Netkiller 手札系列电子书 http://www.netkiller.cn

Netkiller Monitor 手札

陈景峰 著

PROMETHEUS

知乎专栏 https://www.zhihu.com/column/netkiller

Netkiller Monitor 手札

目录

范例清单

Netkiller Monitor 手札

Prometheus, Zibbix, Cacti, Nagios, Scanner, Sniffer and Audit...

ISBN#

Mr. Neo Chan, 陈景峯(BG7NYT)

中国广东省深圳市望海路半岛城邦三期
518067
+86 13113668890

<netkiller@msn.com>

2010-11-18

电子书最近一次更新于 2021-09-28 15:36:19

http://www.netkiller.cn
http://netkiller.github.io
http://netkiller.sourceforge.net
微信订阅号 netkiller-ebook
微信：13113668890 请注明“读者”
QQ：13721218 请注明“读者”
QQ群：128659835 请注明“读者”

Netkiller 手札系列电子书 http://www.netkiller.cn

Netkiller Monitor 手札

陈景峰 著

PROMETHEUS

知乎专栏 https://www.zhihu.com/column/netkiller

知乎专栏 | 多维度架构

2017-02-13

致读者

Netkiller 系列手札 已经被 Github 收录，并备份保存在北极地下250米深的代码库中，备份会保留1000年。

Preserving open source software for future generations

The world is powered by open source software. It is a hidden cornerstone of modern civilization, and the shared heritage of all humanity.

The GitHub Arctic Code Vault is a data repository preserved in the Arctic World Archive (AWA), a very-long-term archival facility 250 meters deep in the permafrost of an Arctic mountain.

We are collaborating with the Bodleian Library in Oxford, the Bibliotheca Alexandrina in Egypt, and Stanford Libraries in California to store copies of 17,000 of GitHub's most popular and most-depended-upon projects—open source's "greatest hits"—in their archives, in museum-quality cases, to preserve them for future generations.

https://archiveprogram.github.com/arctic-vault/

自述

Netkiller 手札系列电子书 http://www.netkiller.cn

Netkiller Monitor 手札

陈景峰 著

知乎专栏 https://www.zhihu.com/column/netkiller

《Netkiller 系列 手札》是一套免费系列电子书，netkiller 是 nickname 从1999 开使用至今，“手札” 是札记，手册的含义。

2003年之前我还是以文章形式在BBS上发表各类技术文章，后来发现文章不够系统，便尝试写长篇技术文章加上章节目录等等。随着内容增加，不断修订，开始发布第一版，第二版......

IT知识变化非常快，而且具有时效性，这样发布非常混乱，经常有读者发现第一版例子已经过时，但他不知道我已经发布第二版。

我便有一种想法，始终维护一个文档，不断更新，使他保持较新的版本不过时。

第一部电子书是《PostgreSQL 实用实例参考》开始我使用Microsoft Office Word 慢慢随着文档尺寸增加 Word 开始表现出力不从心。

我看到PostgreSQL 中文手册使用SGML编写文档，便开始学习Docbook SGML。使用Docbook写的第一部电子书是《Netkiller Postfix Integrated Solution》这是Netkiller 系列手札的原型。

至于“手札”一词的来历，是因为我爱好摄影，经常去一个台湾摄影网站，名字就叫“摄影家手札”。

由于硬盘损坏数据丢失 《Netkiller Postfix Integrated Solution》 的 SGML文件已经不存在；Docbook SGML存在很多缺陷UTF-8支持不好，转而使用Docbook XML.

