# Linux

## Iptables

1. 以bridge网络模式运行Docker容器需要的防火墙配置

|  |
| --- |
| \*nat  :PREROUTING ACCEPT [27:11935]  :INPUT ACCEPT [0:0]  :OUTPUT ACCEPT [598:57368]  :POSTROUTING ACCEPT [591:57092]  :DOCKER - [0:0]  -A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER  -A OUTPUT ! -d 127.0.0.0/8 -m addrtype --dst-type LOCAL -j DOCKER  -A POSTROUTING -s 172.17.0.0/16 ! -o docker0 -j MASQUERADE #docker网桥的子网  -A POSTROUTING -j MASQUERADE  COMMIT  # Completed on Sun Sep 20 17:35:31 2015  # Generated by iptables-save v1.4.21 on Sun Sep 20 17:35:31 2015  \*filter  :INPUT ACCEPT [139291:461018923]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [127386:5251162]  :DOCKER - [0:0]  -A FORWARD -o docker0 -j DOCKER  -A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT  -A FORWARD -i docker0 ! -o docker0 -j ACCEPT  -A FORWARD -i docker0 -o docker0 -j ACCEPT  COMMIT |

1. 开放某范围内的防火墙端口

添加如下防火墙规则：

开放22端口

|  |
| --- |
| -A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT |

开放7000-8000之间的所有端口

|  |
| --- |
| -A INPUT -p tcp -m state --state NEW -m tcp --dport 7000:8000 -j ACCEPT |

开放7000及其以下的所有端口

|  |
| --- |
| -A INPUT -p tcp -m state --state NEW -m tcp --dport :7000 -j ACCEPT |

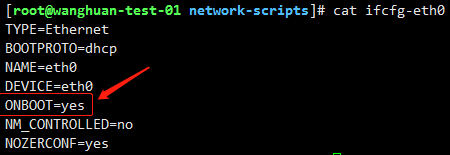
开放8000及其以上的所有端口

|  |
| --- |
| -A INPUT -p tcp -m state --state NEW -m tcp --dport 8000: -j ACCEPT |

1. 卡卡

## CentOS 7虚拟机有网卡，没有IP地址

进入/etc/sysconfig/network-scripts目录下，修改ifcfg-eth0文件

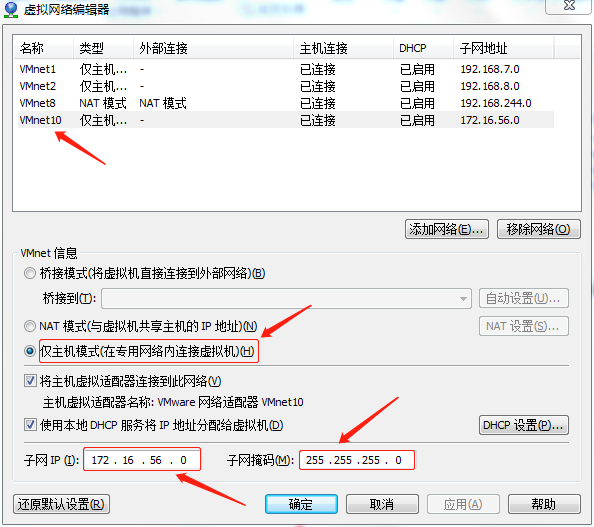


将其中的ONBOOT修改为“yes”，然后重启网络（systemctl restart network）

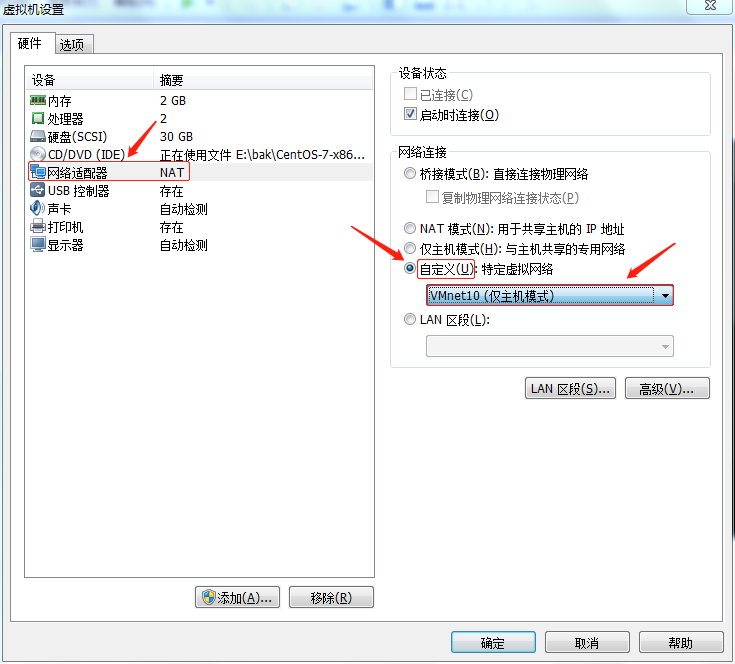
注意：启动Windows下VMware相关的服务

## CentOS 7虚拟机设置静态IP

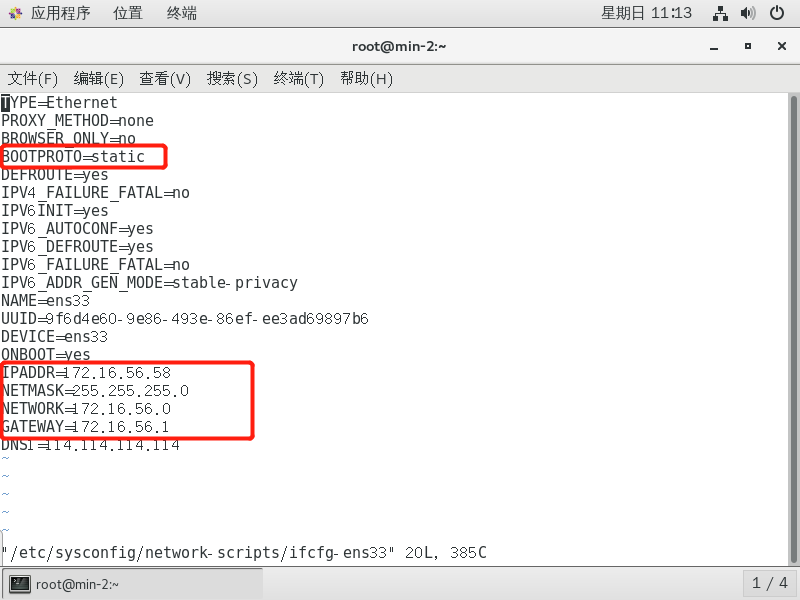
1. 在VMware中，编辑🡪虚拟网络编辑器🡪添加网络，选择合适的网络名称，例如VMnet10，点击“确定”
2. 点击“VMnet10”，选择“仅主机模式”，填写子网和子网掩码，例如子网为：172.16.56.0，子网掩码为：255.255.255.0，点击“应用”。



1. 选择要修改IP地址的虚拟机，点击“编辑虚拟机设置”，点击“网络适配器”，在“网络连接”中选择“自定义”，再选择“VMnet10 仅主机模式”，点击“确定”。



1. 登录虚拟机，修改/etc/sysconfig/network-scripts目录下ifcfg-ens33，修改如下内容，修改BOOTPROTO为static，IPADDR为指定的IP地址，例如172.16.56.58，子网NETWORK为172.16.56.0，子网掩码NETMASK为：255.255.255.0，网关GATEWAY为：172.16.56.1。



重新启动网络，systemctl restart network

## 定时任务

1. 每分钟向/home/xdhuxc.txt中输出“Hello World”

|  |
| --- |
| crontab -e  \*/1 \* \* \* \* echo "" > /home/xdhuxc.txt  systemctl restart crond |

