目录

[1. 运行环境 1](#_Toc514657297)

[2. 安装JDK 1](#_Toc514657298)

[3. 安装nginx 1](#_Toc514657299)

[4. 安装Redis 4](#_Toc514657300)

[4.1 下载Redis 4](#_Toc514657301)

[4.2 上传到liunx服务器 4](#_Toc514657302)

[4.3 解压Redis 5](#_Toc514657303)

[4.4 编译安装Redis 5](#_Toc514657304)

[4.5执行安装命令 5](#_Toc514657305)

[4.6 配置Redis 6](#_Toc514657306)

[4.7 启动Redis 6](#_Toc514657307)

[4.8 添加开机启动项 6](#_Toc514657308)

[5. 安装MySQL数据库 7](#_Toc514657309)

[5.1 到mysql官网下载最新的mysql包 7](#_Toc514657310)

[5.2上传到linux服务器目录 /usr/local 8](#_Toc514657311)

[5.3 进入/usr/local目录解压安装包并重命名 8](#_Toc514657312)

[5.4创建用户组mysql，创建用户mysql并将其添加到用户组mysql中，并赋予读写权限 8](#_Toc514657313)

[5.5创建配置文件 保存并退出 8](#_Toc514657314)

[5.6初始化数据库 9](#_Toc514657315)

[5.7查看初始密码 9](#_Toc514657316)

[5.8启动服务，进入mysql，修改初始密码，运行远程连接 9](#_Toc514657317)

[5.9登录失败修改密码： 10](#_Toc514657318)

[6. 配置Mysql双主 10](#_Toc514657319)

[6.1 修改/etc/my.cnf配置文件 10](#_Toc514657320)

[6.2 配置数据同步账户 11](#_Toc514657321)

[7. 安装rabbitMq 12](#_Toc514657322)

[8. 安装keepalived 13](#_Toc514657323)

[9. 配置keepalived 14](#_Toc514657324)

[9.1配置 keepalived.conf 14](#_Toc514657325)

[9.2 验证keepalived 15](#_Toc514657326)

[9.3 配置Nginx的双机 15](#_Toc514657327)

[9.4配置Redis的双机 16](#_Toc514657328)

[9.5配置mysql的双机 16](#_Toc514657329)

[10. 加入自启动服务 17](#_Toc514657330)

[11. 安装NodeJs 18](#_Toc514657331)

[12. 安装PM2 18](#_Toc514657332)

[13. 部署信息 19](#_Toc514657333)

[14. 启动/停止后台API 19](#_Toc514657334)

[15. 启动/停止前端系统 19](#_Toc514657335)

# 运行环境

操作系统：centos7

JDK版本：jdk1.8

Nodejs版本：8版本及以上

PM2版本：2.5版本及以上

Redis版本：3.2.4版本

数据库：MySQL5.7及以上版本

# 安装JDK

上传新的jdk-8u25-linux-x64.rpm软件到/usr/local/执行以下操作：

[root@zck local]# rpm -ivh jdk-8u25-linux-x64.rpm

说明：JDK默认安装在/usr/java中。

【验证安装】

执行以下操作，查看信息是否正常：

[root@localhost ~]# java -version

java version "1.8.0\_25"

Java(TM) SE Runtime Environment (build 1.8.0\_25-b17)

Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)

【配置环境变量】

修改系统环境变量文件

vi /etc/profile

向文件里面追加以下内容：

JAVA\_HOME=/usr/java/jdk1.8.0\_25

JRE\_HOME=/usr/java/jdk1.8.0\_25/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

【使修改生效】

[root@localhost ~]# source /etc/profile   //使修改立即生效

# 安装nginx

3.1下载安装包[nginx-1.2.4.tar.gz](http://nginx.org/download/nginx-1.2.4.tar.gz)（按住ctrl并单机此处下载）