目前技术书籍的价格一路飙升，动则￥80，￥100，少则￥50，￥60. 技术书籍有时效性，随着技术的革新或淘汰，大批书记成为废纸垃圾。并且这些书技术内容雷同，相互抄袭，质量越来越差，甚至里面给出的例子错误百出，只能购买影印版，或者翻译的版本。

在这种背景下我便萌生了自己写书的想法，资料主要来源是我的笔记与例子。我并不想出版，只为分享，所有我制作了基于CC License 发行的系列电子书。

本书注重例子，少理论（捞干货），只要你对着例子一步一步操作，就会成功，会让你有成就感并能坚持学下去，因为很多人遇到障碍就会放弃，其实我就是这种人，只要让他看到希望，就能坚持下去。

1. 写给读者

为什么写这篇文章

有很多想法,工作中也用不到所以未能实现，所以想写出来,和大家分享.有一点写一点,写得也不好,只要能看懂就行,就当学习笔记了.

开始零零碎碎写过一些文档，也向维基百科供过稿，但维基经常被ZF封锁，后来发现sf.net可以提供主机存放文档，便做了迁移。并开始了我的写作生涯。

这篇文档是作者20年来对工作的总结,是作者一点一滴的积累起来的，有些笔记已经丢失，所以并不完整。

因为工作太忙整理比较缓慢。目前的工作涉及面比较窄所以新文档比较少。

我现在花在技术上的时间越来越少，兴趣转向摄影，无线电。也想写写摄影方面的心得体会。

写作动力:

曾经在网上看到外国开源界对中国的评价，中国人对开源索取无度，但贡献却微乎其微.这句话一直记在我心中，发誓要为中国开源事业做我仅有的一点微薄贡献

另外写文档也是知识积累，还可以增加在圈内的影响力.

人跟动物的不同,就是人类可以把自己学习的经验教给下一代人.下一代在上一代的基础上再创新,不断积累才有今天.

所以我把自己的经验写出来,可以让经验传承

没有内容的章节:

目前我自己一人维护所有文档，写作时间有限，当我发现一个好主题就会加入到文档中，待我有时间再完善章节，所以你会发现很多章节是空无内容的.

文档目前几乎是流水帐试的写作，维护量很大，先将就着看吧.

我想到哪写到哪,你会发现文章没一个中心,今天这里写点,明天跳过本

章写其它的.

文中例子绝对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.

理论的东西,网上大把,我这里就不写了,需要可以去网上查.

我爱写错别字,还有一些是打错的,如果发现请指正.

文中大部分试验是在Debian/Ubuntu/Redhat AS上完成.

写给读者

至读者:

我不知道什么时候，我不再更新文档或者退出IT行业去从事其他工作，我必须给这些文档找一个归宿，让他能持续更新下去。

我想捐赠给某些基金会继续运转，或者建立一个团队维护它。

我用了20年时间坚持不停地写作，持续更新，才有今天你看到的《Netkiller 手扎》系列文档，在中国能坚持20年，同时没有任何收益的技术类文档，是非常不容易的。

有很多时候想放弃，看到外国读者的支持与国内社区的影响，我坚持了下来。

中国开源事业需要各位参与，不要成为局外人，不要让外国人说:中国对开源索取无度，贡献却微乎其微。

我们参与内核的开发还比较遥远，但是进个人能力，写一些文档还是可能的。

系列文档

下面是我多年积累下来的经验总结，整理成文档供大家参考:

Netkiller Architect 手札

Netkiller Developer 手札

Netkiller PHP 手札

Netkiller Python 手札

Netkiller Testing 手札

Netkiller Cryptography 手札

Netkiller Linux 手札
Netkiller FreeBSD 手札
Netkiller Shell 手札
Netkiller Security 手札
Netkiller Web 手札
Netkiller Monitoring 手札
Netkiller Storage 手札
Netkiller Mail 手札
Netkiller Docbook 手札
Netkiller Version 手札
Netkiller Database 手札
Netkiller PostgreSQL 手札
Netkiller MySQL 手札
Netkiller NoSQL 手札
Netkiller LDAP 手札
Netkiller Network 手札
Netkiller Cisco IOS 手札
Netkiller H3C 手札
Netkiller Multimedia 手札
Netkiller Management 手札
Netkiller Spring 手札
Netkiller Perl 手札
Netkiller Amateur Radio 手札