1. 咔咔咔
2. 卡卡

<https://www.cnblogs.com/zoulongbin/p/6187238.html>

## NFS

<https://www.cnblogs.com/liuyisai/p/5992511.html>

<http://blog.51cto.com/lzhnb/2086392s>

## Systemd

1. 检查docker服务环境变量是否加载

|  |
| --- |
| systemctl show docker --property Environment |

1. 异步启动服务

|  |
| --- |
| systemctl --no-block start etcd |

## Rpm命令

1. 查看某个rpm包的依赖关系

|  |
| --- |
| rpm -qpR docker-ce-17.12.1.ce-1.el7.centos.x86\_64.rpm |

1. 卡卡

## DNS

1. 配置CentOS不更新DNS，即不刷新/etc/resolv.conf文件

修改配置/etc/NetworkManager/NetworkManager.conf

添加dns=none

修改完成后，重启NetworkManager

|  |
| --- |
| Systemctl restart NetworkManager |

## 实用小命令

1. 将当前目录下所有文件中的127.0.0.1修改为172.20.15.29

|  |
| --- |
| sed -i "s/127.0.0.1/172.20.15.29/g" ./\* |

将当前目录下所有文件中的127.0.0.1或localhost修改为172.20.15.29

|  |
| --- |
| sed -i "s/\(127.0.0.1\|localhost\)/172.20.15.29/g" ./\* |

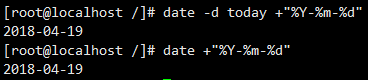
1. 查找当前目录下所有以.sh，.service，.conf，.yaml结尾的文件

|  |
| --- |
| find ./ -regex '.\*\.sh\|.\*\.service\|.\*\.conf\|.\*\.yaml' |

1. 时间格式
2. 仅包含年月日的日期格式

|  |
| --- |
| date +"%Y-%m-%d" 或 date -d today +"%Y-%m-%d" |

执行效果如下图所示：



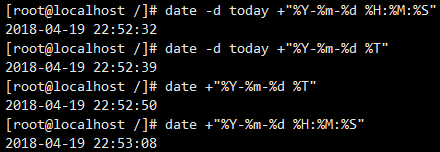
1. 包含日期和时间的格式

|  |
| --- |
| date +"%Y-%m-%d %H:%M:%S" 或 date +"%Y-%m-%d %T" |

或

|  |
| --- |
| date -d today +"%Y-%m-%d %T" 或 date -d today +"%Y-%m-%d %H:%M:%S" |

执行效果如下图所示：



## 时间同步

1. 将Linux系统时钟同步到远程NTP服务器，需要在系统上安装NTP以实现与NTP服务器的自动时间同步，要开始自动时间同步到远程NTP服务器，使用如下命令：

|  |
| --- |
| timedatectl set-ntp true |

执行该命令后，报错如下：



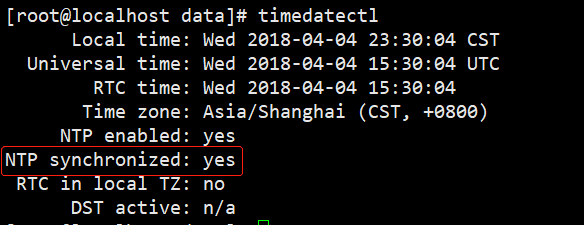
检查本机是否安装了ntp客户端，如果没有，使用如下命令安装：

|  |
| --- |
| yum install -y ntp |

重启ntp客户端并设置开机启动

|  |
| --- |
| systemctl restart ntpd  systemctl enable ntpd |

可看到NTP synchronized 已经为yes



要禁用NTP时间同步，使用如下命令：

|  |
| --- |
| timedatectl set-ntp false |

1. 同步若干台内网机器的时间

Vmstat

## 僵尸进程

1. 僵尸进程的清理方法

* 重启机器
* 杀死父进程

## 查看端口网络流量

iptraf 是一个基于ncurses开发的IP局域网监控工具，可以实时地监视网卡流量，可以生成各种网络统计数据，包括TCP信息、UDP统计、ICMP和OSPF信息、以太网负载信息、节点统计、IP校验和错误和其他一些信息。

