖包pcre和zlib的源码包。nginx.conf为nginx的配置文件，nginx.init为启动脚本 既/etc/init.d目录下的service控制服务脚本。

nginx-install.sls 为nginx的安装脚本 nginx-service.sls启动nginx服务脚本 nginx-user.sls 为创建nginx用户脚本 pcre-install.sls zlib-install.sls 分别为安装pcre和zlib的脚本。

files目录下：

```
[root@master files]# cat nginx.conf
user  nginx;
worker_processes  auto;
error_log  logs/error.log  error;
worker_rlimit_nofile 30000;
pid        /var/run/nginx.pid;

events {
    use epoll;
    worker_connections 65535;
}

http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile      on;
    tcp_nopush    on;
    underscores_in_headers on;
    keepalive_timeout 10;
    send_timeout  60;
    gzip on;
    include /usr/local/nginx/conf/vhost/*.conf;

    server {

        listen      80;
        root /usr/local/nginx/html;
        index index.html;
        server_name 127.0.0.1;

        location /nginx_status {
            stub_status on;
            access_log off;
            allow 127.0.0.1;
            deny all;
        }

    }
}
```

```
[root@master files]# cat nginx.init
#!/bin/sh
```

```

#
# nginx - this script starts and stops the nginx daemon
#
# chkconfig:   - 85 15
# description:  Nginx is an HTTP(S) server, HTTP(S) reverse \
#               proxy and IMAP/POP3 proxy server
# processname: nginx
# config:      /etc/nginx/nginx.conf
# config:      /etc/sysconfig/nginx
pidfile:      /var/run/nginx.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/local/nginx/sbin/nginx"
prog=$(basename $nginx)
##指定nginx的配置文件目录
NGINX_CONF_FILE="/usr/local/nginx/conf/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

make_dirs() {
    # make required directories
    user=`$nginx -V 2>&1 | grep "configure arguments:" | sed 's/^[^]*--user=\([^ ]*\).*\/\1/g' -`
    if [ -z "`grep $user /etc/passwd`" ]; then
        useradd -M -s /bin/nologin $user
    fi
    options=`$nginx -V 2>&1 | grep 'configure arguments:'`
    for opt in $options; do
        if [ `echo $opt | grep '.*-temp-path'` ]; then
            value=`echo $opt | cut -d "=" -f 2`
            if [ ! -d "$value" ]; then
                # echo "creating" $value
                mkdir -p $value && chown -R $user $value
            fi
        fi
    done
}

start() {
    [ -x $nginx ] || exit 5
    [ -f $NGINX_CONF_FILE ] || exit 6
    make_dirs
    echo -n $"Starting $prog: "
    daemon $nginx -c $NGINX_CONF_FILE
}

```



```

    retval=$?
    echo
    [ $retval -eq 0 ] && touch $lockfile
    return $retval
}

stop() {
    echo -n "Stopping $prog: "
    killproc $prog -QUIT
    retval=$?
    echo
    [ $retval -eq 0 ] && rm -f $lockfile
    return $retval
}

restart() {
    configtest || return $?
    stop
    sleep 1
    start
}

reload() {
    configtest || return $?
    echo -n "Reloading $prog: "
    $nginx -s reload
    RETVAL=$?
    echo
}

force_reload() {
    restart
}

configtest() {
    $nginx -t -c $NGINX_CONF_FILE
}

rh_status() {
    status $prog
}

rh_status_q() {
    rh_status >/dev/null 2>&1
}

case "$1" in
    start)
        rh_status_q && exit 0
        $1
        ;;
    stop)
        rh_status_q || exit 0

```

```

        $1
        ;;
restart|configtest)
        $1
        ;;
reload)
        rh_status_q || exit 7
        $1
        ;;
force-reload)
        force_reload
        ;;
status)
        rh_status
        ;;
condrestart|try-restart)
        rh_status_q || exit 0
        ;;
*)
        echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-
reload|configtest}"
        exit 2
esac