2. 作者简介

陈景峯 (ㄔㄣ ㄐㄧㄥ ㄈㄥ)

Nickname： netkiller | English name: Neo chen | Nippon name: ちんけいほう (音訳) | Korean name: 천징봉 | Thailand name: ภูมิภาพภูเขา | Vietnam: Trần Cảnh Phong

Callsign: BG7NYT | QTH: ZONE CQ24 ITU44 ShenZhen, China

程序猿，攻城狮，挨踢民工, Full Stack Developer, UNIX like Evangelist, 业余无线电爱好者（呼号：BG7NYT）,户外运动，山地骑行以及摄影爱好者。

《Netkiller 系列 手札》的作者

成长阶段

1981年1月19日(庚申年腊月十四)出生于黑龙江省青冈县建设乡双富大队第一小队

1989年9岁随父母迁居至黑龙江省伊春市，悲剧的天朝教育，不知道那门子归定，转学必须降一级，我本应该上一年级，但体制让我上学前班，那年多都10岁了

1995年小学毕业，体制规定借读要交3000两银子(我曾想过不升初中)，亲戚单位分楼告别平房，楼里没有地方放东西，把2麻袋书送给我，无意中发现一本电脑书BASIC语言，我竟然看懂了，对于电脑知识追求一发而不可收，后面顶零花钱，压岁钱主要用来买电脑书《MSDOS 6.22》《新编Unix实用大全》《跟我学Foxbase》。。。。。。

1996年第一次接触UNIX操作系统，BSD UNIX, Microsoft Xinux(盖茨亲自写的微软Unix，知道的人不多)

1997年自学Turbo C语言，苦于没有电脑，后来学校建了微机室才第一次使用QBASIC(DOS 6.22 自带命令)，那个年代只能通过软盘拷贝转播，Trubo C编译器始终没有搞到，

1997年第一次上Internet网速只有9600Bps, 当时全国兴起各种信息港域名格式是www.xxxx.info.net, 访问的第一个网站是NASA下载了很多火星探路者拍回的照片，还有“淞沪”sohu的前身

1998~2000年在哈尔滨学习计算机，充足的上机时间，但老师让我们练打字（明伦五笔/WT）打字不超过80个/每分钟还要强化训练，不过这个给我的键盘功夫打了好底。

1999年学校的电脑终于安装了光驱，在一张工具盘上终于找到了Turbo C, Borland C++与Quick Basic编译器，当时对VGA图形编程非常感兴趣，通过INT33中断控制鼠标，使用绘图函数模仿windows界面。还有操作 UCDOS 中文字库，绘制矢量与点阵字体。

2000年沉迷于Windows NT与Back Office各种技术，神马主域控制器，DHCP，WINS，IIS，域名服务器，Exchange邮件服务器，MS Proxy, NetMeeting...以及ASP+MS SQL开发；用56K猫下载了一张LINUX。ISO镜像，安装后我兴奋的24小时没有睡觉。

职业生涯

2001 年来深圳进城打工,成为一名外来务工者. 在一个4人公司做PHP开发，当时PHP的版本是2.0, 开始使用Linux Redhat 6.2.当时很多门户网站都是用FreeBSD,但很难搞到安装盘，在网易社区认识了一个网友,从广州给我寄了一张光盘，FreeBSD 3.2

2002 年我发现不能埋头苦干,还要学会"做人".后辗转广州工作了半年，考了一个Cisco CCNA认证。回到深圳重新开始，在车公庙找到一家工作做Java开发

2003 年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004 年开始加入分布式计算团队,目前成绩，工作仍然是Java开发并且开始使用PostgreSQL数据库。

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员,呼号BG7NYT,进了一部Yaesu FT-60R手台。公司的需要转回PHP与MySQL，相隔几年发现PHP进步很大。在前台展现方面无人能敌，于是便前台使用PHP，后台采用Java开发。

2006 年单身生活了这么多年,终于找到归宿. 工作更多是研究PHP各种框架原理

2007 物价上涨,金融危机，休息了4个月（其实是找不到工作），关外很难上439.460中继，搞了一台Yaesu FT-7800.