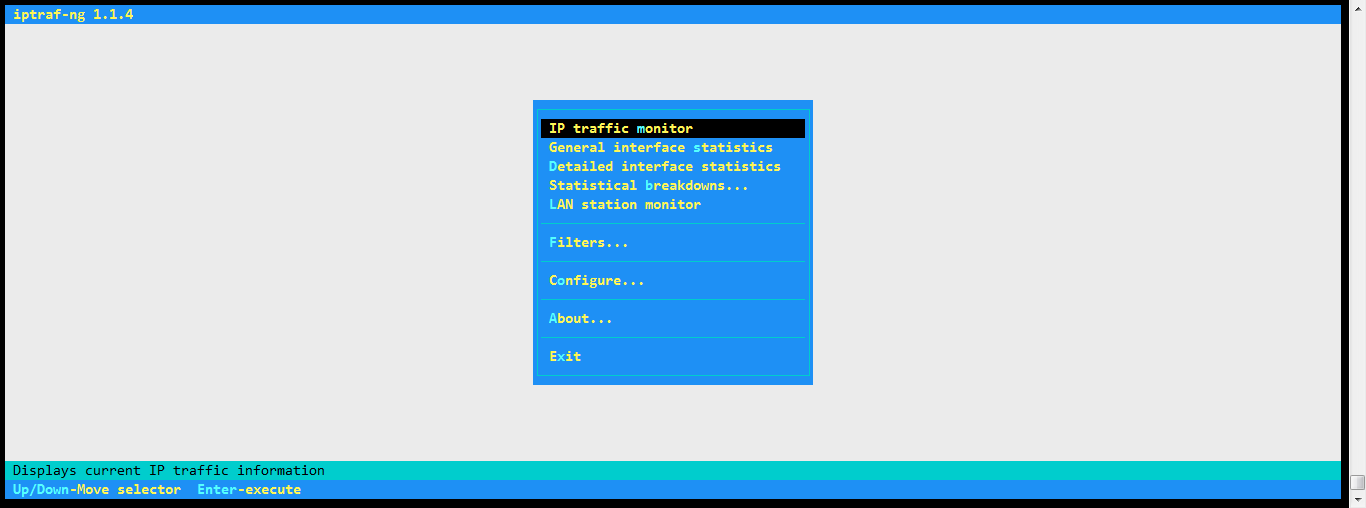
1. 安装iptraf

|  |
| --- |
| yum install -y iptraf-ng-1.1.4-6.el7.x86\_64 |

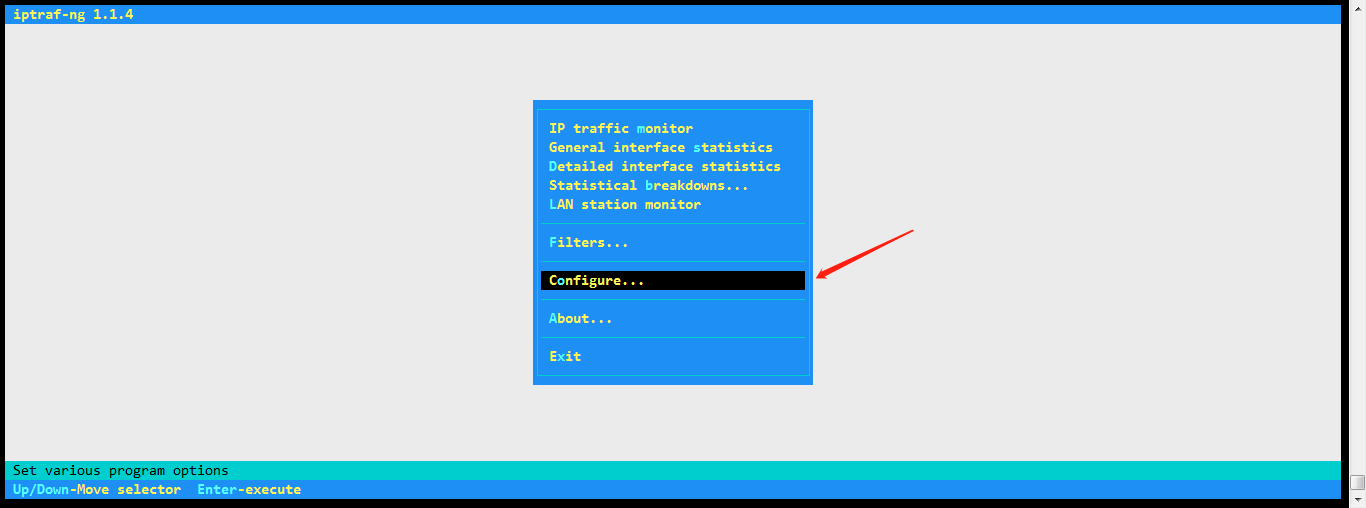
1. 启动iptraf-ng

|  |
| --- |
| iptraf-ng |

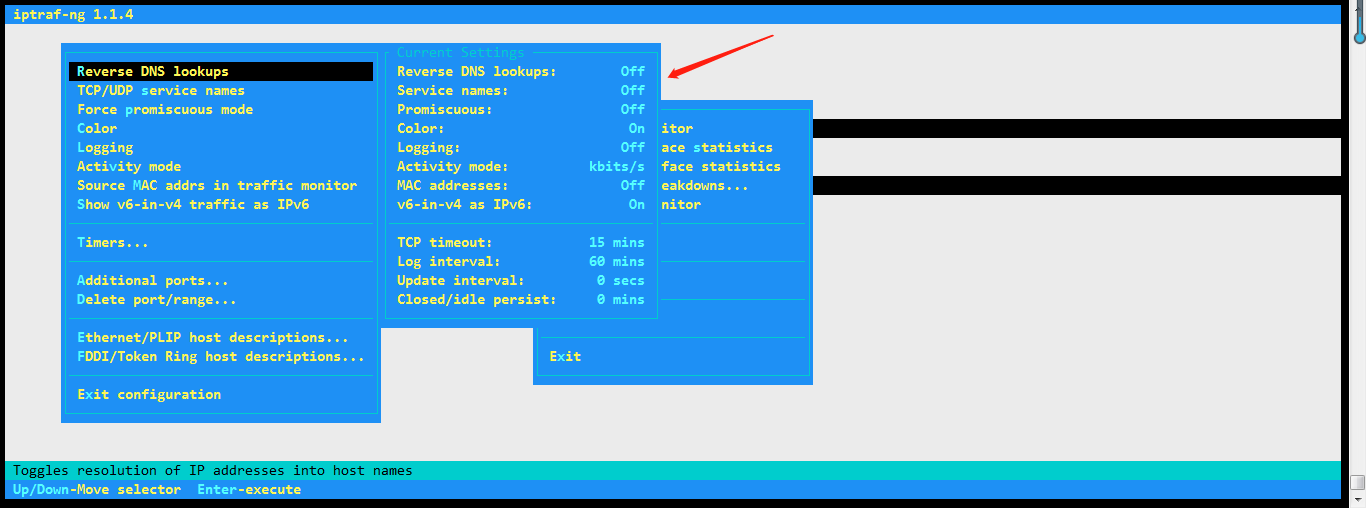
进入如下页面：



使用“↑”或“↓”，选择“Configure”



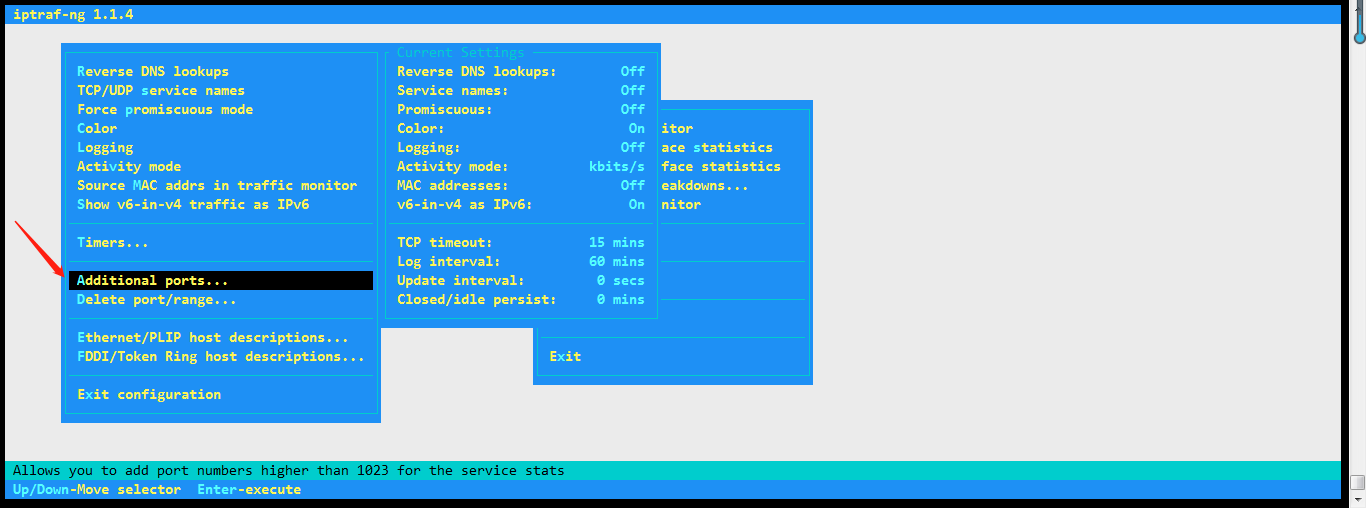
按“Enter”，进入设置页面，



进行适当的配置，可以让统计的结果更直观，信息更加丰富。

* Reverse DNS lookups：查看连接的IP所对应的域名，在IP traffic monitor的pkt captured 对话框中就可以看到域名结果，这个不是很直观，开启后会有点影响抓包性能。
* TCP/UDP service names：在有端口的地方都会把端口号换成相应的服务名，非常有用，很直观。
* Activity mode：显示流量是按Kbits/s还是Kbytes/s，改为Kbytes，更符合习惯。
* Additional ports：按端口号监控额外需要监控的端口，默认只监控小于1024的。

选择“Additional ports”

