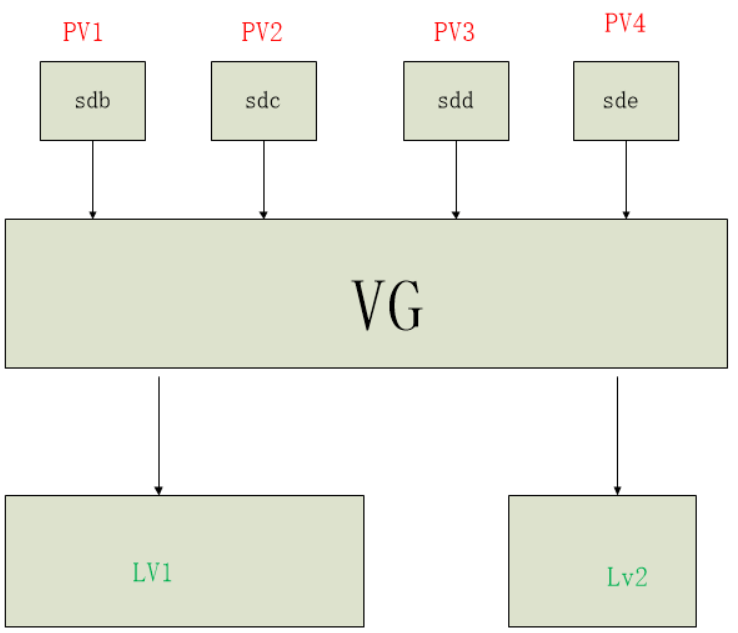
# 服务器环境搭建

## LVM磁盘挂载

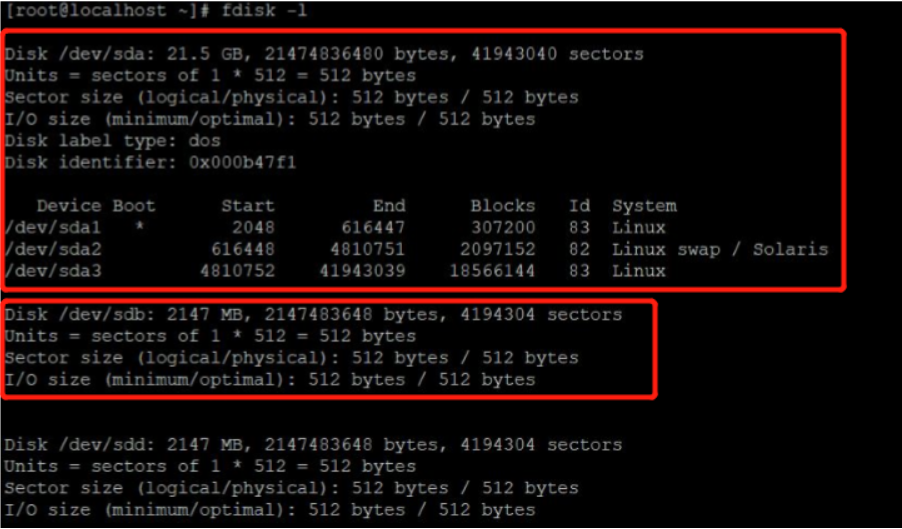
磁盘分区分为直接挂载和LVM挂载,使用工具:fdisk 两种分区思路都是:分区>分区格式化>挂载分区>开机自启动;整块磁盘可以分四个主分区:1-4;下面以LVM磁盘挂载介绍:



### 挂载

#### 磁盘格式化

查看服务器磁盘:fdisk -l ;下图中第一块磁盘sda被分为三个区,第二块磁盘sdb未分区:



磁盘分区:fdisk /dev/sdb;命令执行后,进入分区界面,



1 输入按键:n,接下来一直回车

2 输入按键t:修改磁盘属性,输入8e修改磁盘属性为LVM

3 输入w:保存磁盘修改并退出

4 再次查看磁盘文件:fdisk -l

5 磁盘文件系统格式化,推荐使用xfs文件系统

mke2fs -t ext4 /dev/sdb14

备注:磁盘空间大于2T,无法使用fdis命令格式化磁盘,推荐使用parted命令

#### 创建物理卷PV:

1 创建物理卷pv:pvcreate /dev/sdb1

2 创建成功提示:Physical volume "\*\*\*\*\*\*\*\*\*\*\*\*" successfully created.

3 查看PV:pvs或pvdisplay

4 删除物理卷:pvremove

#### **创建物理卷组VG:**

1 vgcreate sky\_vg /dev/sdb1

2 创建一个名为sky\_vg的卷组,并将物理卷/dev/sdb1加入到这个卷组

3 查看卷组:vgs 或 vgdisplay

#### **创建逻辑卷组LV:**

1 在卷组sky\_vg的基础上,划分一个指定名为:sky\_lv的逻辑卷

2 lvcreate -l 100%VG -n sky\_lv sky\_vg

3 查看逻辑卷:lvdisplay或者 lvs

### 开机自启动

1 编辑/etc/fstab文件追加挂载内容

2 格式:LVM名 挂载点 文件系统类型 defaults 0 0

例如: /dev/sky\_vg/sky\_lv /home xfs defaults 0 0

### LVM扩容

LVM扩容就是将一块全新的磁盘经过格式化后,扩容到现有的卷组中,然后再从卷组中划分磁盘空间到逻辑卷LVM,新磁盘为sdc 步骤如下

1 磁盘格式化

fdisk /dev/sdc

2 创建物理卷PV(sdc1)

pvcreate /dev/sdc

3 将现有的物理卷加入到卷组VG中

vgextend sky\_vg /dev/sdc1

4 划分卷组大小,扩容到LVM中

lvextend -L +200G /dev/sky\_vg/sky\_lv

5 刷新lv大小:resize2fs /dev/sky\_vg/sky\_lv

6 查看磁盘大小:df -Th

## 服务器系统安全加固

### 安装JDK

1 Linux 系统自带openJDK,在安装jdk前需要先删除Openjdk;

2 上传jdk1.8压缩包到 /usr/local/ 解压

3 修改jdk解压后的文件名:mv 解压目录 java

4 编译环境变量,vi /etc/profile 追加以下内容:

export JAVA\_HOME=/usr/local/java

export JRE\_HOME=$JAVA\_HOME/jre

export CLASSPATH=.:$JAVA\_HOME/lib/dt.tar:$JRE\_HOME/lib

export PATH=${JAVA\_HOME}/bin:$PATH

### 修改文件句柄数

1 系统错误提示:error: too many open files,新安装的操作系统默认打开文件数1024,服务器负载过大,会上述错误提示,部署系统前需要在etc目录下修改profile文件,追加以下内容:

ulimit -d unlimited

ulimit -m unlimited

ulimit -s unlimited

ulimit -n 165535

2 配置文件生效: source /etc/profile

### 密码加固

1 安装密码检测模块

yum install -y cr1acklib 密码检测模块

yum install -y libpwquality　密码复杂度评分

2 修改密码复杂度: vim /etc/pam.d/system-auth

在 password requisite pam\_cracklib.so 一行换成如下内容

password requisite pam\_cracklib.so retry=5 difok=3 minlen=10 ucredit=-1 lcredit=-3 dcredit=-3 dictpath=/usr/share/cracklib/pw\_dict