3.2将安装包分别上传到192.168.103.97、192.168.103.98两台服务器

上传目录：*/home/app/soft/*

3.3 解压安装包

*tar zxvf nginx-1.2.4.tar.gz*

3.4 进入解压后的文件夹

*cd nginx-1.2.4*

3.5 安装

*make && make install*

3.5 配置nginx

*vi /usr/local/nginx/conf/nginx.conf*

配置内容参考

|  |
| --- |
| #user nobody;  worker\_processes 1;  error\_log logs/error.log;  #error\_log logs/error.log notice;  #error\_log logs/error.log info;  #pid logs/nginx.pid;  events {  worker\_connections 1024;  }  http {  include mime.types;  default\_type application/octet-stream;  #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  # '$status $body\_bytes\_sent "$http\_referer" '  # '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  #access\_log logs/access.log main;  sendfile on;  #tcp\_nopush on;  #keepalive\_timeout 0;  keepalive\_timeout 65;  #gzip on;  #设置反向代理  upstream proxy\_local {  server 192.168.103.100:3300 weight=1;  #负载的第一个台机器  server 192.168.103.101:3310 weight=1;  #负载的第二个台机器  server 192.168.103.102:3300 weight=1;  #负载的第三个台机器  #ip\_hash; #当负载两台以上用ip来hash解决session的问题，一台就别hash了。  }  upstream proxy\_trunk {  server 192.168.103.100:8000 weight=1;  #负载的第一个台机器  server 192.168.103.101:8000 weight=1;  #负载的第二个台机器  server 192.168.103.102:8000 weight=1;  #负载的第三个台机器  #ip\_hash; #当负载两台以上用ip来hash解决session的问题，一台就别hash了。  }  server {  listen 8000;  server\_name localhost;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  proxy\_pass http://proxy\_trunk;  #使用反向代理  #Proxy Settings  proxy\_redirect off;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;  proxy\_connect\_timeout 90;  proxy\_send\_timeout 90;  proxy\_read\_timeout 90;  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }  }  # HTTPS server  #  server {  listen 443 ssl;  server\_name localhost;  ssl\_certificate /home/app/server.pem;  ssl\_certificate\_key /home/app/server.key;  ssl\_session\_cache shared:SSL:1m;  ssl\_session\_timeout 5m;  ssl\_ciphers HIGH:!aNULL:!MD5;  ssl\_prefer\_server\_ciphers on;    location / {  proxy\_pass http://proxy\_local;  #Proxy Settings  proxy\_redirect off;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;  proxy\_connect\_timeout 90;  proxy\_send\_timeout 90;  proxy\_read\_timeout 90;  }  }  } |

# 安装Redis

[**Redis**](http://lib.csdn.net/base/redis)是一个高性能的，开源key-value型[**数据库**](http://lib.csdn.net/base/mysql)。是构建高性能，可扩展的Web应用的完美解决方案，可以内存存储亦可持久化存储。因为要使用跨进程，跨服务级别的数据缓存，在对比多个方案后，决定使用Redis。顺便整理下Redis的安装过程，以便查阅。

### 下载Redis

目前，最新的Redist版本为4.0，在浏览器中输入一下地址下载

<http://download.redis.io/releases/redis-4.0.2.tar.gz>

### 4.2 上传到liunx服务器

？？？？？

### 4.3 解压Redis

下载完成后，使用tar命令解压下载文件：

# tar -xzvf redis-4.0.2.tar.gz

### 4.4 编译安装Redis

切换至程序目录，并执行make命令编译：

# cd redis-3.0.4

# make

备注：如果不能编译请重新下载安装包

### 4.5执行安装命令

# make install

make install安装完成后，会在/usr/local/bin目录下生成下面几个可执行文件，它们的作用分别是：

redis-server：Redis服务器端启动程序

redis-cli：Redis客户端操作工具。也可以用telnet根据其纯文本协议来操作

redis-benchmark：Redis性能测试工具

redis-check-aof：数据修复工具

redis-check-dump：检查导出工具

备注

有的机器会出现类似以下错误：

make[1]: Entering directory `/root/redis/src'