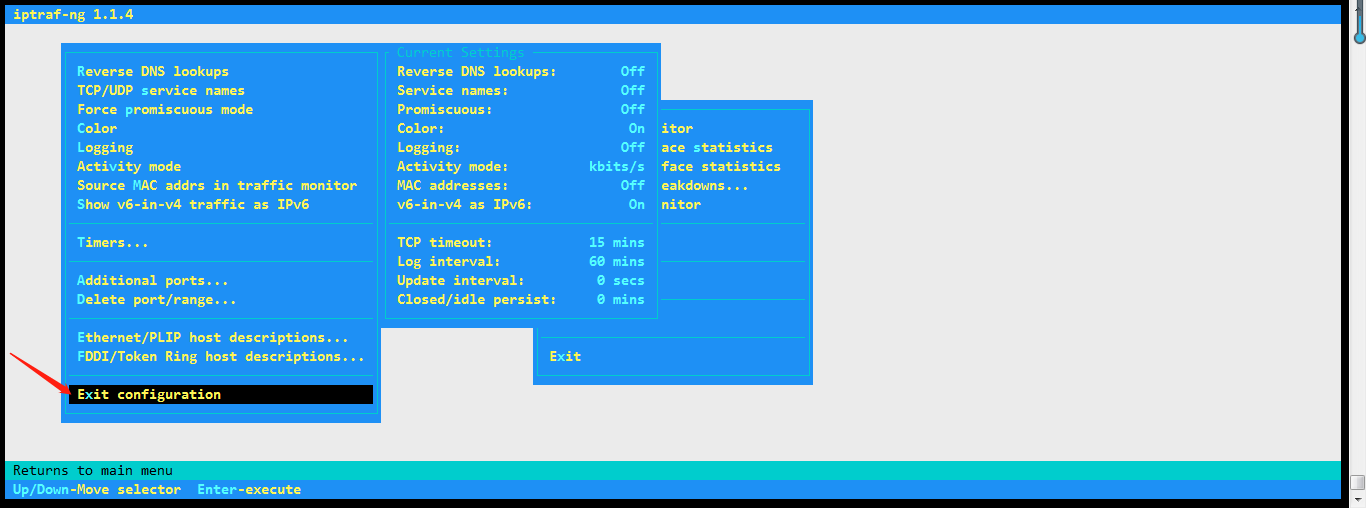


按“Enter”，进入端口设置页面，设置端口范围为：1024~10000，按“Tab”键跳转至第二个输入框。

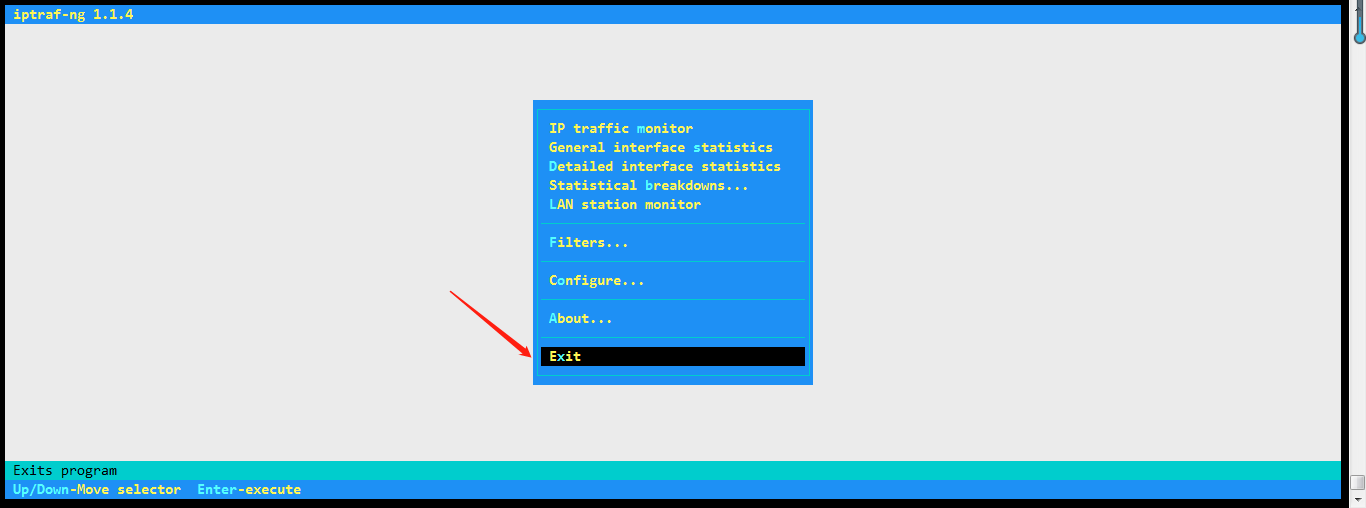


按“Enter”键保存端口设置。

然后退出配置页面



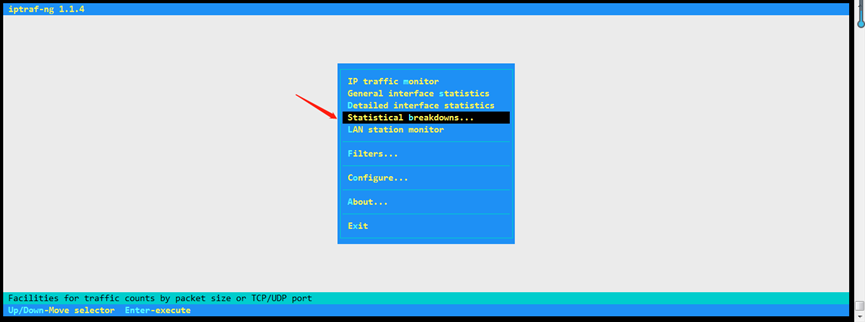
退出iptraf，



1. 重新打开iptraf

|  |
| --- |
| iptraf-ng |

选择“Statistical breakdowns”，



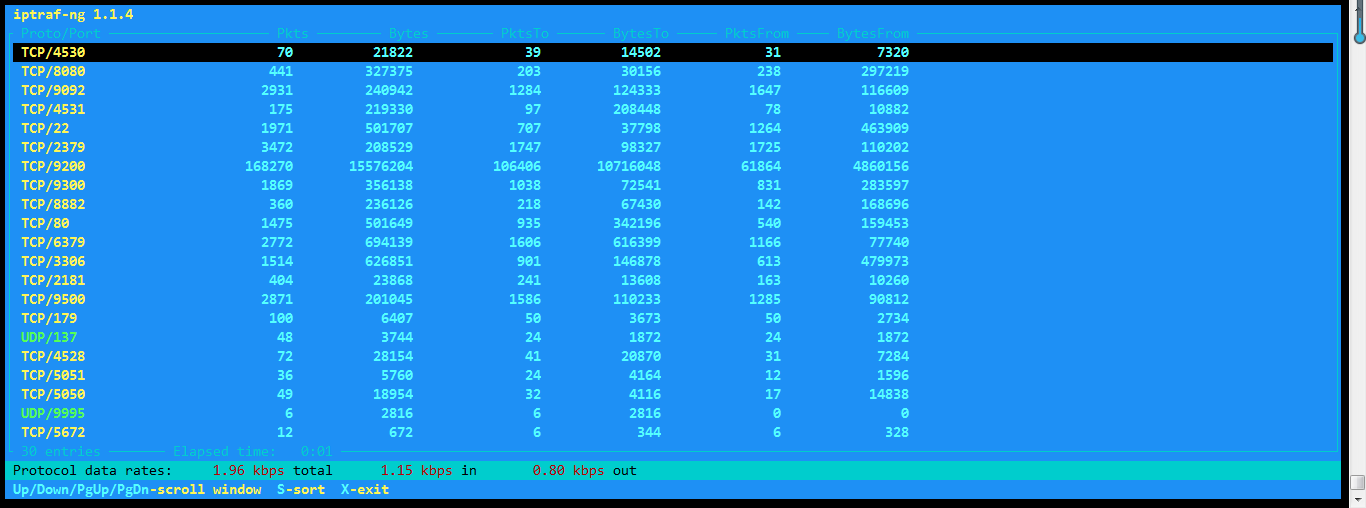
按“Enter”键，选择“By TCP/UDP port”



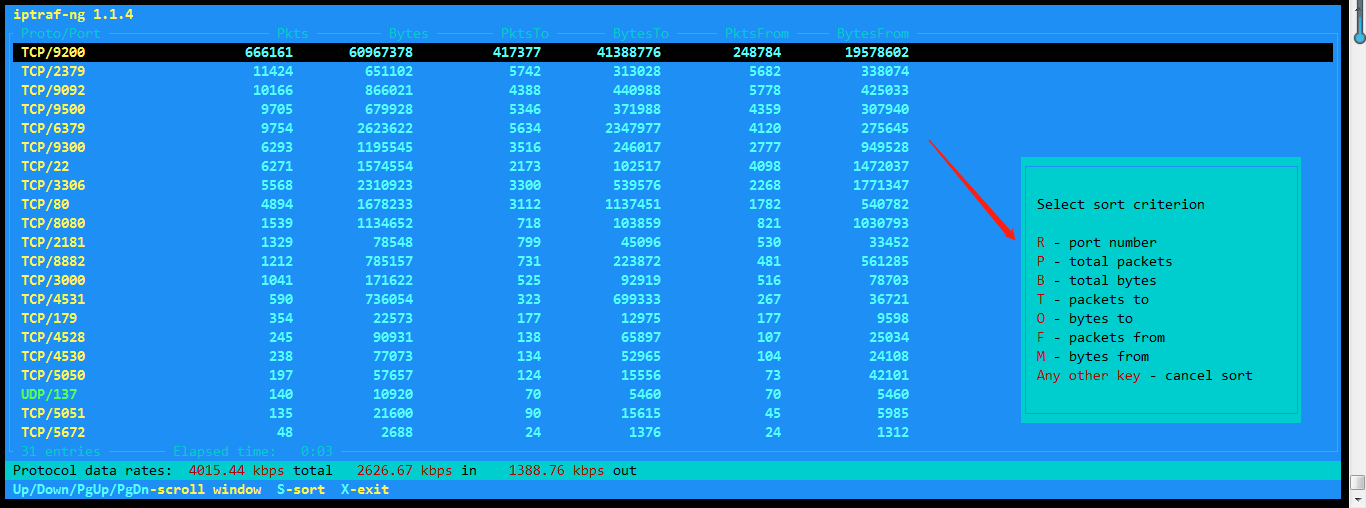
选择待检测的网卡“ens192”



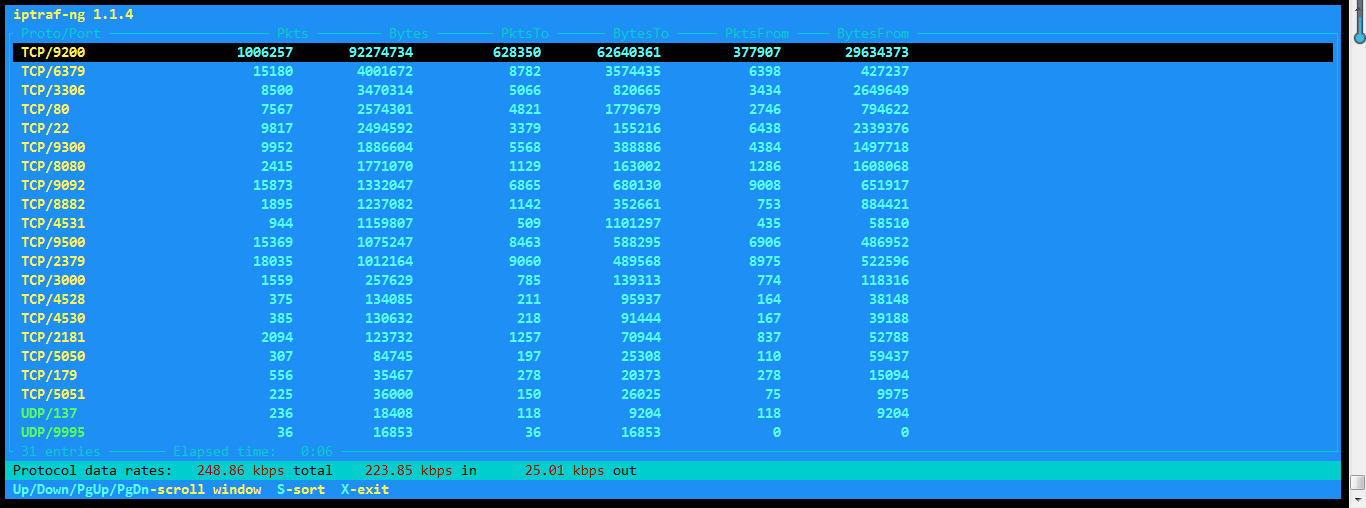
按“Enter”键，就可以看到此网卡上的端口对应的协议类型和网络流量了。



按“S”键，对输出信息进行排序



按“B”键，按照总的字节数进行排序



按“X”键退出。

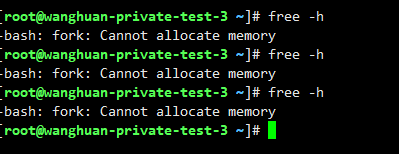
参考资料：

https://www.cnblogs.com/taosim/articles/4030292.html

https://blog.csdn.net/quiet\_girl/article/details/50777210

## 常见问题及解决

1. 使用SSH无法登陆服务器，VNC操作命令提示“fork:cannot allocate memory”，或者，在已经登录至该服务器上的终端中执行命令，所有命令均无法执行，一直提示如下错误：