字段解释:

retry=5 失败尝试次数5次

difok=3 至少3种不同字符

minlen=10 最小密码长度

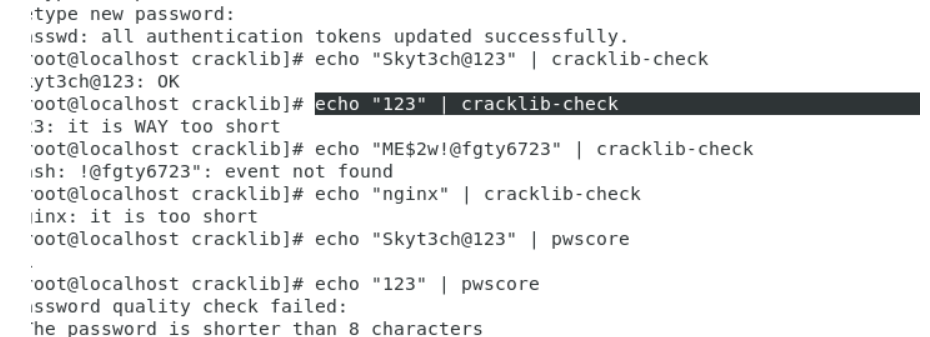
ucredit=-1 最少1个大写字母

lcredit=-3 最少3小写字母数

dcredit=-3 最少3个数字

dictpath=/usr/share/cracklib/pw\_dict 应用数字字典

3 密码测试:



### 服务器超时断开

1 编辑配置文件/etc/profile 新增环境变量

export TMOUT=180 #3分钟

2 变量生效

Source /etc/profile

### 历史命令显示条数

1 编辑配置文件/etc/profile 新增环境变量

export HISTSIZE=50

2 变量生效

Source /etc/profile

### 配置国内yum源

1 修改本地默认yum源文件:

cd /etc/yum.repos.d

mv CentOS-Base.repo CentOS-Base.repo.bak

2 下载和配置阿里的yum源:

Wget -O /etc/yum.repos.d/ CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

3 清理缓存

yum clean all

4重新建立缓存

yum makecache

### 限制root用户直接登录

1 切换到sshd目录

cd /etc/ssh/

2 编辑sshd\_config文件，修改PermitRootLogin 参数为no

3 重启sshd服务

### 关闭selinux防护

1 使用getenforce命令 查看selinux状态，一般默认是enforcing

2 使用setenfoce 0 命令 设置selinux成为permissive模式

3修改/etc/selinux/config文件，将SELINUX=enforcing改为SELINUX=disabled

## 高并发内核优化

#### 进程资源限制(文件打开最大数)

为了防止失控的进程破坏系统的性能,Unix 和 Linux 跟踪进程使用的大部分资源，允许 用户和系统管理员使用对进程的资源限制,

vi /etc/security/limits.conf

\* soft nproc 65535

\* hard nproc 65535

\* soft nofile 65535

\* hard nofile 65535

#### 内核参数优化

编辑/etc/sysctl.conf文件,追加以下内容:

fs.file-max = 6553560 整个系统最大文件句柄数

net.ipv4.tcp\_max\_tw\_buckets = 10000 timewait 的数量，默认是 180000。

net.ipv4.ip\_local\_port\_range = 1024 65000 允许系统打开的端口范围。

net.ipv4.tcp\_tw\_recycle = 1 启用 timewait 快速回收。

net.ipv4.tcp\_tw\_reuse = 1 开启重用。允许将 sockets 重新用于新的 TCP 连接。

net.ipv4.tcp\_syncookies = 1 当出现 SYN 等待队列溢出时，启用 cookies 来处理。

## 安装应用

### Nginx(yum源安装为例)

#### 安装

1 新建nginx本地镜像文件: /etc/yum.repos.d/nginx.repo 添加以下内容:

[nginx-stable]

name=nginx stable repo

baseurl=http://nginx.org/packages/centos/$releasever/$basearch/

gpgcheck=1

enabled=1

gpgkey=https://nginx.org/keys/nginx\_signing.key

[nginx-mainline]

name=nginx mainline repo

baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/

gpgcheck=1

enabled=0

gpgkey=https://nginx.org/keys/nginx\_signing.key2 重新建立本地yun源缓存

3 执行安装命令

yum install nginx -y

4 启动并查看nginx服务

systtemctl start nginx

systemctl status nginx

#### http模块优化

1 切换到nginx配置文件目录

cd /etc/nginx

2 找到nginx.conf文件,在http模块编辑内容

worker\_processes 4; nginx进程数,一般按照cpu数目倍数设置

worker\_cpu\_affinity 00000001 00000010 00000100 00001000 ;

worker\_rlimit\_nofile 102400;

use epoll; 使用 epoll 的 I/O 模型, 提高程序在大量并发连接

worker\_connections 102400; 每个进程允许的最多连接数，

keepalive\_timeout 60; keepalive 超时时间

client\_header\_buffer\_size 4k; 客户端请求头部缓存区大小

open\_file\_cache max=102400 inactive=20s; 为打开文件指定缓存, inactive 是指经过多长时间文件没被请求后删除缓存。

open\_file\_cache\_valid 30s; 这个是指多长时间检查一次缓存的有效信息。

sendfile on; 开启高效文件模式

autoindex on ;开启目录列表访问

tcp\_nopush on;

server\_tokens off;隐藏nginx版本信息

#gzip模块设置

gzip on; #开启gzip压缩输出

gzip\_comp\_level 2; #9个压缩等级

#FastCGI相关参数是为了改善网站的性能：减少资源占用，提高访问速度。

fastcgi\_connect\_timeout 300;

fastcgi\_send\_timeout 300;

fastcgi\_read\_timeout 300;

fastcgi\_buffer\_size 64k;

fastcgi\_buffers 4 64k;

fastcgi\_busy\_buffers\_size 128k;

fastcgi\_temp\_file\_write\_size 128k;

underscores\_in\_headers on; 识别下划线

Includ /etc/nginx/conf.d 配置nginx虚拟主机文件目录

#### 配置虚拟主机

一个配置文件可以配置多个web虚拟主机,每个web虚拟主机内可以配置多个反向代理模块

Server {

#配置web虚拟主机

Server\_name ip地址或者域名;

listen 端口;