You need tcl 8.5 or newer in order to run the Redis test

……

这是因为没有安装tcl导致，yum安装即可：

yum install tcl

### 4.6 配置Redis

复制配置文件到/etc/目录：

# cp redis.conf /etc/

为了让Redis后台运行，一般还需要修改redis.conf文件：

vi /etc/redis.conf

修改daemonize配置项为yes，使Redis进程在后台运行：

daemonize yes

### 4.7 启动Redis

配置完成后，启动Redis：

# cd /usr/local/bin

# ./redis-server /etc/redis.conf

检查启动情况：

# ps -ef | grep redis

看到类似下面的一行，表示启动成功：

root 18443 1 0 13:05 ? 00:00:00 ./redis-server \*:6379

### 4.8 添加开机启动项

让Redis开机运行可以将其添加到rc.local文件，也可将添加为系统服务service。本文使用rc.local的方式，添加service请参考：Redis 配置为 Service 系统服务 。

为了能让Redis在服务器重启后自动启动，需要将启动命令写入开机启动项：

echo "/usr/local/bin/redis-server /etc/redis.conf" >>/etc/rc.local

**6.9 Redis配置参数**  
在 前面的操作中，我们用到了使Redis进程在后台运行的参数，下面介绍其它一些常用的Redis启动参数：

daemonize：是否以后台daemon方式运行

pidfile：pid文件位置

port：监听的端口号

timeout：请求超时时间

loglevel：log信息级别

logfile：log文件位置

databases：开启数据库的数量

save \* \*：保存快照的频率，第一个\*表示多长时间，第三个\*表示执行多少次写操作。在一定时间内执行一定数量的写操作时，自动保存快照。可设置多个条件。

rdbcompression：是否使用压缩

dbfilename：数据快照文件名（只是文件名）

dir：数据快照的保存目录（仅目录）

appendonly：是否开启appendonlylog，开启的话每次写操作会记一条log，这会提高数据抗风险能力，但影响效率。

appendfsync：appendonlylog如何同步到磁盘。三个选项，分别是每次写都强制调用fsync、每秒启用一次fsync、不调用fsync等待系统自己同步

# 安装MySQL数据库

1. 卸载MariaDB

CentOS的默认数据库已经不再是MySQL了，而是MariaDB。MariaDB数据库管理系统是MySQL的一个分支，主要由开源社区在维护，采用GPL授权许可。开发这个分支的原因之一是：甲骨文公司收购了MySQL后，有将MySQL闭源的潜在风险，因此社区采用分支的方式来避开这个风险。MariaDB的目的是完全兼容MySQL，包括API和命令行，使之能轻松成为MySQL的代替品。

查看当前安装的mariadb包：

[root@iZ23xz8uttcZ ~]# rpm -qa | grep mariadb

将它们统统强制性卸载掉：

[root@iZ23xz8uttcZ ~]# rpm -e --nodeps mariadb-libs-5.5.56-2.el7.x86\_64

### 5.1 到mysql官网下载最新的mysql包

[mysql-5.7.21-linux-glibc2.12-x86\_64https://dev.mysql.com/downloads/file/?id=474755](https://dev.mysql.com/downloads/file/?id=474755)

### 5.2上传到linux服务器目录 /usr/local

### 5.3 进入/usr/local目录解压安装包并重命名

*cd /usr/local/*

*tar -xzvf* [*mysql-5.7.21-linux-glibc2.12-x86\_64*](https://dev.mysql.com/downloads/file/?id=474755)*.tar.gz*

*mv* [*mysql-5.7.21-linux-glibc2.12-x86\_64*](https://dev.mysql.com/downloads/file/?id=474755) *mysql*