查看最大进程数

|  |
| --- |
| sysctl kernel.pid\_max |

显示如下



查看进程数

|  |
| --- |
| ps -eLf|wc -l |

显示如下：



解决：

①修改最大进程数，即时生效

|  |
| --- |
| echo 1000000 > /proc/sys/kernel/pid\_max |

②永久生效的方式

|  |
| --- |
| echo "kernel.pid\_max=1000000 " >> /etc/sysctl.conf  sysctl -p |

# Shell

## If判断

### 字符串比较

字符串比较的作用：测试字符串是否相等，长度是否为零，字符串是否为NULL等

常用的字符串比较符号如下：

|  |  |
| --- | --- |
| 比较符 | 说明 |
| = | 比较两个字符串是否相同，相同则为“是” |
| != | 比较两个字符串是否不同，不同则为“是” |
| -n | 比较字符串的长度是否大于0，如果大于0则为“是” |
| -z | 比较字符串的长度是否等于0，如果等于0则为“是” |

### 数字比较

常用的数字比较符如下：

|  |  |
| --- | --- |
| 比较符 | 说明 |
| -eq | 等于 |
| -ge | 大于等于 |
| -le | 小于等于 |
| -ne | 不等于 |
| -gt | 大于 |
| -lt | 小于 |

<https://blog.csdn.net/zhouhong1026/article/details/8075687>

## 脚本示例

## 字符串截取

# Docker

## Docker相关的文件

1. docker.service
   1. docker.service文件路径

|  |
| --- |
| /usr/lib/systemd/system/docker.service |

* 1. docker.service文件内容

|  |
| --- |
| **[**Unit**]**  Description**=**Docker Application Container Engine  Documentation**=**http://docs.docker.com  After**=**network.target  Wants**=**docker-storage-setup.service  Requires**=**docker-cleanup.timer  **[**Service**]**  Type**=**notify  NotifyAccess**=**all  EnvironmentFile**=-/**run**/**containers**/**registries.conf  EnvironmentFile**=-/**etc**/**sysconfig**/**docker  EnvironmentFile**=-/**etc**/**sysconfig**/**docker-storage  EnvironmentFile**=-/**etc**/**sysconfig**/**docker-network  Environment**=**GOTRACEBACK**=**crash  Environment**=**DOCKER\_HTTP\_HOST\_COMPAT**=**1  Environment**=**PATH**=/**usr**/**libexec**/**docker**:/**usr**/**bin**:/**usr**/**sbin  ExecStart**=/**usr**/**bin**/**dockerd-current **\**  **--**add-runtime docker-runc**=/**usr**/**libexec**/**docker**/**docker-runc-current **\**  **--**default-runtime**=**docker-runc **\**  **--**exec-opt native.cgroupdriver**=**systemd **\**  **--**userland-proxy-path**=/**usr**/**libexec**/**docker**/**docker-proxy-current **\**  **$OPTIONS** **\**  **$DOCKER\_STORAGE\_OPTIONS** **\**  **$DOCKER\_NETWORK\_OPTIONS** **\**  **$ADD\_REGISTRY** **\**  **$BLOCK\_REGISTRY** **\**  **$INSECURE\_REGISTRY\**  **$REGISTRIES**  ExecReload**=/**bin**/**kill -s HUP **$MAINPID**  LimitNOFILE**=**1048576  LimitNPROC**=**1048576  LimitCORE**=**infinity  TimeoutStartSec**=**0  Restart**=**on-abnormal  MountFlags**=**slave  KillMode**=**process  **[**Install**]**  WantedBy**=**multi-user.target |

* 1. 启动docker时各参数的含义

|  |  |
| --- | --- |
| 参数 | 含义 |
| -g，--graph=“/var/lib/docker” | 设置docker的运行时根目录 |
| -s，--storage-driver=“devicemapper” | 设置容器运行时使用指定的存储驱动为devicemapper |
| --insecure-registry 0.0.0.0/0 | 允许不安全的仓库连接 |
| --registry-mirror ycr.yonyoucloud.com | 配置镜像加速地址 |
|  |  |
|  |  |
|  |  |

1. 日志文件

docker的日志文件写入写入到/var/log/message文件中，查看的命令为：

|  |
| --- |
| tail -f /var/log/message | grep docker |

1. daemon.json
   1. 文件路径为：/etc/docker/daemon.json
   2. 内容及含义：
2. 其他文件

## Docker修改空间大小

Docker默认空间大小分为两个，一个是池空间大小，默认为100G；另一个是容器空间大小，默认为10G

## 访问镜像仓库

1. 查看镜像仓库目录
2. 咔咔咔
3. 卡卡卡卡

## 限制日志文件大小和保留文件个数

在docker run命令中增加如下参数，限制生成的json.log单个文件大小和保留文件个数：

|  |
| --- |
| docker run -it --log-opt max-size=100m --log-opt max-file=3 nginx:latest |

在docker-compose.yml文件中的书写格式为：

|  |
| --- |
| log\_opt:  max-size: '100M'  max-file: '3' |

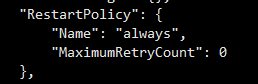
## 指定Cgroup Driver的类型

在/usr/lib/system/system/docker.service中给docker进程启动命令中增加如下参数：

|  |
| --- |
| --exec-opt native.cgroupdriver=systemd |

## Docker容器启动时参数

1. 重启策略



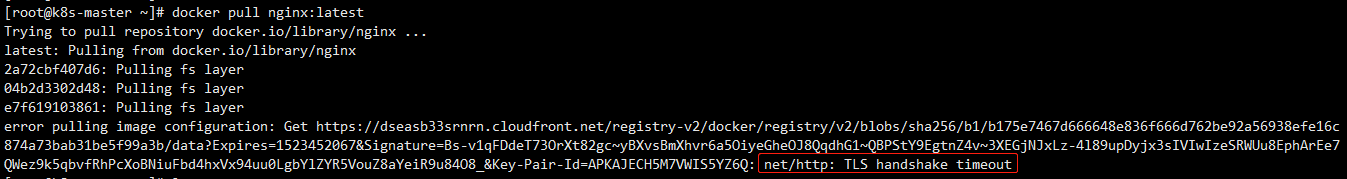
1. 咔咔咔

## 不常用的docker镜像使用方法

1. Elasticsearch
2. Fastdfs
3. Oracle
4. Weblogic

## 常见问题及解决方法

1. 在使用VMware workstation创建的虚拟机中，拉取docker镜像时，报错如下：



在一些实验室环境，服务器没有直接连接外网的权限，需要通过网络代理。通常会将网络代理直接配置在/etc/environment、/etc/profile之类的配置文件中，这对于大部分操作都是可行的。然而，docker命令却使用不了这些代理。比如docker pull时需要从外网下载镜像，就会出现上述错误。

配置docker网络代理

1. 为docker服务创建一个内嵌的systemd目录

|  |
| --- |
| mkdir -p /etc/systemd/system/docker.service.d |

1. 创建/etc/systemd/system/docker.service.d/http-proxy.conf文件，并添加HTTP\_PROXY环境变量。

|  |
| --- |
| **[**Service**]**  Environment**=**"HTTP\_PROXY=http://10.3.15.206:8888/" "HTTPS\_PROXY=http://10.3.15.206:8888/" |

10.3.15.206为代理的IP地址，8888为代理的端口

可以使用http代理https

1. 如果还有内部的不需要使用代理来访问的Docker registries，那么还需要指定NO\_PROXY环境变量

|  |
| --- |
| **[**Service**]**  Environment**=**"HTTP\_PROXY=http://[proxy-addr]:[proxy-port]/" "HTTPS\_PROXY=http://[proxy-addr]:[proxy-port]/" "NO\_PROXY=localhost,127.0.0.1,docker-registry.somecorporation.com" |

1. 更新配置并重启docker服务

|  |
| --- |
| systemctl daemon-reload  systemctl restart docker |

或者直接在/usr/lib/system/system/docker.service中添加环境变量，重启加载配置，启动docker服务。

1. 在docker.service中添加如下环境变量

|  |
| --- |
| Environment**=**HTTP\_PROXY**=**http://10.3.15.206:8888/  Environment**=**HTTPS\_PROXY**=**http://10.3.15.206:8888/ |

1. 重新加载配置并重启docker服务

|  |
| --- |
| systemctl daemon-reload  systemctl restart docker |

1. 使用docker save保存镜像，在另外的机器上重新加载该镜像，但是当docker pull该镜像时，还是会从docker store上拉取，并没有直接使用本地的，很可能是因为docker save和docker load时使用的docker的版本不一致。
2. 配置的镜像加速地址不生效，可能是因为同时配置了代理的缘故，取消代理的配置，重启docker
3. 容器退出状态码及原因：

|  |  |
| --- | --- |
| 状态码 | 原因 |
| 127 |  |
| 143 |  |
| 0 |  |
|  |  |
|  |  |