#配置反向代理服务

location /test {

proxy\_pass <http://ip:端口> 反向代理

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $host;

proxy\_connect\_timeout 600; nginx与后端服务器连接时间,默认60秒

proxy\_send\_timeout 600; 后端服务器按规定时间内完成传输,默认75秒

proxy\_read\_timeout 600; 后端与nginx请求的响应时间 默认60秒

}

}

#### 配置负载均衡

在http模块添upstream模块

upstream baken {

#upstream的负载均衡，weight是权重，可以根据机器配置定义权重。weigth参数表示权值，权值越高被分配到的几率越大。

server 192.168.80.121:80 weight=3;

server 192.168.80.122:80 weight=2;

server 192.168.80.123:80 backup;其他节点访问压力大或有节点存在异常,则参与访问工作

}

需要使用负载均衡的server的location模块增加 proxy\_pass http://bakend/;

#### 限制请求

为保障系统平稳运行,对个别IP或域名限制访问

限制访问虚拟主机:

server {

……

deny all；

allow IP地址 仅允许信任ip访问

}

限制访问某个反向代理模块

location /baken {

……

deny all；

allow IP地址 仅允许信任ip访问

}

#### nginx限流

ngix限制分为请求限制和并发限制，分别使用nginx的ngx\_http\_limit\_req\_module模块和ngx\_http\_limit\_conn\_module模块，在http模块定义nginx的全局变量limit\_req\_zone和limit\_conn\_zone；在server模块或者location模块实现具体请求限制

**#以下统一在http域中进行配置**

#限制请求

limit\_req\_zone $uri zone=api\_read:20m rate=50r/s;

#nginx并发限制

#按ip配置一个连接 zone

limit\_conn\_zone $binary\_remote\_addr zone=perip\_conn:10m;

#按server配置一个连接 zone

limit\_conn\_zone $server\_name zone=perserver\_conn:100m;

**以下在location模块配置**

#请求限流排队通过 burst默认是0

limit\_req zone=api\_read burst=100;

#连接数限制,每个IP并发请求为50

limit\_conn perip\_conn 50;

#服务所限制的连接数(即限制了该server并发连接数量)

limit\_conn perserver\_conn 200;

#### nginx限速

location ~ /download/ {

...

limit\_rate\_after 150m;

limit\_rate 1m;

...

#说明：limit\_rate\_after定义当一个文件下载到指定大150m 之后开始限速；

limit\_rate 定义下载速度为1m/s。

}

#### Nginx安全平滑启动

Nginx正在运行过程中，需要添加新的反向代理或者是web虚拟主机，在重启的过程中需要保障之前的应用服务进程不受影响。

1 检测新配置是否正确 使用nginx -t 命令 提示出现successfully则配置正确，反之faild则配置错误

2 使用nginx -s reload 命令 平滑重启nginx服务，直接使用systemctl restart nginx 具有一定的风险性，需要断开之前正在运行的nginx进程

### Redis

#### 安装

1 下载和安装fedora的epel仓库

Yum install epel-release -y

2 下载和安装redis服务

Yum install redis -y

3 启动redis服务

Systemctl start redis

4 查看redis服务状态

Systemctl status redis

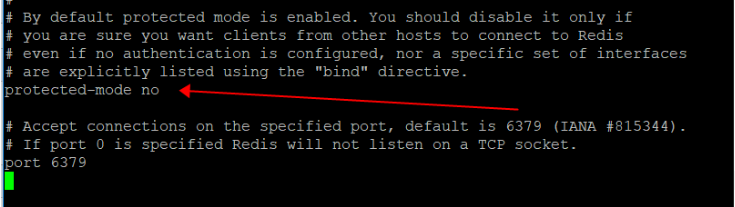
5 设置服务开机自启

Systemctl enable redis

#### 加固优化

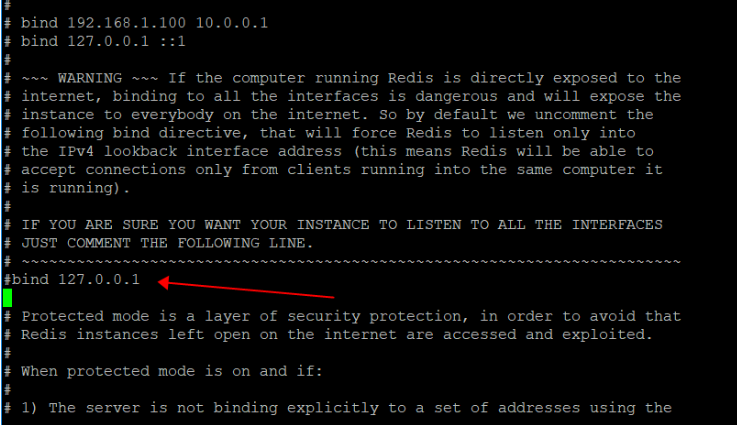
**端口**

redis默认端口6379 在使用期间容易引起网络攻击，需要修改使用端口



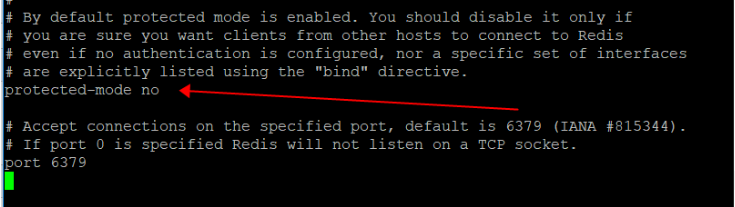
**绑定ip**

Redis默认只允许本地连接访问，这里需要修改为特定的IP访问或是设置为0.0.0.0 开放所有ip访问



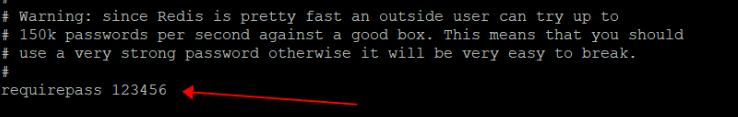
**设置安全模式**

Redis默认使用安全模式，在这里我们将安全模式关闭



**设置密码**

为增加redis缓存安全性，为redis设置密码这里将密码设置为123456



**开放端口**

firewall-cmd –add-port=端口/tcp --zone=public --permanent

firewall-cmd –reload

### MYSQL

#### 安装

1 下载和安装本地镜像库

wget -c <http://dev.mysql.com/get/mysql57-community-release-el7-10.noarch.rpm>

yum -y install mysql57-community-release-el7-10.noarch.rpm

2 建立新的yum 缓存

Yum makecache

3 安装mysql服务端

yum -y install mysql-community-server

4 查看mysql 初始密码

grep "password" /var/log/mysqld.log

5 登录mysql 修改初始密码

ALTER user 'root'@'%' IDENTIFIED BY '9920417'