```

查看其他配置文件nginx目录下，其他安装包安装配置文件

```

nginx安装配置
[root@master nginx]# cat nginx-install.sls
include:
  - pkg.pkg-init
  - nginx.nginx-user
  - nginx.pcre-install
  - nginx.zlib-install

/var/cache/nginx:
file.directory:
  - user: nginx
  - group: nginx
  - mode: 755
  - makedirs: True

nginx_dependence:
pkg.installed:
  - names:
    - gd
    - gd-devel

nginx-source-install:
file.managed:
  - name: /usr/local/src/nginx-1.12.2.tar.gz
  - source: salt://nginx/files/nginx-1.12.2.tar.gz
  - user: root

```

```

- group: root
- mode: 755
cmd.run:
- name: cd /usr/local/src && tar xf nginx-1.12.2.tar.gz && cd nginx-1.12.2 && ./configure --
prefix=/usr/local/nginx --lock-path=/var/run/nginx.lock --http-client-body-temp-
path=/var/cache/nginx/client_temp --http-proxy-temp-path=/var/cache/nginx/proxy_temp --http-
fastcgi-temp-path=/var/cache/nginx/fastcgi_temp --user=nginx --group=nginx --with-file-aio --
with-threads --with-http_addition_module --with-http_auth_request_module --with-http_flv_module
--with-http_gunzip_module --with-http_gzip_static_module --with-http_mp4_module --with-
http_realip_module --with-http_secure_link_module --with-http_ssl_module --with-
http_stub_status_module --with-http_sub_module --with-http_v2_module --with-stream --with-
stream_ssl_module --with-http_image_filter_module --with-pcre=/usr/local/src/pcre-8.41 --with-
zlib=/usr/local/src/zlib-1.2.11 && make && make install
- unless: test -d /usr/local/nginx
- require:
- file: nginx-source-install
- pkg: pkg-init
- cmd: pcre-source-install
- cmd: zlib-source-install
- user: nginx-user-group

```

## 启动服务配置

```

[root@master nginx]# cat nginx-service.sls
include:
- nginx.nginx-install

nginx-init:
  file.managed:
    - name: /etc/init.d/nginx
    - source: salt://nginx/files/nginx.init
    - user: root
    - group: root
    - mode: 755
  cmd.run:
    - name: chkconfig --add nginx && chkconfig --level 2345 nginx on
    - unless: chkconfig --list | grep nginx
    - require:
      - file: nginx-init

/usr/local/nginx/conf/nginx.conf:
  file.managed:
    - source: salt://nginx/files/nginx.conf
    - user: nginx
    - group: nginx
    - mode: 644

nginx-vhost:
  file.directory:
    - name: /usr/local/nginx/conf/vhost
    - require:
      - cmd: nginx-source-install

```

```
service.running:
  - name: nginx
  - enable: True
  - reload: True
  - require:
    - cmd: nginx-init
  - watch:
    - file: /usr/local/nginx/conf/nginx.conf
```

#### 创建nginx user配置

```
[root@master nginx]# cat nginx-user.sls
```

```
nginx-user-group:
```

```
  group.present:
```

- name: nginx
- gid: 1010

```
  user.present:
```

- name: nginx
- fullname: nginx
- shell: /sbin/nologin
- uid: 1010
- gid: 1010

#### pcre源码包安装配置

```
[root@master nginx]# cat pcre-install.sls
```

```
include:
```

- pkg.pkg-init

```
pcre-source-install:
```

```
  file.managed:
```

- name: /usr/local/src/pcre-8.41.tar.gz
- source: salt://nginx/files/pcre-8.41.tar.gz
- user: root
- group: root
- mode: 755

```
  cmd.run:
```

- name: cd /usr/local/src && tar xf pcre-8.41.tar.gz && cd pcre-8.41 && ./configure --prefix=/usr/local/pcre && make && make install
- unless: test -d /usr/local/pcre
- require:
- file: pcre-source-install

#### zlib安装包安装配置

```
[root@master nginx]# cat zlib-install.sls
```

```
include:
```

- pkg.pkg-init

```
zlib-source-install:
```

```
  file.managed:
```

- name: /usr/local/src/zlib-1.2.11.tar.gz
- source: salt://nginx/files/zlib-1.2.11.tar.gz
- user: root
- group: root

```

- mode: 755
cmd.run:
- name: cd /usr/local/src && tar xf zlib-1.2.11.tar.gz && cd zlib-1.2.11 && ./configure --
prefix=/usr/local/zlib && make && make install
- unless: test -d /usr/local/zlib
- require:
- file: zlib-source-install