2008 终于找到英文学习方法，《Netkiller Developer 手札》，《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009《Netkiller Database 手札》,2009-6-13学车，年底拿到C1驾照

2010 对电子打击乐产生兴趣，计划学习爵士鼓。由于我对Linux热爱，我轻松的接管了公司的运维部，然后开发运维两把抓。我印象最深刻的是公司一次上架10个机柜，我们用买服务器纸箱的钱改善伙食。我将40多台服务器安装BOINC做压力测试，获得了中国第二的名次。

2011 平凡的一年，户外运动停止，电台很少开，中继很少上，摄影主要是拍女儿与家人，年末买了一辆山地车

2012 对油笔画产生了兴趣，活动基本是骑行银湖山绿道，

2013 开始学习民谣吉他，同时对电吉他也极有兴趣；最终都放弃了。这一年深圳开始推数字中继2013-7-6日入手Motorola

MOTOTRBO XIR P8668，Netkiller 系列手机从Sourceforge向Github迁移；年底对MYSQL UDF，Engine与PHP扩展开发产生很浓的兴趣，拾起遗忘10+年的C，写了几个mysql扩展（图片处理，fifo管道与ZeroMQ），10月份入Toyota Rezi 2.5V并写了一篇《攻城狮的苦逼选车经历》

2014-9-8 在淘宝上买了一架电钢琴 Casio Privia PX-5S pro 开始陪女儿学习钢琴，由于这家钢琴是合成器电钢，里面有打击乐，我有对键盘鼓产生了兴趣。

2014-10-2号罗浮山两日游，对中国道教文化与音乐产生了兴趣，10月5号用了半天时间学会了简谱。10月8号入Canon 5D Mark III + Canon Speedlite 600EX-RT香港过关被查。

2014-12-20号对乐谱制作产生兴趣（https://github.com/SheetMusic/Piano），给女儿做了几首钢琴伴奏曲，MuseScore制谱然后生成MIDI与WAV文件。

2015-09-01 晚饭后拿起爵士鼓基础教程尝试在Casio Privia PX-5S pro演练，经过反复琢磨加上之前学钢琴的乐理知识，终于在02号晚上，打出了简单的基本节奏，迈出了第一步。

2016 对弓箭（复合弓）产生兴趣，无奈天朝法律法规不让玩。每周游泳轻松1500米无压力，年底入 xbox one s 和 Yaesu FT-2DR, 同时开始关注功放音响这块

2017 7月9号入 Yamaha RX-V581 功放一台，连接Xbox打游戏爽翻了，入Kindle电子书，计划学习蝶泳，果断放弃运维和开发知识体系转攻区块链。

2018 从溪山美地搬到半岛城邦，丢弃了多年攒下的家底。11 月开始玩 MMDVM，使用 Yaesu FT-7800 发射，连接MMDVM中继板，树莓派，覆盖深圳湾，散步骑车通联两不误。

2019 卖了常德的房子，住了5次院，哮喘反复发作，决定停止电子书更新，兴趣转到知乎，B站

2020 准备找工作

职业生涯路上继续打怪升级

3. 如何获得文档

下载 Netkiller 手札 (epub,kindle,chm,pdf)