1. 咔咔咔

# Docker-compose

## Docker-compose安装

https://docs.docker.com/compose/compose-file/

## 中间件安装

### 启动kafka

docker-compose.yaml文件的内容为：

|  |
| --- |
| zookeeper:  image: yonyoucloud-middleware/zookeeper:3.4.10  restart: always  net: "host"  hostname: zookeeper  container\_name: zookeeper  ports:  - "2181:2181"  environment:  ZOO\_MY\_ID: 1  ZOO\_SERVERS: server.1=zookeeper:2888:3888  volumes:  - ${LOGS\_CENTER}/zookeeper:/datalog  - ${DATA\_CENTER}/zookeeper:/data  kafka:  privileged: true  image: wurstmeister/kafka:latest  restart: always  networks: "host"  ports:  - "9092:9092"  - "9094:9094"  container\_name: kafka  environment:  KAFKA\_ADVERTISED\_HOST\_NAME: 10.10.24.75  KAFKA\_MESSAGE\_MAX\_BYTES: "2000000"  KAFKA\_ZOOKEEPER\_CONNECT: ${LOCAL\_IP}:2181  log-opt:  max-size: '50M'  max-file: '1'  volumes:  - /var/run/docker.sock:/var/run/docker.sock |

# MySQL

## 数据库和数据表的授权

|  |
| --- |
| GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'\*' IDENTIFIED BY 'Ufsoft\*123' WITH GRANT OPTION; |

## my.cnf配置文件

|  |
| --- |
| [mysqld]  character-set-server = utf8  collation-server = utf8\_bin  max\_allowed\_packet = 20M  [mysql]  default-character-set = utf8 |

<https://blog.csdn.net/solmyr_biti/article/details/50674077>

## 启动MySQL容器

|  |
| --- |
| docker run -d \  --restart=always \  --privileged=true \  -p 3306:3306 \  -e MYSQL\_ROOT\_PASSWORD=Ufsoft\*123 \  -v /data/mysql\_data:/var/lib/mysql \  -v /data/disc/script/sql:/docker-entrypoint-initdb.d \  --name mysql \  mysql:5.7.21 |

该启动命令包括初始化root用户、初始化数据库（将待导入的sql文件放入**/**data**/**disc**/**script**/**sql目录下）和持久化数据目录。

## 数据库操作

1. 创建数据库

|  |
| --- |
| DROP DATABASE IF EXISTS `disconf`;  CREATE DATABASE IF NOT EXISTS `disconf` DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci; |

1. 导出数据和表结构

|  |
| --- |
|  |

1. 导出数据

|  |
| --- |
|  |

1. 导出表结构

|  |
| --- |
| mysqldump -uuser\_name -ppassword -d database\_name > /tmp/database\_name.sql;  或  mysqldump -uuser\_name -p -d database\_name > /tmp/database\_name.sql，敲回车后，会提示输入密码  或  mysqldump -uuser\_name -d database\_name > /tmp/database\_name.sql; |

## MySQL数据库调优

分库分表

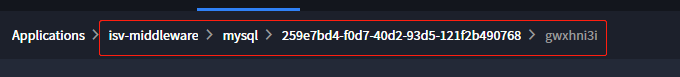
高可用，主从复制集群，主写从读

# Tomcat

# Marathon-lb

1. 域名解析配置

目录结构如下：



连接配置：

|  |
| --- |
| ping gwxhni3i.259e7bd4-f0d7-40d2-93d5-121f2b490768.mysql.isv-middleware.marathon.mesos |

# Nginx配置

情景描述：

marathon平时是带端口访问的，现在需要将其统一到80端口访问，使用nginx作为代理。

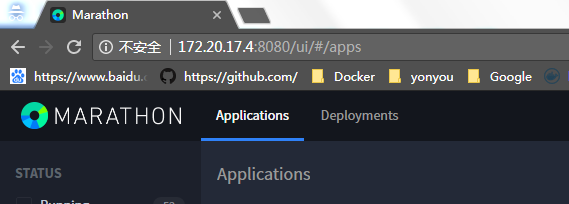
一般情况下，访问地址为：

|  |
| --- |
| http://172.20.17.4:8080 |

返回的显示地址为：

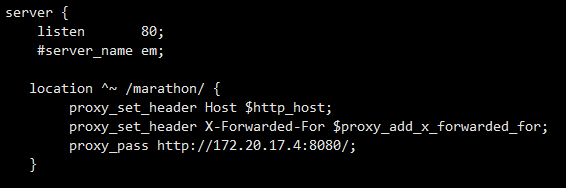
|  |
| --- |
| http://172.20.17.4:8080/ui/#/apps |

浏览器的地址栏显示为：



注意到，地址栏中变动的部分在 /ui/#/ 之后

在nginx中配置如下：



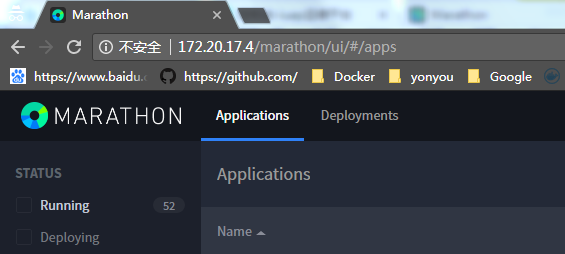
8080后面的斜杠至关重要。

访问如下地址

|  |
| --- |
| http://172.20.17.4/marathon |

浏览器地址栏显示：

|  |
| --- |
| http://172.20.17.4/marathon/ui/#/apps |

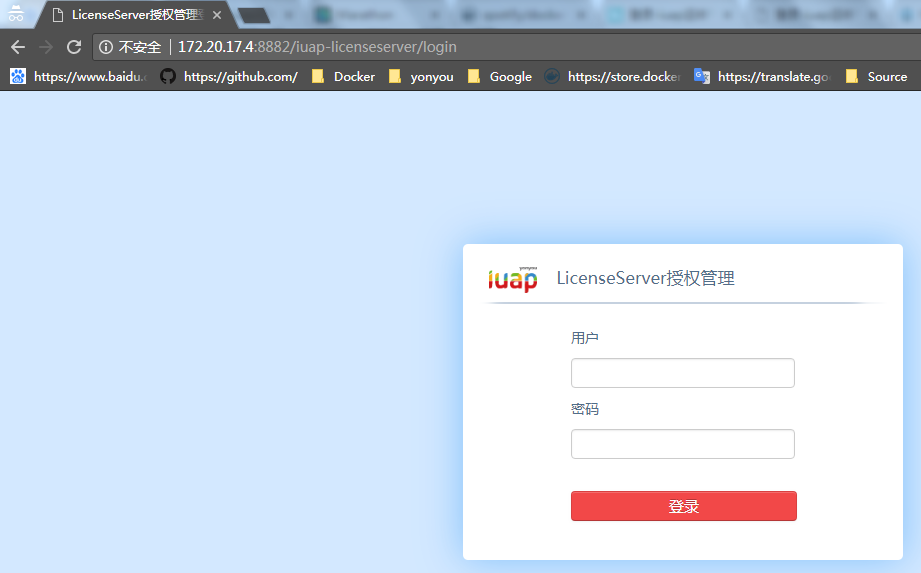


对于如下访问地址：

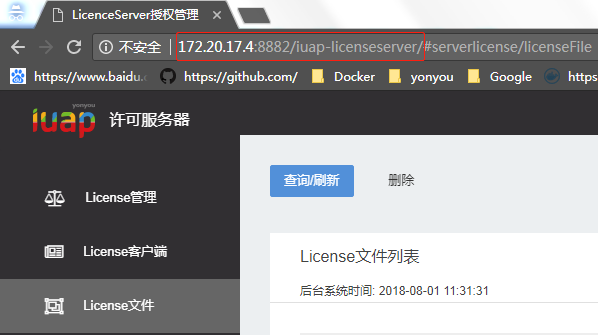
|  |
| --- |
| http://172.20.17.4:8882/iuap-licenseserver/ |