```

以上所有配置结合top.sls文件后都能安装配置成功。下面结合测试环境增加并修改haproxy keepalived配置 实现nginx服务的负载均衡以及高可用。

## cluster目录

```

[root@master cluster]# tree
.
├── files
│   ├── haproxy-outside.cfg
│   └── haproxy-outside-keepalived.cfg
├── haproxy-outside-keepalived.sls
└── haproxy-outside.sls

1 directory, 4 files

```

首先介绍2个sls文件 为salt的配置文件，haproxy-outside.sls为配置haproxy，haproxy-outside-keepalived.sls为配置haproxy的keepalived的配置。files目录里面分别为haproxy keepalived的配置文件。可结合实际生产环境进行修改调整。

```

#修改haproxy配置文件并启动服务
[root@master cluster]# cat haproxy-outside.sls
include:
- haproxy.install_haproxy

haproxy-service:
  file.managed:
    - name: /etc/haproxy/haproxy.cfg
    - source: salt://cluster/files/haproxy-outside.cfg
    - user: root
    - group: root
    - mode: 644

  service.running:
    - name: haproxy
    - enable: True
    - reload: True
    - require:
      - cmd: haproxy-install
    - watch:
      - file: haproxy-service

```

#修改keepalived配置文件并启动服务。注意这里用到了jinja模块，对多后端通过变量进行设置参数。这里因为2个keepalived配置文件需要的master backup priority等值不一样。通过变量指定。

```

[root@master cluster]# cat haproxy-outside-keepalived.sls
include:
  - keepalived.install_keepalived
keepalived-service:
  file.managed:
    - name: /etc/keepalived/keepalived.conf
    - source: salt://cluster/files/haproxy-outside-keepalived.cfg
    - user: root
    - group: root
    - mode: 644
    - template: jinja
    {% set STATEID = ["MASTER","BACKUP"] %}
    {% set PRIORITYID = [120,100] %}
    {% if grains['fqdn'] == 'minion1' %}
    - ROUTEID: minion1
    - STATEID: {{ STATEID[0] }}
    - PRIORITYID: {{ PRIORITYID[0] }}
    {% elif grains['fqdn'] == 'minion2' %}
    - ROUTEID: minion2
    - STATEID: {{ STATEID[1] }}
    - PRIORITYID: {{ PRIORITYID[1] }}
    {% endif %}
  service.running:
    - name: keepalived
    - enable: True
    - watch:
      - file: keepalived-service

####haproxy的配置文件
[root@master files]# pwd
/srv/salt/prod/cluster/files
[root@master files]# ll
total 8
-rw-r--r-- 1 root root 1296 Jun  7 16:47 haproxy-outside.cfg
-rw-r--r-- 1 root root  375 Jun  8 12:22 haproxy-outside-keepalived.cfg
[root@master files]# cat haproxy-outside.cfg
global
    log          127.0.0.1 local2
    chroot       /usr/local/haproxy
    pidfile      /usr/local/haproxy/haproxy.pid
    maxconn      10000
    daemon
    nbproc 1

defaults
    option http-keep-alive
    maxconn 10000
    mode      http
    log                          global
    option                          httplog
    timeout http-request 10s
    timeout queue 1m
    timeout connect 10s
    timeout client 1m

```

```

    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
#####通过haproxy节点8888端口/haproxy-status 查看haproxy状态
listen status
    mode http
    bind *:8888
    stats enable
    stats hide-version
    stats uri /haproxy-status
    stats auth haproxy:saltstack
    stats admin if TRUE
    stats realm Haproxy\ Statistics
#####前端绑定VIP指向后端default_backend nginx
frontend web
    bind 192.168.4.16:80
    mode http
    option httplog
    log global
    default_backend nginx

#####定义nginx后端的2台实际nginx物理机节点
backend nginx
    option forwardfor header X-REAL-IP
    option httpchk HEAD / HTTP/1.0
    balance roundrobin
    server minion3 192.168.4.13:80 check inter 2000 rise 30 fall 15
    server minion4 192.168.4.14:80 check inter 2000 rise 30 fall 15