6 关闭数据库服务

systemctl stop mysql

7 设置mysql使用的磁盘空间

cp -rf /var/lib/mysql /home/ 复制mysql数据库的文件信息到home目录

mv /var/lib/mysql /var/lib/mysql\_bak 备份之前的mysql数据库文件信息

ln -s /home/mysql /var/lib/mysql 建立软连接

chown -R mysql:mysql /var/lib/mysql 修改软连接的属主信息

chown -R mysql:mysql /home/mysql 修改home目录中mysql文件的属主信息

8 重新启动mysql服务

Systemctl restart mysql

#### 优化

1 mysql数据库innodb引擎参数优化

For advice on how to change settings please see

# http://dev.mysql.com/doc/refman/5.7/en/server-configuration-defaults.html

[client]

default-character-set=utf8

[mysqld]

character\_set\_server=utf8

skip-name-resolve

# Remove leading # and set to the amount of RAM for the most important data

# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.

# innodb\_buffer\_pool\_size = 128M

# Remove leading # to turn on a very important data integrity option: logging

# changes to the binary log between backups.

# log\_bin

#

# Remove leading # to set options mainly useful for reporting servers.

# The server defaults are faster for transactions and fast SELECTs.

# Adjust sizes as needed, experiment to find the optimal values.

# join\_buffer\_size = 128M

# sort\_buffer\_size = 2M

# read\_rnd\_buffer\_size = 2M

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

innodb\_log\_files\_in\_group = 4

innodb\_buffer\_pool\_chunk\_size = 1024M

innodb\_buffer\_pool\_instances = 10

innodb\_buffer\_pool\_size = 10240M

innodb\_log\_file\_size = 128M

innodb\_log\_buffer\_size = 8M

innodb\_flush\_log\_at\_trx\_commit = 1

innodb\_lock\_wait\_timeout = 50

innodb\_file\_per\_table = 1

innodb\_open\_files = 2000

max\_connections = 512

sql\_mode=STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION

lower\_case\_table\_names=1

event\_scheduler = 1

innodb\_temp\_data\_file\_path = ibtmp1:12M:autoextend:max:50G

### Rabbitmq

#### 安装

1 按钻erlang依赖

yum -y install make gcc gcc-c++ kernel-devel m4 ncurses-devel openssl-devel

2 下载erlang rmp包

wget <https://packages.erlang-solutions.com/erlang/rpm/centos/7/x86_64/esl-erlang_22.1.8-1~centos~7_amd64.rpm>

3安装rpm 包

yum install esl-erlang\_22.1.8-1~centos~7\_amd64.rpm

4 下载和安装rabbitmq服务

rpm –import https://www.rabbitmq.com/rabbitmq-release-signing-key.asc  
yum install rabbitmq-server-3.8.2-1.el7.noarch.rpm

5 开启服务

systemctl start rabbitmq-serve 开启rabbitmq服务

rabbitmqctl status 查看rabbitmq状态

chkconfig rabbitmq-server on 服务开机自启

6开放外部访问端口：5672和15672，其中15672为管理端口

firewall-cmd –add-port=5672/tcp --zone=public --permanent

firewall-cmd –add-port=15672/tcp --zone=public --permanent

firewall-cmd –reload

7 添加外网用户和外网访问

rabbitmqctl add\_user admin abc124  
rabbitmqctl set\_user\_tags admin administrator

rabbitmqctl set\_permissions -p “/” admin ‘.\*’ ‘.\*’ ‘.\*’  
rabbitmqctl list\_user\_permissions admin

### Ealsticsearch

#### 准备（已修改，则此步省略）

1 修改内核参数

vi/etc/sysctl.conf

vm.max\_map\_count=655360

sysctl -p

2 修改用户打开句柄数

vi/etc/security/limits.conf 追加内容

\* soft nofile 65536

\* hard nofile 131072

\* soft nproc 65536

\* hard nproc 131072

3 修改ES用户软连接

vi/etc/security/limits.d/20-nproc.conf

ES soft nproc 65536

#### 安装

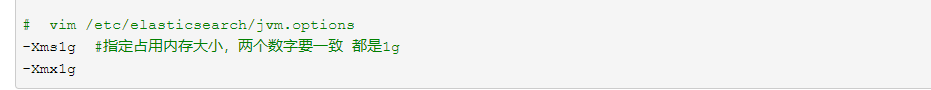
1 下载Elasticsearch 压缩包到指定文件下解压（/home/ealsticsearch）

2 修改elasticsearch.yml

Vi /home/elasticsearch/config/elasticsearch.yml 修改以下内容

###### ======================== Elasticsearch Configuration =========================  
  
###### ---------------------------------- Cluster 客户端-----------------------------------  
  
cluster.name: sky  
  
###### ------------------------------------ Node 节点 ------------------------------------  
#  
node.name: node-1  
#  
  
###### ----------------------------------- Paths 登记数据和日志路径 ------------------------------------  
  
path.data: /var/lib/elasticsearch  
  
path.logs: /var/log/elasticsearch  
----------------------------------- Memory -----------------------------------  
  
无  
  
###### ---------------------------------- Network 开放远程连接端口和地址 -----------------------------------  
  
network.host: 127.0.0.1  
  
http.port: 9200  
  
###### --------------------------------- Discovery ----------------------------------  
  
  
discovery.zen.ping.unicast.hosts: ["127.0.0.1"]  
  
###### ---------------------------------- Gateway -----------------------------------  
无  
###### ---------------------------------- Various   增加配置参数-----------------------------------