### 5.4创建用户组mysql，创建用户mysql并将其添加到用户组mysql中，并赋予读写权限

*groupadd mysql*  --创建mysql用户组组

*useradd -r -g mysql mysql*  --创建mysql用户并添加到mysql用户组中

*chown -R mysql mysql/*  --将mysql目录访问权限赋为myql用户

*chgrp -R mysql mysql/* --改变mysql目录的用户组属于mysql组

注：

chmod 命令

用于改变文件或目录的访问权限。用户用它控制文件或目录的访问权限。

chgrp命令 改变文件或目录所属的组。

-R处理指定目录以及其子目录下的所有文件

### 5.5创建配置文件 保存并退出

*vim /etc/my.cnf*

#复制以下内容

|  |
| --- |
| [client]  port = 3306  socket = /tmp/mysql.sock  [mysqld]  character\_set\_server=utf8  init\_connect='SET NAMES utf8'  basedir=/usr/local/mysql  datadir=/usr/local/mysql/data  socket=/tmp/mysql.sock  log-error=/var/log/mysqld.log  pid-file=/var/run/mysqld/mysqld.pid  #不区分大小写  lower\_case\_table\_names = 1  sql\_mode=STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION  max\_connections=5000  default-time\_zone = '+8:00' |

### 5.6初始化数据库

#先安装一下这个东东，要不然初始化有可能会报错，初始化失败再进行安装

*yum install libaio*

#手动编辑一下日志文件，什么也不用写，直接保存退出

*cd /var/log/*

*vim mysqld.log*

*：wq*

*chmod 777 mysqld.log*

*chown mysql:mysql mysqld.log*

*/usr/local/mysql/bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data*

### 5.7查看初始密码

*cat /var/log/mysqld.log*

将密码记下来

最后一行：root@localhost: 这里就是初始密码

### 5.8启动服务，进入mysql，修改初始密码，运行远程连接

#然后执行如下操作开启mysql服务，以及设置相关权限

*cd /var/run/*

*mkdir mysqld*

*chmod 777 mysqld*

*cd mysqld*

*vim mysqld.pid*

*:wq*

*chmod 777 mysqld.pid*

*chown mysql:mysql mysqld.pid*

启动服务

*/usr/local/mysql/support-files/mysql.server start*

登录mysql

*/usr/local/mysql/bin/mysql -uroot -p* 你在上面看到的初始密码

### 5.9登录失败修改密码：

*vi /etc/my.cnf*

添加一行： skip-grant-tables

保存退出 :wq

重启mysql服务：*/usr/local/mysql/support-files/mysql.server start*

无密码登录：*/usr/local/mysql/bin/mysql*

修改密码：

*mysql> use mysql;*

*mysql> update user set authentication\_string =password('A1b2c3#4'), host = '%' where user = 'root';*

*mysql> flush privileges;*

*mysql> exit;*

注释掉刚才添加的那行配置（skip-grant-tables）后重启mysql服务，使用密码登录验证是否修改成功

# 配置Mysql双主

### 6.1 修改/etc/my.cnf配置文件

主机A添加配置如下：

|  |
| --- |
| skip-grant-tables=1  server-id = 1  log-bin = mysql-bin  sync\_binlog = 1  binlog\_checksum = none  binlog\_format = mixed  auto-increment-increment = 2  auto-increment-offset = 1  slave-skip-errors = all |

主机B添加配置如下：

|  |
| --- |
| skip-grant-tables=1  server-id = 2  log-bin = mysql-bin  sync\_binlog = 1  binlog\_checksum = none  binlog\_format = mixed  auto-increment-increment = 2  auto-increment-offset = 2  slave-skip-errors = all |

### 6.2 配置数据同步账户

mysql> flush privileges;

mysql> grant replication slave on \*.\* to 'repl'@'192.168.103.104' identified by 'A1b2c3#4';

mysql> flush privileges;