访问后浏览器地址显示如下：

|  |
| --- |
| http://172.20.17.4:8882/iuap-licenseserver/login |

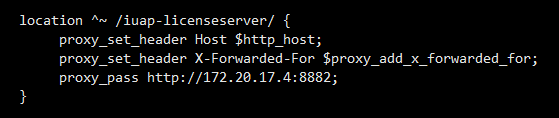


登录后地址如下：



注意到，地址中变动的部分始终在在 /# 后面

Nginx中配置如下：

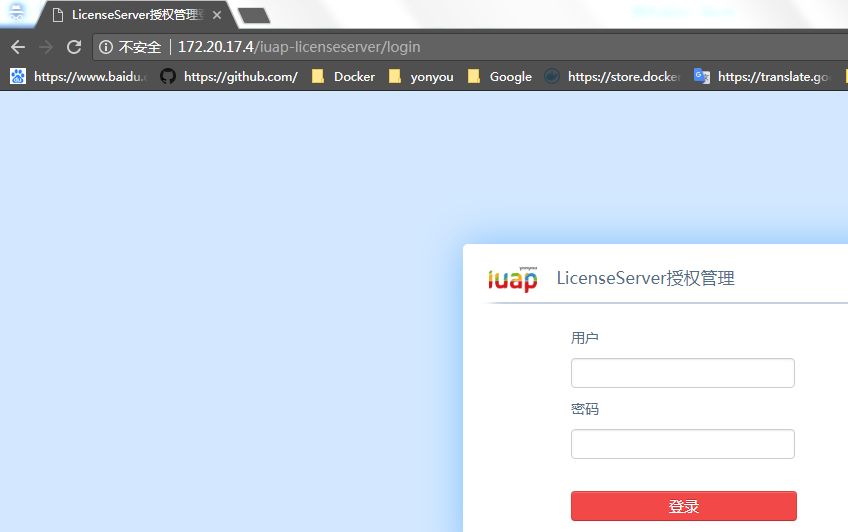


重启nginx后，访问如下路径：

|  |
| --- |
| http://172.20.17.4/iuap-licenseserver |

则会跳转至如下路径：

|  |
| --- |
| http://172.20.17.4/iuap-licenseserver/login |



# Elasticsearch

Elasticsearch的索引不能直接删除机器上的文件，只能通过页面手动删除索引或者通过curl执行rest api删除

# Jlogstash

让elasticsearch按月、按日建立索引，修改syslog.yml中index的模板，

按月：index: 'doclog-%{D\_LOG\_ISV}-%{+YYYY.MM}'

按日：index: 'doclog-%{D\_LOG\_ISV}-%{+YYYY.MM.dd}'

# Sqlite

## 下载和安装

<https://blog.csdn.net/yanbober/article/details/45567149>

http://blog.jobbole.com/92796/

## 基本命令

|  |  |
| --- | --- |
| 命令 | 描述 |
| .databases | 列出数据库的名称及其所依附的文件 |
| .exit | 退出 SQLite 提示符 |
| .quit | 退出 SQLite 提示符 |
|  |  |

# Redis

https://blog.csdn.net/yjqyyjw/article/details/73293455

# Nginx

https://blog.csdn.net/chenweijiSun/article/details/70823482

# Zookeeper

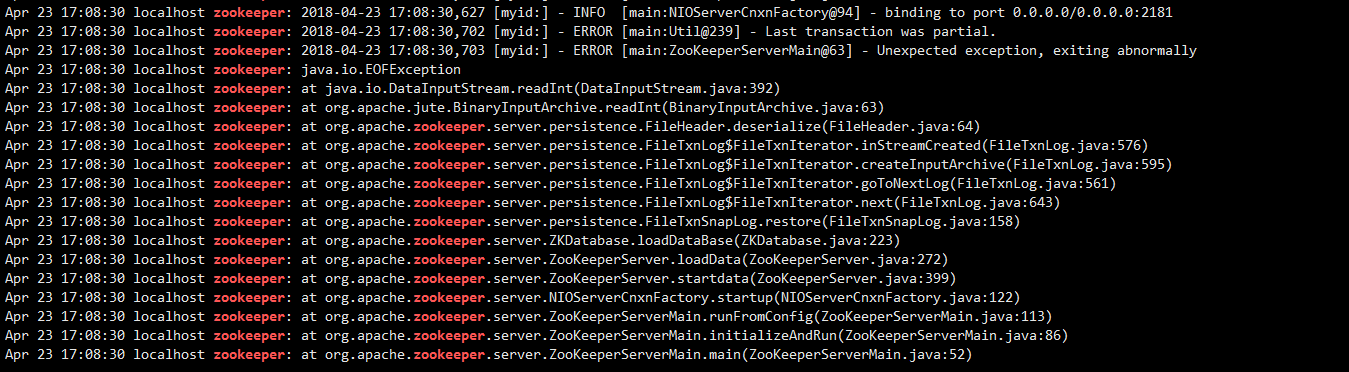
Zookeeper

<https://blog.csdn.net/hongweigg/article/details/52933113>

## 常见问题及解决方法

1. 重启zookeeper时报错如下：

|  |
| --- |
| Apr 23 17:08:30 localhost zookeeper: 2018-04-23 17:08:30,702 [myid:] - ERROR [main:Util@239] - Last transaction was partial.  Apr 23 17:08:30 localhost zookeeper: 2018-04-23 17:08:30,703 [myid:] - ERROR [main:ZooKeeperServerMain@63] - Unexpected exception, exiting abnormally  Apr 23 17:08:30 localhost zookeeper: java.io.EOFException  Apr 23 17:08:30 localhost zookeeper: at java.io.DataInputStream.readInt(DataInputStream.java:392)  Apr 23 17:08:30 localhost zookeeper: at org.apache.jute.BinaryInputArchive.readInt(BinaryInputArchive.java:63) |



解决：找到/etc/zookeeper/conf/zoo.cfg文件，找到zookeeper的数据目录，删除该数据目录，重新启动zookeeper即可。

1. 卡卡

# Kafka

https://github.com/wurstmeister/kafka-docker

# Etcd

Etcd镜像地址：<https://quay.io/repository/coreos/etcd?tag=latest&tab=tags>

使用地址：<https://github.com/coreos/etcd/blob/master/Documentation/op-guide/container.md#docker>

<https://blog.csdn.net/whackw/article/details/52701253>

## 常用命令

1. 设置键值对

|  |
| --- |
| etcdctl set xdhuxc adobe |

或

|  |
| --- |
| etcdctl set /xdhuxc adobe |

或

|  |
| --- |
| curl -X PUT http://10.10.24.75:2379/v2/keys/xdhuxc -d value="adobe" |

1. 获取所有的key

|  |
| --- |
| curl -s http://10.10.24.75:2379/v2/keys | python -m json.tool |

1. 获取键值对

|  |
| --- |
| etcdctl get xdhuxc |

或

|  |
| --- |
| etcdctl get **/**xdhuxc |

或

|  |
| --- |
| curl -s http://10.10.24.75:2379/v2/keys/xdhuxc | python -m json.tool |

1. 删除键值对

|  |
| --- |
| etcdctl rm xdhuxc |

或

|  |
| --- |
| etcdctl rm /xdhuxc |

或

|  |
| --- |
| curl -X –s DELETE http://127.0.0.1:2379/v2/keys/xdhuxc |

1. 查看当前集群中的节点和leader节点

|  |
| --- |
| etcdctl member list |

或

|  |
| --- |
| curl -s http://172.20.26.149:2379/v2/members|python -m json.tool |

1. 查看etcd节点状态

|  |
| --- |
| etcdctl cluster-health |

1. 卡啊啊啊

# flannld

# Calico

# 安装部署calico

Calicoctl下载地址：https://github.com/projectcalico/calicoctl/releases/

<https://docs.projectcalico.org/v2.6/getting-started/docker/>

镜像下载地址：

https://quay.io/repository/calico/node?tab=tags

https://blog.csdn.net/felix\_yujing/article/details/55213239

https://www.cnblogs.com/kevingrace/p/6864804.html?utm\_source=itdadao&utm\_medium=referral

查看IP池

calicoctl get ippool -o wide

资源定义：

https://docs.projectcalico.org/v2.0/reference/calicoctl/resources/

# RabbitMQ

<https://packagecloud.io/rabbitmq/rabbitmq-server/install#manual-rpm>

# grafana

<http://docs.grafana.org/installation/configuration/>

sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3000

# Maven

## 配置maven仓库

配置阿里云的maven仓库，在windows系统中，在apache-maven-3.5.2/conf/settings.xml中配置如下

|  |
| --- |
| **<**mirrors**>**  **<**mirror**>**  **<**id**>**alimaven**</**id**>**  **<**name**>**aliyun maven**</**name**>**  **<**url**>**http://maven.aliyun.com/nexus/content/groups/public/**</**url**>**  **<**mirrorOf**>**central**</**mirrorOf**>**  **</**mirror**>**  **</**mirrors**>** |

或者在pom.xml的repositories中添加如下配置

|  |
| --- |
| **<**repositories**>**  **<**repository**>**  **<**id**>**aliyunmaven**</**id**>**  **<**url**>**http://maven.aliyun.com/nexus/content/groups/public/**</**url**>**  **</**repository**>**  **</**repositories**>** |

## Properties的配置

|  |
| --- |
| **<**properties**>**  **<**project.build.sourceEncoding**>**UTF-8**</**project.build.sourceEncoding**>**  **<**project.reporting.outputEncoding**>**UTF-8**</**project.reporting.outputEncoding**>**  **<**java.version**>**1**.**8**</**java.version**>**  **</**properties**>** |