3 修改/home/Elasticsearch/config/jvm.optionswen文件，设置启动运行内存 物理内存80%



#### 启动

1 创建一个ES用户，并设置密码

useradd ES

passwd ES

2 修改Elasticsearch文件属主

chown -R ES:ES /home/Elasticsearch

3 切换用户为ES 启动 Elasticsearch

su ES

4 切换到bin目录

cd /home/Elasticsearch/bin

./elasticsearch -d 后台运行或nohup ./elasticsearch &

5 验证是否安装成功

curl <http://IP:9200/_search?pretty>



### Tomcat

#### 安装

1 上传最新的tomcat压缩包到指定目录：/home/tomcat

2 修改配置文件server.xml的应用端口以及优化性能

<Connector port="8088"

protocol="org.apache.coyote.http11.Http11NioProtocol"

connectionTimeout="20000"

redirectPort="8443"

maxThreads="5000"

minSpareThreads="20"

acceptCount="1000"

disableUploadTimeout="true"

enableLookups="false"

URIEncoding="UTF-8" />

3 上传war包到至指定的目录

4 启动服务

Cd /home/tomcat/bin

./start.sh 启动服务

./shutdown 关闭tomcat服务

5 查看端口

lsof -i:8088

### Rsync服务

rsync 是一个用于增量文件传输，用于不同服务器间传输文件、同步文件的利器，第一次同步时 rsync 会复制全部内容，但在下一次只传输修改过的文件，rsync只拷贝发生过变化的文件，提高传输效率非常高。，配置 rsync 配置文件，默认/etc/不存在 rsyncd.conf 配置文件，需要手动创建。

#### 安装服务端：

1 yum -y install rsync

2 服务端配置文件:/etc/rsync.conf

uid = root

gid = root

port = 873

use chroot = yes

#read only = no

# # list = no

max connections = 4

# # pid file = /var/run/rsyncd.pid

transfer logging = yes

timeout = 900

motd file = /etc/rsyncd/rsyncd.motd

log file = /var/log/rsyncd.log

lock file = /var/run/rsyncd.lock

ignore nonreadable = yes

dont compress=\*.gz \*.tgz \*.zip \*.z \*.Z \*.rpm \*.deb \*.bz2

[web]

path=/home/data/web #设置服务端文件目录

comment=web #设置path的别名

read only = yes

auth users = root

secrets file = /etc/rsyncd.passwd

hosts allow = 129.18.30.12 #设置IP客户端连接

3 设置服务端密码：

echo 'root:1233423452asdfg'>/etc/rsyncd.passwd

chmod 600 /etc/rsyncd.passwd

启动服务

systemctl enable rsyncd

systemctl restart rsyncd

开放端口

firewall-cmd --zone=public --add-port=873/tcp --permanent

firewall-cmd –reload

#### 安装客户端：

yum install rsync -y

填入服务端密码：

echo '1233423452asdfg'>/etc/rsyncd.passwd

chmod 600 /etc/rsyncd.passwd

#### 使用方式

客户端拉取服务端文件

rsync -avz --password-file=/etc/rsyncd.passwd root@47.101.99.11::web /backup/webbackup

### Squid缓存代理服务

Squid cache（简称为 Squid）是一个流行的自由软件Squid 作为网页服务器的前置 cache 服务器，可以代理用户向 web 服务器请求数据并进行缓存，也可以用在局域网中，使局域网用户通过代理上网。以下配置squid代理访问互联网

#### 安装

##### 服务端设置

1 yum install squid -y

2 修改 /etc/squid/下配置文件:squid.conf

3 找到文件中的 http\_access deny all 将其修改为 http\_access allow all

4 找到 http\_port 3128 修改为 http\_port x.x.x.x:3128 这里的IP及端口是 squid的代理IP及端口。(本机可不用修改,该IP可以连接外网)

5 启动代理:

执行squid -k parse命令

执行squid -z命令

启动squid服务:systemctl start squid

##### 请求端设置

将服务端IP地址写入请求代理的服务器的系统变量内

export http\_proxy=http://10.251.112.63:3128

export https\_proxy=http://10.251.112.63:3128

以上请求端代理是仅限本次有效,若要永久生效,将上述变量写入/etc/profile内

执行 source /etc/profile

测试 wget https://www.baidu.com 如若失败 考虑查看防火墙是否设置

# 开源工具

## InfluxDB监控系统

相关网址:

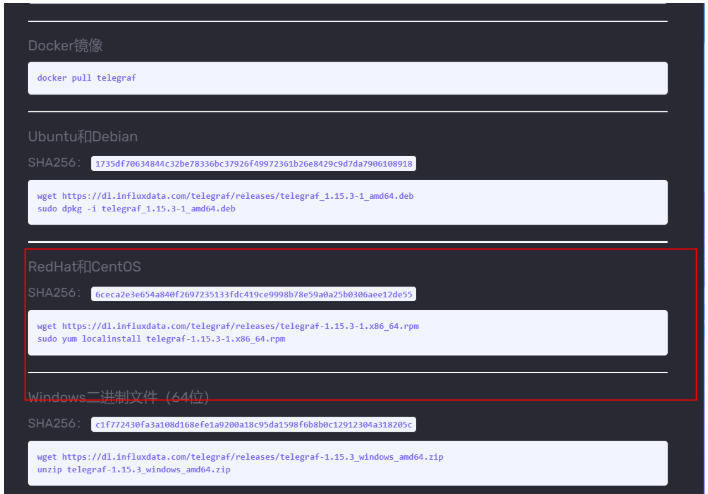
官方网址下载网址:https://portal.influxdata.com/downloads/

官方文档:https://docs.influxdata.com/influxdb/v2.0/

### 下载:

influxdb2.0(CentOS系统)





### 安装

#### 安装influxdb:

1. 创建目录:mkdir /home/app

2. 上传 influxdb压缩包到app目录下

3解压文:tar zxvf [influxdb\_2.0.0-beta.16\_linux\_amd64.tar.gz](https://dl.influxdata.com/influxdb/releases/influxdb\_2.0.0-beta.16\_linux\_amd64.tar.gz)