备注：'repl' 同步的账户（两台一样） ；192.168.103.104 在MasterA填写的MasterB的IP；'A1b2c3#4' 是同步的密码（两台一样）

mysql> show master status;

|  |
| --- |
| +------------------+----------+--------------+------------------+-------------------+  | File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |  +------------------+----------+--------------+------------------+-------------------+  | mysql-bin.000001 | 2816 | | | |  +------------------+----------+--------------+------------------+-------------------+ |

mysql> change master to master\_host='192.168.103.104',master\_user='repl',master\_password='A1b2c3#4',master\_log\_file='mysql-bin.000001',master\_log\_pos=2816;

**备注：**master\_host，master\_log\_file，master\_log\_pos是填写对方的信息

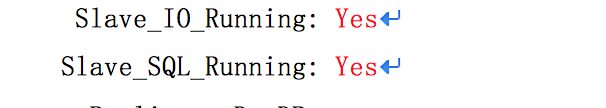
mysql> flush privileges;

#启动复制

mysql> start slave;

#查看是否设置成功，有如下图中的标示就是OK了

mysql> show slave status\G;



# 安装rabbitMq

9.1安装erlang  
以root身份执行下面命令  
yum install erlang  
yum install erlang xmlto  
9.2到官网下载rabbitmq安装包

<http://www.rabbitmq.com/install-rpm.html>

9.3上传安装包到linux服务器  
9.4执行安装命令

rpm -Uvh rabbitmq-server-3.5.6-1.noarch.rpm

9.5启动rabbitmq，并验证启动情况   
rabbitmq-server --detached &ps aux |grep rabbitmq  
  
9.6以服务的方式启动  
service rabbitmq-server start  
  
9.7打开端口5672  
firewall-cmd --zone=public --add-port=5672/tcp --permanent

firewall-cmd --reload  
  
9.8启用维护插件  
rabbitmq-plugins enable rabbitmq\_management   
  
9.9重启rabbitmq  
service rabbitmq-server restart  
  
9.10登录  
http://IP:15672/ 用户名密码 guest  
创建一个帐号 支持远程登录

rabbitmqctl add\_user test 123456

rabbitmqctl set\_user\_tags test administrator

rabbitmqctl set\_permissions -p "/" test ".\*" ".\*" ".\*"

rabbitmqctl list\_users

无法登陆解决办法  
vim /etc/rabbitmq/rabbitmq.config  
写入信息，并保存

[{rabbit, [{loopback\_users, []}]}].

-----------------------------------------------------------分割线-------------------------------------------------

1、服务器启动与关闭  
启动:service rabbitmq-server start  
关闭:service rabbitmq-server stop  
重启:service rabbitmq-server restart  
  
2、用户管理  
新增 rabbitmqctl add\_user admin admin  
删除 rabbitmqctl delete\_user admin  
修改 rabbitmqctl change\_password admin admin123  
  
用户列表 rabbitmqctl  list\_users  
设置角色 rabbitmqctl set\_user\_tags admin administrator monitoring policymaker management  
  
设置用户权限 rabbitmqctl  set\_permissions  -p  VHostPath  admin  ConfP  WriteP  ReadP  
查询所有权限 rabbitmqctl  list\_permissions  [-p  VHostPath]  
指定用户权限 rabbitmqctl  list\_user\_permissions  admin  
清除用户权限 rabbitmqctl  clear\_permissions  [-p VHostPath]  admin