```

```

###keepalived的配置文件，引用了之前文件haproxy-outside-keepalived.sls变量
[root@master files]# cat haproxy-outside-keepalived.cfg
global_defs {
    router_id {{ROUTEID}}

}

vrrp_instance haproxy_ha {
    state {{STATEID}}
    interface eno16777736
    virtual_router_id 36
    priority {{PRIORITYID}}
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.4.16
    }
}

```

这里所有配置均已介绍完毕。下面开始统一部署测试。回到base目录下，编写top.sls文件

```
[root@master base]# cat top.sls
#base:定义*既所有的主机执行init目录下的env_init.sls文件即节点初始化的配置
base:
  '*':
    - init.env_init
##prod 定义了不同的minion节点需要执行的步骤，此例中minion1 minion2需要安装haproxy keepalived 以及配置
高可用以及负载均衡。 minion3 minion4节点只需要安装nginx而已。
prod:
  'minion1':
    - haproxy.install_haproxy
    - keepalived.install_keepalived
    - cluster.haproxy-outside
    - cluster.haproxy-outside-keepalived
  'minion2':
    - haproxy.install_haproxy
    - keepalived.install_keepalived
    - cluster.haproxy-outside
    - cluster.haproxy-outside-keepalived
  'minion3':
    - nginx.nginx-service
  'minion4':
    - nginx.nginx-service
```

运行脚本，部署该例环境

```
[root@master base]# salt '*' state.highstate
```

此例必须返回所有成功。本例测试环境中均已调试OK 执行OK。

下面查看运行完成之后的效果，这里修改后端nginx minion3 minion4的首页配置文件

```
[root@minion3 html]# cat /usr/local/nginx/html/index.html
minion3
[root@minion4 ~]# cat /usr/local/nginx/html/index.html
minion4
```

浏览器上登入192.168.4.11:8888/haproxy-status 192.168.4.12:8888/haproxy-status 以及VIP查看haproxy状态，用户名密码为之前配置文件中定义的haproxy/saltstack

minion1登入haproxy查看





192.168.4.16:8888/haproxy-status

HAProxy

Statistics Report for pid 37791

> General process information

pid = 37791 (process #1, nbproc = 1, nbthread = 1)  
uptime = 0d 2h54m15s  
system limits: memmax = unlimited; ulimit-n = 20014  
maxsock = 20014; maxconn = 10000; maxpipes = 0  
current conns = 1; current pipes = 0/0; conn rate = 1/sec  
Running tasks: 1/7, idle = 100 %

active UP  
active UP, going down  
active DOWN, going up  
active or backup DOWN  
active or backup DOWN for maintenance (MAINT)  
active or backup SOFT STOPPED for maintenance

backup UP  
backup UP, going down  
backup DOWN, going up  
not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:  
Scope:   
Hide "DOWN" servers  
Refresh now  
CSV export

External resources:  
Primary site  
Updates (v1.8)  
Online manual

status

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle	
Frontend				1	2	-	1	1		10 000	9		3 688	82 184	0	0	0					OPEN									
Backend	0	0		0	1		0	1	1 000	4	0	0s	3 688	82 184	0	0		4	0	0	0	2h54m UP			0	0	0		0		

web

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle	
Frontend				0	2	-	0	1		10 000	7		3 334	1 659	0	0	0					OPEN									

nginx

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle
<input type="checkbox"/> minion3	0	0	-	0	1		0	1	-	4	4	2h51m	1 894	948	0	0	0	0	0	0	0	2h53m UP	L7OK/200 in 0ms	1	Y	-	1	1	0s	-
<input type="checkbox"/> minion4	0	0	-	0	1		0	1	-	3	3	2h51m	1 440	711	0	0	0	0	0	0	0	2h52m UP	L7OK/200 in 0ms	1	Y	-	1	1	0s	-
Backend	0	0		0	2		0	1	1 000	7	7	2h51m	3 334	1 659	0	0		0	0	0	0	2h53m UP		2	2	0		1	1m8s	

Choose the action to perform on the checked servers :  Apply

minion3节点登入nginx

192.168.4.13

minion3

minion4节点登入nginx

192.168.4.14

minion4

VIP登入nginx 并刷新浏览器

192.168.4.16

minion4

192.168.4.16

minion3

可以看到测试效果已经实现了。