## 设置maven工程使用的java版本

1. 全局配置，在maven的安装目录下，apache-maven-3.5.2/conf目录下的settings.xml文件中，在<profiles></profiles>之间添加如下内容：

|  |
| --- |
| **<**profile**>**  **<**id**>**jdk18**</**id**>**  **<**activation**>**  **<**activeByDefault**>**true**</**activeByDefault**>**  **<**jdk**>**1**.**8**</**jdk**>**  **</**activation**>**  **<**properties**>**  **<**maven.compiler.source**>**1**.**8**</**maven.compiler.source**>**  **<**maven.compiler.target**>**1**.**8**</**maven.compiler.target**>**  **<**maven.compiler.compilerVersion**>**1**.**8**</**maven.compiler.compilerVersion**>**  **</**properties**>**  **</**profile**>** |

1. 局部配置，在该工程的pom文件中，添加如下内容

|  |
| --- |
| **<**build**>**  **<**plugins**>**  **<**plugin**>**  **<**groupId**>**org.apache.maven.plugins**</**groupId**>**  **<**artifactId**>**maven-compiler-plugin**</**artifactId**>**  **<**configuration**>**  **<**source**>**1**.**8**</**source**>**  **<**target**>**1**.**8**</**target**>**  **</**configuration**>**  **</**plugin**>**  **</**plugins**>**  **</**build**>** |

## 常见问题及解决

1. pom.xml报错如下，提示如下：

|  |
| --- |
| web.xml is missing and **<**failOnMissingWebXml**>** is set to true |

在pom.xml文件中添加如下内容：

|  |
| --- |
| **<**build**>**  **<**plugins**>**  **<**plugin**>**  **<**groupId**>**org.apache.maven.plugins**</**groupId**>**  **<**artifactId**>**maven-war-plugin**</**artifactId**>**  **<**version**>**3**.**2**.**0**</**version**>**  **<**configuration**>**  **<**failOnMissingWebXml**>**false**</**failOnMissingWebXml**>**  **</**configuration**>**  **</**plugin**>**  **</**plugins**>**  **</**build**>** |

1. 咔咔咔

# Python

## 配置python源

在当前目录下建立~/.pip/pip.conf文件，添加如下内容

|  |
| --- |
| **[**global**]**  index-url **=** http://mirrors.aliyun.com/pypi/simple/  **[**install**]**  trusted-host**=**mirrors.aliyun.com |

在Windows系统中，在C:\Users\wanghuan\AppData\Local\pip目录下创建pip.ini文件，写入以上内容并保存即可。

# Redis

# Ansible

http://www.ansible.com.cn/docs/intro\_installation.html#githubansible

# Fabric

## Fabric简介

Fabric是一个Python的库，提供了丰富的同SSH交互的接口，可以用来在本地或远程机器上自动化、流水化地执行Shell命令，非常适合用来做应用的远程部署及系统维护。

## Fabirc安装

1. Python的版本必须是2.7以上
2. 使用pip来进行安装，命令如下

|  |
| --- |
| pip install fabric |

http://python.jobbole.com/87241/

http://python.jobbole.com/87241/

# 开发者中心私有化

## 常用命令

## 常见问题及解决方法

1. 添加主机后，marathon无法发布任务到该机器，Debug处提示譬如invalid信息

在资源池节点机器上，执行如下命令：

|  |
| --- |
| #!/bin/sh  cat > /etc/sysconfig/iptables <<EOF  \*nat  :PREROUTING ACCEPT [27:11935]  :INPUT ACCEPT [0:0]  :OUTPUT ACCEPT [598:57368]  :POSTROUTING ACCEPT [591:57092]  :DOCKER - [0:0]  -A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER  -A OUTPUT ! -d 127.0.0.0/8 -m addrtype --dst-type LOCAL -j DOCKER  -A POSTROUTING -s 172.17.0.0/16 ! -o docker0 -j MASQUERADE  -A POSTROUTING -j MASQUERADE  COMMIT  # Completed on Sun Sep 20 17:35:31 2015  # Generated by iptables-save v1.4.21 on Sun Sep 20 17:35:31 2015  \*filter  :INPUT ACCEPT [139291:461018923]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [127386:5251162]  :DOCKER - [0:0]  -A FORWARD -o docker0 -j DOCKER  -A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT  -A FORWARD -i docker0 ! -o docker0 -j ACCEPT  -A FORWARD -i docker0 -o docker0 -j ACCEPT  COMMIT  EOF  systemctl stop firewalld && systemctl disable firewalld  systemctl restart iptables || iptables-restore </etc/sysconfig/iptables |

1. Rabbitmq容器一直在重启，日志输出信息如下：

## 磁盘空间清理

1. 清除fastdfs中的war包和tar包，确保用户不会再下载这些文件再删除，需要手动删除，路径为：

|  |
| --- |
| **/**data**/**developercenter\_enterprise**/**data\_center**/**fastdfs**/**store\_path |

1. 清除elasticsearch的过期索引数据，路径为：

|  |
| --- |
| **/**data**/**developercenter\_enterprise**/**data\_center**/**elasticsearch**/**data**/**elasticsearch-cluster**/**nodes**/**0**/**indices |

1. 清理用户应用的过期日志文件，路径为：

|  |
| --- |
| **/**data**/**log\_center**/**syslog |

清理脚本为：

1. 清理docker垃圾数据
2. 删除已经退出的docker容器

|  |
| --- |
| docker ps -a|grep Exited|awk '{print $1}'|xargs docker rm |

1. mesos-dns和marathon容器的日志文件过大

在容器启动命令中增加如下参数，限制生成的json.log单个文件大小和保留文件个数：

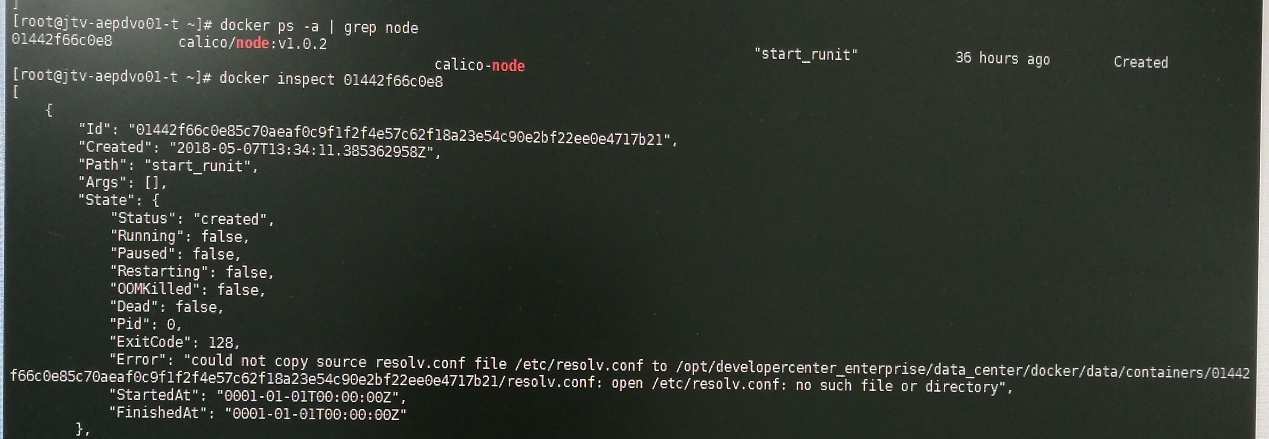
|  |
| --- |
| --log-opt max-size=100m --log-opt max-file=3 |

在docker-compose.yml文件中的书写格式为：

|  |
| --- |
| log\_opt:  max-size: '100M'  max-file: '3' |

当生成3个文件，每个文件达到100M时，会循环写入新的日志。

1. calico容器启动时，处于“Createed”状态，docker inspect后错误如下：



解决：重启calico容器即可。

1. 咔咔咔

# SaltStack

## SaltStack安装部署

https://www.cnblogs.com/lgeng/p/6567424.html

# 常见问题

长连接和短连接

Session和Cookies

# 其他问题

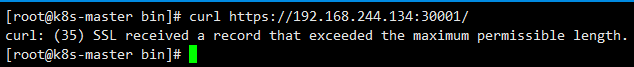
1. Google浏览器配置可访问自定义的https

Windows系统下，在chrome快捷方式的属性中，修改快捷方式中的目标，添加“--test-type --ignore-certificate-errors”，最后目标项的值为：

|  |
| --- |
| "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --test-type --ignore-certificate-errors |

然后使用该快捷方式打开浏览器，即可访问自定义的https地址。

1. 使用https访问时，报错如下：



1. 有状态和无状态

https://blog.csdn.net/romandion/article/details/1800025

1. 咔咔咔