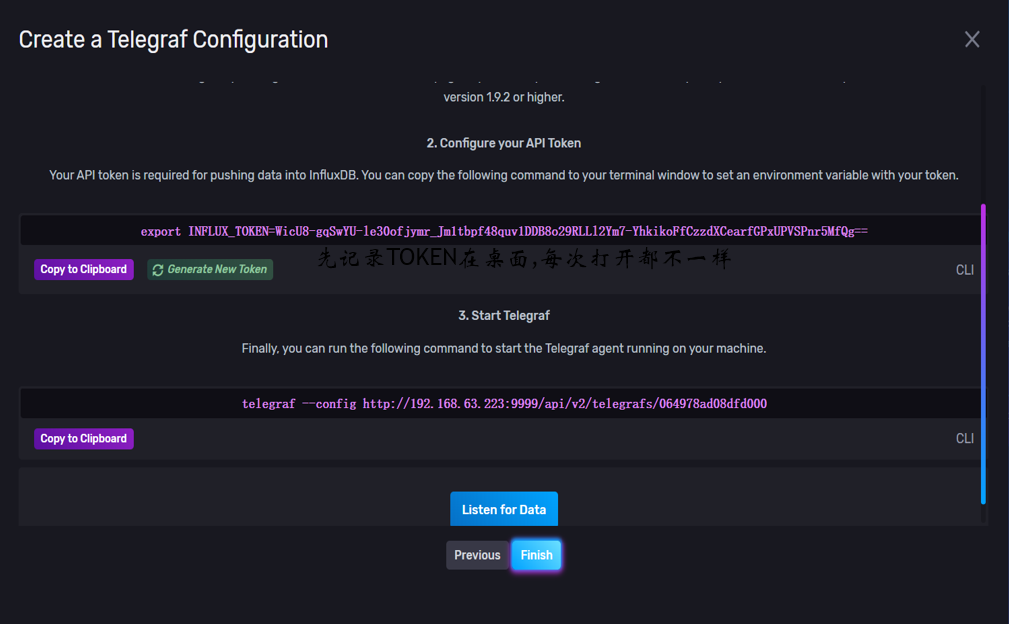
4. 修改文件名称: mv influxdb\_2.0.0 influxdb

5. 运行 influxdb服务: nohup ./influxd &

6. 查看运行日志 tail -f nohup.out

7. 浏览器访问:http://ip:8086/ 打开可视化界面

8. 创建buket,获取token 认证码



#### 安装Telegraf采集节点

1.下载telegraf的rpm文件到app目录下

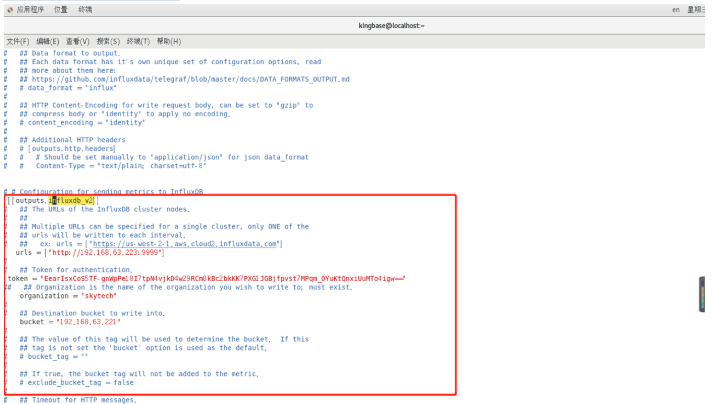
2.rpm –ivh telegraf-1.15.2-1.x86\_64.rpm

3.启动采集服务:systenctl start telegraf.service

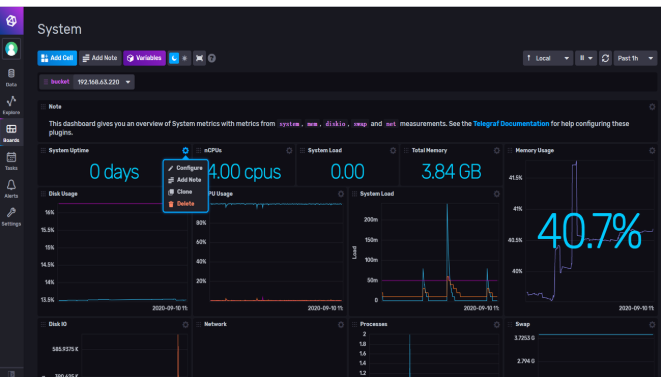
4.查看telegraf 状态:systemctl status telegraf.service

5.编辑配置文件:vi /etc/telegraf/telegraf.conf

6.填写influxdb信息



7 返回仪表盘,查看服务信息

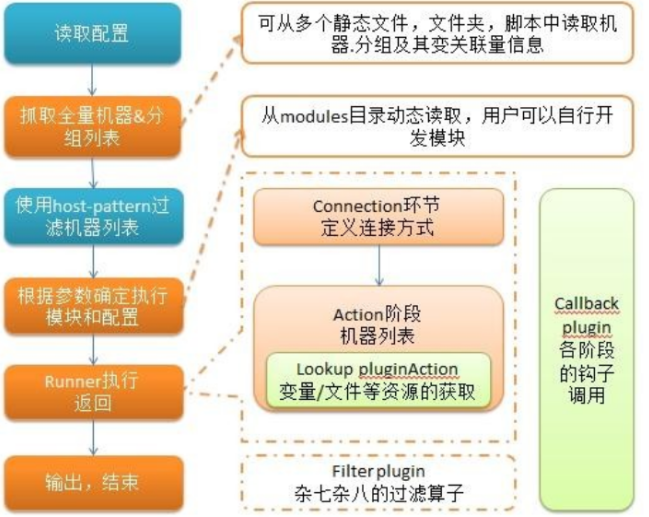


## EFK日志系统

## Ansible 批量管理

ansible基于Pythob开发,集合(puppet、chef、func、fabric)众多运维工具有点，具有批量管理。基于ssh实现远程主机通讯，技术参考文档：<https://www.cnblogs.com/keerya/p/7987886.html>

### ansible执行流程



### ansible命令执行过程

首先读取`ansible.cfg`中的配置， 根据规则获取`Inventory`中的管理主机列表， 并行的在这些主机中执行配置的任务， 最后等待执行返回的结果。

1. 加载自己的配置文件，默认`/etc/ansible/ansible.cfg`；

2. 查找对应的主机配置文件，找到要执行的主机或者组；

3. 加载自己对应的模块文件，如 command；

4. 通过ansible将模块或命令生成对应的临时py文件(python脚本)， 并将该文件传输至远程服务器；

5. 对应执行用户的家目录的`.ansible/tmp/XXX/XXX.PY`文件；

6. 给文件 +x 执行权限；

7. 执行并返回结果；

8. 删除临时py文件，`sleep 0`退出；

### ansible安装

yum install -y ansible

yum install epel-release -y

### ansible 配置

ansible的配置文件在/etc/ansible/ansible.cfg,相关参数如下：

inventory = /etc/ansible/hosts #这个参数表示资源清单inventory文件的位置

library = /usr/share/ansible #指向存放Ansible模块的目录，支持多个目录方式，只要用冒号（：）隔开就可以

forks = 5 #并发连接数，默认为5

sudo\_user = root #设置默认执行命令的用户

remote\_port = 22 #ssh端口，默认为22端口，建议修改，能够更加安全

host\_key\_checking = False #设置是否检查SSH主机的密钥，值为True/False。

timeout = 60 #设置SSH连接的超时时间，单位为秒

log\_path = /var/log/ansible.log #指定一个存储ansible日志的文件（默认不记录日志）

### ansible定义主机清单

在/etc/ansible/host文件中直接指定主机地址或者域名,定义一个机组

[机组名]