# 安装keepalived

4.1上传目录：/home/app/soft/

4.2 解压安装包

*tar zxvf keepalived-1.4.0.tar.gz*

4.3 切换到root用户

*sudo –s*

4.4 切换到soft目录

*cd /home/app/soft*

4.5 安装四个包（最好手动输入下面的命令，否则可能执行失败）

*rpm –ivh libnl-1.1.4-3.el7.x86\_64.rpm*

*rpm –ivh libnl-devel-1.1.4-3.el7.x86\_64.rpm*

*rpm –ivh libnfnetlink-1.0.1-4.el7.x86\_64.rpm*

*rpm –ivh libnfnetlink-devel-1.0.1-4.el7.x86\_64.rpm*

4.6进去keepalived-1.4.0目录

*cd /home/app/soft/keepalived-1.4.0*

4.7执行

*./configure --prefix=/usr/local/keepalived*

4.8执行

*make && make install*

4.9创建目录

*mkdir /etc/keepalived*

4.10复制文件

*cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf*

*cp /home/app/soft/keepalived-1.4.0/keepalived/etc/init.d/keepalived /etc/rc.d/init.d/keepalived*

*cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/keepalived*

# 配置keepalived

### 9.1配置 keepalived.conf

*vi /etc/keepalived/keepalived.conf*

内容如下，**黄色背景改为实际值**：

|  |
| --- |
| global\_defs {  notification\_email {  #指定keepalived在发生切换时需要发送email到的对象，一行一个  xuyy@redoornetwork.com  }  #指定发件人  notification\_email\_from wanghaikuo@hysec.com  #指定smtp服务器地址  smtp\_server mail.qq.com  #指定smtp连接超时时间  smtp\_connect\_timeout 30  #运行keepalived机器的一个标识  router\_id nginx\_master\_97  }  vrrp\_instance VI\_1 {  state MASTER  ##设置实例绑定的网卡  interface eno16777984  #VPID标记，和备机一样  virtual\_router\_id 51  #优先级，高优先级竞选为master  priority 101  advert\_int 1 #检查间隔，默认1秒  #设置认证  authentication {  auth\_type PASS  auth\_pass A1b2  }  #设置vip  virtual\_ipaddress {  192.168.103.1  }  } |

常用命令

启动： *service keepalived start*

停止： *service keepalived stop*

检查启动成功与否：*ip addr*

### 9.2 验证keepalived

5.2.1先后在主、从服务器上启动keepalived

5.2.2在主服务器上查看是否已经绑定了虚拟IP： ip addr

5.2.3停止主服务器上的keepalived 然后在从服务器上查看是否已经绑定了虚拟IP

5.2.4启动主服务器上的keepalived，看看主服务器能否重新接管虚拟IP

5.2.5在备机上ping虚拟IP，是否能够ping通，如果不通采用下面的方法：

* + 若两台机器配置了防火墙，需要放开组播地址的权限：

*iptables -I INPUT -i eth3 -d 224.0.0.18 -j ACCEPT*

*systemctl restart firewalld.service*

*5.2.6 如果在主、备机可以ping通，别的机器不通，可能是IP不在同一个网段，如果端口不通，可能防火墙没有开放端口*

### 9.3 配置Nginx的双机

*vi /etc/keepalived/keepalived.conf*

加入以下配置（参考keepalived.conf.bak）：

|  |
| --- |
| vrrp\_script chk\_http\_port {  #黄色背景为nginx的检查脚本，需要给root执行权限  script "/home/app/soft/chk\_nginx.sh"  interval 2  weight 2  }  track\_script {  chk\_http\_port  } |

### 9.4配置Redis的双机

*vi /etc/keepalived/keepalived.conf*

加入以下配置（参考keepalived.conf.bak）：

|  |
| --- |
| vrrp\_script chk\_http\_port {  #黄色背景为nginx的检查脚本，需要给root执行权限  script "/home/app/soft/redis-backup/chk\_redis.sh"  interval 2  weight 2  }  track\_script {  chk\_http\_port  } |

5.4.3 修改Redis配置文件，将IP设置为：0.0.0.0

*vi /etc/redis.conf*

将 bind 127.0.0.1 改为 bind 0.0.0.0

### 9.5配置mysql的双机

*vi /etc/keepalived/keepalived.conf*

加入以下配置（参考keepalived.conf.bak）：

|  |
| --- |
| vrrp\_script chk\_http\_port {  #黄色背景为nginx的检查脚本，需要给root执行权限  script "/home/app/soft/chk\_mysql.sh"  interval 2  weight 2  }  track\_script {  chk\_http\_port  } |