服务器IP1

服务器IP2

服务器IP3

服务器IP4

### ansible 命令

ansible [ip地址|机组] [-f forks] [-m 模块] [-a 模块执行参数]

**参数解读：**

ansible -h 帮助 查看常用选项

ansible -a 模块执行参数，默认执行commad 模块

ansible -i 指定清单路径 默认是 /etc/ansible/hosts

ansible -m 执行模块，默认使用command模块，执行单条命令可不使用-m

ansible --list-hosts 查看有哪些主机组

ansible -T 指定ssh 超时连接 默认10s

ansible -u user 远程用户，默认root

ansible --ask-pass 输入登录密码

### 案例

#### Elasticsearch集群服务器运维管理

在/etc/ansible/hosts文件配置服务器组,将明文密码写在配置文件中

1 生成私钥：ssh-keygen

2 主机配置私钥：ssh-copy-id 用户名@服务器IP地址

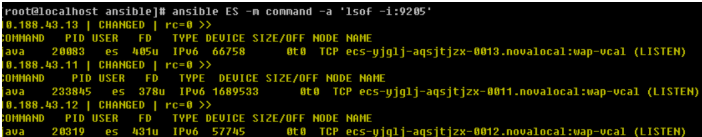
[ES] #以下明文配置安全性较低

10.188.43.11 ansible\_ssh\_user='用户名' ansible\_ssh\_pass='password' [ansible\_ssh\_port='1212']

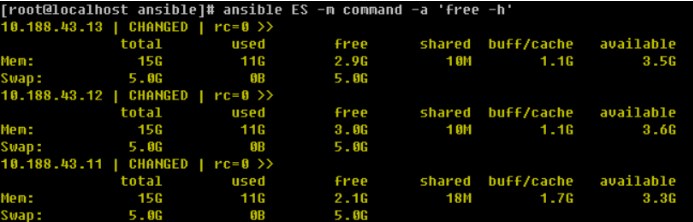
10.188.43.12 ansible\_ssh\_user='用户名' ansible\_ssh\_pass='password' [ansible\_ssh\_port='1212']

10.188.43.13 ansible\_ssh\_user='用户名' ansible\_ssh\_pass='password' [ansible\_ssh\_port='1212']

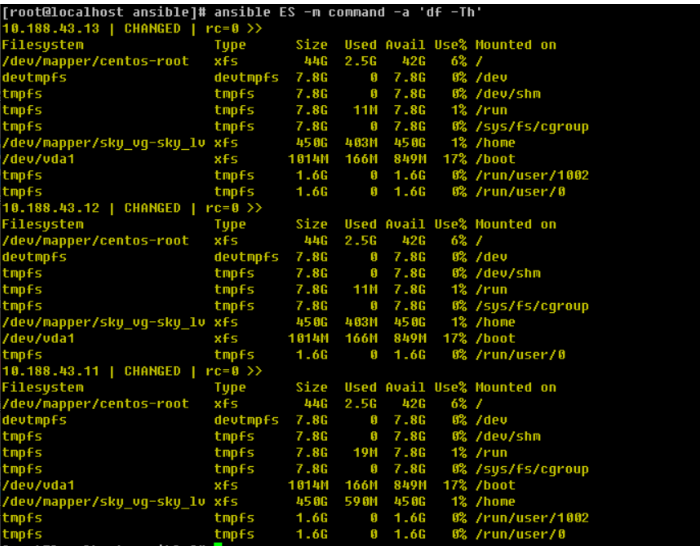
##### 情景1: 查看ES集群进程



##### 情景2:查看ES集群内存



##### 情景3:查看磁盘空间使用



### ansibla-valut加密

使用格式: ansible-vault [create|decrypt|edit|view|encrypt|rekey|encrypt\_string]

Create 创建一个加密文件

decrypt 解密文件

edit 编译加密文件

view 查看加密文件

encrypt 现有文件加密

encrypt\_string Encrypt a string

rekey 修改加密文件的密钥

在/etc/ansible/hosts文件中填写明文密码,但存在一定的漏洞,通过ansibla-valut加密,在需要使用hosts文件的时候,为hosts文件解密

### Ansible 常用模块

Command

Shell: 远程在主机上调用shell解释器，支持管道符

Copy: 将本地文件复制到远程主机，同时支持给定内容生成文件和修改权限等

File:

Script:

## Guacamole网络桌面

# 日常巡检

对应用系统和承载应用系统运行的虚拟服务器CPU、内存、存储的使用情况，提供每周工作1次巡检服务，并做好巡检记录

## 服务器基础项巡检

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 检查项 | 检查操作 | 巡检结果 |
| 1 | 系统账号检查 | 用户密码登录 | 正常 |
| 2 | 服务器主机名检查 | Hostname | 正常 |
| 3 | 操作系统内核版本检查 | uname －a | 正常 |
| 4 | 操作系统版本检查 | cat /proc/version | 正常 |
| 5 | 当前操作系统发行信息 | cat /etc/issue | 正常 |
| 6 | 系统时间检查 | date | 正常 |
| 7 | CPU运行模式检查 | getconfig LONG\_BIT | 正常 |
| 8 | 服务器磁盘挂载检查 | df-h | 正常 |
| 9 | 磁盘开机自挂载 | 查看/etc/fstab文件 | 正常 |

## 登录信息巡检

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 检查项 | 检查操作 | 巡检结果 |
| 1 | 检查当前登录用户 | who | 正常 |
| 2 | 系统运行时间 | uptime | 正常 |
| 3 | 系统账户安全检查 | more /etc/passwd  more /etc/shadow | 正常 |
| 4 | 系统登录情况检查 | lastlog | 正常 |
| 5 | 登录失败记录检查 | lastb | 无异常情况 |
| 6 | 查看登录日志 | /var/log/secure | 无可疑IP登录记录 |
| 7 | 查看日志报错信息 | dmesg -T | tail | 日志中无报错信息或报错信息不影响系统运行 |
| 8 | 系统开放端口检查 | netstat -an |grep LISTEN | 无异常的端口打开 |