# 加入自启动服务

echo "/usr/local/nginx/sbin/nginx" >> /etc/rc.local

echo "service keepalived start" >> /etc/rc.local

# 安装NodeJs

1. $ sudo yum group install 'Development Tools'
2. 下载最新的安装包：

<http://nodejs.org/dist/node-latest.tar.gz>

1. 通过sftp上传安装包到linux服务器
2. 解压到当前文件夹

$ tar xvfvz node-latest.tar.gz

1. 执行完上一布，当前文件夹会出现一个新的文件夹，是最新的node。

$ cd node-v0.10.21(文件名取决于上一步新增的文件夹)

1. $ ./configure
2. $ make
3. $ sudo make install
4. 依次执行上面的指令，其中make过程可能非常久（久到怀疑人生），请耐心等待，指令执行完毕，Node也就安装好了，可以用node -v和npm -v来检查下。
5. 如果没有版本信息，可能是路径没有配对。

$ echo $PATH, 看看输出是什么，里面有没有 /usr/local/bin。

1. $ vim /etc/profile
2. 在文档最后，添加:
3. export PATH="/usr/local/bin:$PATH"
4. 保存，退出，然后运行：
5. $ source /etc/profile
6. 不报错则成功。
7. 淘宝镜像地址 <http://npm.taobao.org> 执行 npm 命令时下载速度更快
8. $ npm install -g cnpm --registry=https://registry.npm.taobao.org

安装依赖就用 cnpm

# 安装PM2

1. 安装 pm2，$ cnpm install –g pm2（这个下载速度更快） | $ npm install -g

相关命令：

1.1 $ pm2 list 查看运行的 nodejs 服务

1.2 启动项目（aiovs）为例

进入 aiovs /bin/

$ pm2 start start.js

1.3 停止项目 pm2 stop 实例ID

1.4 重启项目 $ pm2 restart 0（实例ID）

# 部署信息

1. 后端部署位置：/home/app/appServer/pallet/api
2. 前端部署位置：/home/app/appServer/ pallet/web
3. 部署前备份位置：/home/app/appServer/backup

# 启动/停止后台API

1. 进入后端部署位置：
2. 启动运行命令 ：./startUp.sh start
3. 停止运行命令：./startUp.sh stop
4. 重启运行命令：./startUp.sh restart

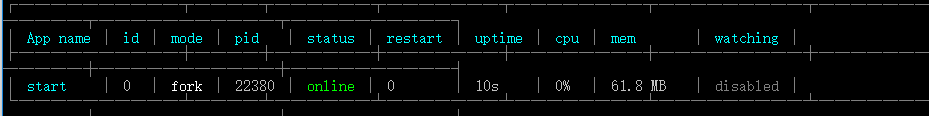
# 启动/停止前端系统

1. $进入目录web
2. $ cnpm install 安装项目依赖
3. 启动 $ pm2 start pm2.json

在 linux 下启动有可能会出现 underscore 这个模块找不到.

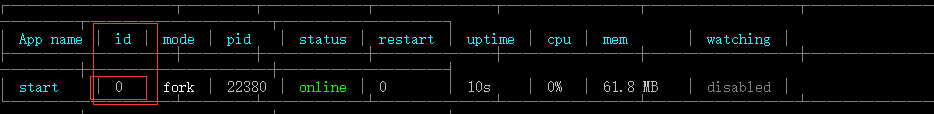
进入 ../node\_modules 重命名 underscore 把首字母 u 改为大写即可

4 $ pm2 list 查看运行的实例



5 停止前端系统 。

$ pm2 stop 0（前端服务的id）



6 重启前端系统

$pm2 restart 0 （前端服务的id）