## 服务器性能巡检

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 检查项 | 检查操作 | 巡检结果 |
| 1 | CPU进程占用率 | top | 正常 |
| 2 | 内存占用率 | free -g | 正常 |
| 3 | 内存swap使用率 | free -m | swap使用率小于10% |
| 4 | 文件系统占用率 | df -Th | 正常 |
| 5 | 进出占用资源检查 | Top shift+m  top -p 进程名 | 无异常进程 |
| 6 | 磁盘I/O性能检查 | iostat–x 2 5  每隔2秒刷新一次，共执行5次 | 正常 |
| 7 | CPU负载 | uptime | 正常 |

## 应用服务巡检

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 检查项 | 检查操作 | 巡检结果 |
| 1 | 应用日志 | 检查日志文件 | 正常 |
| 2 | 应用进程 | 检查端口进程 | 正常 |
| 3 | 应用日志大小 | du | 正常 |
| 4 | 信任端口 | 查看防火墙端口 | 正常 |
| 5 | 应用代理 | 检查Nginx服务 | 正常 |
| 6 | Tomcat应用服务 | 日志查看 | 内存溢出` |
| 7 | 其他 |  |  |

# 数据安全

## 数据备份策略

1 将cron\_backup.sh文件和dbbackup.sh文件上传到指定目录,这里指定目录为/home/emergency/taobao\_prod/script

2 脚本内容按月保留每天的备份文件,文件名一般以年月命名

2 修改文件属主为root用户

3 新建定时任务,每日全量备份,新增定时任务:crontab -e

12 0 \* \* /etc/profile;/bin/sh /home/emergency/taobao\_prod/script/dbbackup.sh> /home/emergency/taobao\_prod/script/backlog/cevadbbackup.log 2>&1

4 重新启动crond 服务:systemctl restart crond.service

5 第二个工作日检查 是否正常备份

## 数据保留(异地)

1 查看磁盘空间: df -Th ,查看磁盘空间大小

2 磁盘剩余空间小于20% 则 删除最早期的备份文件

3 删除每个月的备份时 必须保留该月第一天与最后一天的备份文件

4 检查备份sql文件的大小,合理释放磁盘空间,一般磁盘可用空间要大于50%

## 数据恢复

1 选择最新的备份文件,上传到数据库本地服务器

2

## 数据库服务器迁移

1 服务器磁盘挂载、数据库安装优化

2 和研发确定服务器迁移时间（尽量避免在使用期间）、IP地址、账户信息

3 原数据库锁定,导出最新备份数据库文件backupfile.sql

4 把备份sql文件上传到新的服务器 此处目录为/home

5 数据导入：mysql -u username -p password databasename < /homebackupfile.sql

6 通知研发测试，确定没有问题

7 原数据库服务器关闭，相关数据库配置与文件暂时保留一段时间

8 修改备份任务脚本，重新启动

# 安全运维

## 事故分类

事件分为网络攻击事件、信息破坏事件、信息内容安全事件、网络故障事件、软件系统故障事件、灾难性事件、其他事件等八类事件。

## 事故等级划分

按照造成系统的中断运行时间，将系统事件级别划分为一般III级、较大II级、重大I级。

|  |  |
| --- | --- |
| 一般III级 | 软硬件正常、系统访问正常，用户使用操作规范问题 |
| 较大II级 | 网络故障或软件故障导致提示：404或500报错提示 |
| 重大I级 | 虚机操作系统故障、数据磁盘挂载故障、服务器硬件故障等基础环境故障以及云平台遭恶意攻击等信息安全事件，都会对应用系统带来灾难性、不可恢复的高风险事件 |

## 应急组织设置

项目运维设置应急小组，负责系统应急处理工作，决定系统应急处理工作的重大事项，组织实施、业务协调和发布系统应急指令，发布系统应急故障级别、决策处理方案，明确应急小组组长和组成人员。

### 运维公司应急预案小组

运维公司应急预案小组由组长、执行组长、实施组成员、开发组成员4人组成，组长负责应急保障工作总体协调；执行组长为应急保障工作的第一责任人，发生系统故障事件，应第一时间反馈执行组长配合解决；运维成员为应急保障工作的第二责任人，当无法与执行组长取得联系时，应反馈第二责任人配合解决；开发组成员负责应急保障工作的代码技术支持，协助解决系统的编码缺陷；相关人员信息如下表所示，记录在公司项目应急预案内。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **姓名** | **职责** | **联系方式** | **备注** |
| 1 | xxx | 组长 | xxx | 总体协调 |
| 2 | xxx | 执行组长 | xxx | 第一责任人 |
| 3 | xxx | 运维成员 | xxx | 第二责任人 |
| 4 | xxx | 开发组成员 | xxx | 技术负责人 |

### 运维管理部门应急预案小组

运维管理部门应急预案小组由组长、执行组长、联络员3人组成，组长负责应急保障工作总体协调；执行组长为应急保障工作的第一责任人，发生系统故障事件，应第一时间将事件情况反馈组长和执行组长；联络员为应急保障工作的基础设施责任人，负责协调解决基础软硬件问题，相关人员信息如下表所示，记录在公司项目应急预案内。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **姓名** | **职责** | **联系方式** | **备注** |
| 1 | xxx | 组长 | xxx | 总体协调 |
| 2 | xxx | 执行组长 | xxx | 第一负责人 |
| 3 | xxx | 联络员 | xxx | 基础设施负责人 |

## 应急响应

### 重大I级事件

当系统发生重大I级事件时，应立即向运维管理部门应急预案小组组长和执行组长反馈，工作日运维公司驻场人员应即刻响应并处理，非工作日运维公司应在接到报修后2小时内到达现场进行处理。事故处理完成后，2个工作日内编制完成事故处理分析报告提交信息中心确认。

### 较大II级事件

当系统发生较大II级事件时，应立即向运维管理部门应急预案小组执行组长反馈，工作日运维公司驻场人员应即刻响应并处理，非工作日运维公司应在接到报修后4小时到达现场进行处理。

### 一般III级事件

当系统发生一般III级事件时，工作日运维公司驻场人员应即刻响应并处理，非工作日运维公司应在接到报修后的第一个工作日进行处理

## 事件处理流程



确定问题来源，如果是客户反馈给项目组长或项目经理，项目组长或项目经理在将问题通知到运维后，对问题进行分析，确定事件分类，启动相应的应急措施；如果是代码缺陷导致的问题来源，协调相应的开发人员，对代码做出修改，通知运维人员重新部署，同时将部署信息告诉项目组长或项目经理，反馈给客户，保障客户满意度