EPUB https://github.com/netkiller/netkiller.github.io/tree/master/download/epub

MOBI https://github.com/netkiller/netkiller.github.io/tree/master/download/mobi

PDF https://github.com/netkiller/netkiller.github.io/tree/master/download/pdf

CHM https://github.com/netkiller/netkiller.github.io/tree/master/download/chm

通过 GIT 镜像整个网站

https://github.com/netkiller/netkiller.github.com.git

$ git clone https://github.com/netkiller/netkiller.github.com.git

镜像下载

整站下载

```
wget -m http://www.netkiller.cn/index.html
```

指定下载

```
wget -m wget -m http://www.netkiller.cn/linux/index.html
```

Yum 下载文档

获得光盘介质，RPM包，DEB包，如有特别需要，请联系我

YUM 在线安装电子书

http://netkiller.sourceforge.net/pub/repo/

```
# cat >> /etc/yum.repos.d/netkiller.repo <<EOF
[netkiller]
```

```
name=Netkiller Free Books
baseurl=http://netkiller.sourceforge.net/pub/repo/
enabled=1
gpgcheck=0
gpgkey=
EOF
```

查找包

```
# yum search netkiller

netkiller-centos.x86_64 : Netkiller centos Cookbook
netkiller-cryptography.x86_64 : Netkiller cryptography Cookbook
netkiller-docbook.x86_64 : Netkiller docbook Cookbook
netkiller-linux.x86_64 : Netkiller linux Cookbook
netkiller-mysql.x86_64 : Netkiller mysql Cookbook
netkiller-php.x86_64 : Netkiller php Cookbook
netkiller-postgresql.x86_64 : Netkiller postgresql Cookbook
netkiller-python.x86_64 : Netkiller python Cookbook
netkiller-version.x86_64 : Netkiller version Cookbook
```

安装包

```
yum install netkiller-docbook
```

4. 打赏（Donations）

If you like this documents, please make a donation to support the authors' efforts. Thank you!

您可以通过微信，支付宝，贝宝给作者打赏。

银行(Bank)

招商银行(China Merchants Bank)

开户名：陈景峰

账号：9555500000007459

微信（Wechat）

支付宝（Alipay）

PayPal Donations

https://www.paypal.me/netkiller

5. 联系方式

主站 http://www.netkiller.cn/

备用 http://netkiller.github.io/

繁体网站 http://netkiller.sourceforge.net/

联系作者

Mobile: +86 13113668890

Email: netkiller@msn.com

QQ群: 128659835 请注明“读者”

QQ: 13721218

ICQ: 101888222

注：请不要问我安装问题！

博客 Blogger

知乎专栏 https://zhuanlan.zhihu.com/netkiller

LinkedIn: http://cn.linkedin.com/in/netkiller

OSChina: http://my.oschina.net/neochen/

Facebook: https://www.facebook.com/bg7nyt

Flickr: http://www.flickr.com/photos/bg7nyt/

Disqus: http://disqus.com/netkiller/

solidot: http://solidot.org/~netkiller/

SegmentFault: https://segmentfault.com/u/netkiller

Reddit: https://www.reddit.com/user/netkiller/

Digg: http://www.digg.com/netkiller

Twitter: http://twitter.com/bg7nyt

weibo: http://weibo.com/bg7nyt

Xbox club

我的 xbox 上的ID是 netkiller xbox，我创建了一个俱乐部 netkiller 欢迎加入。

Radio

CQ CQ CQ DE BG7NYT:

如果这篇文章对你有所帮助,请寄给我一张QSL卡片, qrz.cn or qrz.com or hamcall.net

Personal Amateur Radiostations of P.R.China

ZONE CQ24 ITU44 ShenZhen, China

Best Regards, VY 73! OP. BG7NYT

守听频率 DMR 438.460 -8 Color 12 Slot 2 Group 46001

守听频率 C4FM 439.360 -5 DN/VW

MMDVM Hotspot:

Callsign: BG7NYT QTH: Shenzhen, China

YSF: YSF80337 - CN China 1 - W24166/TG46001

DMR: BM_China_46001 - DMR Radio ID: 4600441

第 1 章 Prometheus

1. 安装 Prometheus

1.1. Docker 安装

```
docker run -d -p 9090:9090 -v
~/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus -
config.file=/etc/prometheus/prometheus.yml -
storage.local.path=/prometheus -storage.local.memory-chunks=10000
```

```
docker run -d -p 9100:9100 --user 995:995 \
-v "/:/hostfs" \
--net="host" \
prom/node-exporter \
--path.rootfs=/hostfs
```

检查 node-exporter 是否正常工作

```
$ curl http://localhost:9100/metrics
```

安装 grafana

```
$ docker run -d --name grafana -p 3000:3000 --net=host -e
"GF_SECURITY_ADMIN_PASSWORD=passw0rd" grafana/grafana
```

-e "GF_SERVER_ROOT_URL=http://grafana.server.name"

```
docker exec -it grafana cat /etc/grafana/grafana.ini > grafana.ini
```

环境变量配置的默认路径

```
环境变量                                        默认值
GF_PATHS_CONFIG                 /etc/grafana/grafana.ini
GF_PATHS_DATA                   /var/lib/grafana
GF_PATHS_HOME                   /usr/share/grafana
GF_PATHS_LOGS                   /var/log/grafana
GF_PATHS_PLUGINS                /var/lib/grafana/plugins
GF_PATHS_PROVISIONING   /etc/grafana/provisioning
```

1.2. docker swarm

```
$ docker service create --replicas 1 --name prometheus \
    --mount
type=bind,source=`pwd`/prometheus.yml,destination=/etc/prometheus/promet
heus.yml \
    --publish published=9090,target=9090,protocol=tcp \
    prom/prometheus
```

1.3. docker-compose

```
```

1.4. 防火墙设置

```
firewall-cmd --zone=public --add-port=9090/tcp --permanent
firewall-cmd --zone=public --add-port=3000/tcp --permanent
firewall-cmd --zone=public --add-port=9191/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=9093/tcp --permanent
firewall-cmd --zone=public --add-port=9323/tcp --permanent
firewall-cmd --reload
```

查看端口策略是否已经生效

```
firewall-cmd --permanent --zone=public --list-ports
```

2. Prometheus 配置

2.1. Prometheus 命令行工具

刷新配置文件

```
#方式1:
kill -HUP ${prometheus_pid}

docker kill -s HUP <容器ID>

#方式2:
# 需要 --web.enable-lifecycle 参数为true
curl -X POST http://10.0.209.140:9090/-/reload
```

promtool 配置文件校验工具

安装 promtool

```
go get github.com/prometheus/prometheus/cmd/promtool
promtool check rules /path/to/example.rules.yml
```

```
promtool check config /etc/prometheus/prometheus.yml
```

2.2. rules 规则配置

prometheus.yml 配置文件

```
rule_files:
  - "rules/node.yml"    # 载入单个配置文件
  - "rules/*.rules"             # 通过通配符载入文件
```

prometheus 支持两种 rules

- recording rules
- alerting rules

recording rules

```
groups:
- name: cpu-node
  rules:
  - record: job_instance_mode:node_cpu_seconds:avg_rate5m
    expr: avg by (job, instance, mode) (rate(node_cpu_seconds_total[5m]))
```

alerting rules

```
groups:
- name: example
  rules:

  # Alert for any instance that is unreachable for >5 minutes.
  - alert: InstanceDown
    expr: up == 0
    for: 5m
    labels:
      severity: page
    annotations:
      summary: "Instance {{ $labels.instance }} down"
      description: "{{ $labels.instance }} of job {{ $labels.job }} has been down for
more than 5 minutes."

  # Alert for any instance that has a median request latency >1s.
  - alert: APIHighRequestLatency
    expr: api_http_request_latencies_second{quantile="0.5"} > 1
    for: 10m
    annotations:
      summary: "High request latency on {{ $labels.instance }}"
      description: "{{ $labels.instance }} has a median request latency above 1s
(current value: {{ $value }}s)"
```

2.3. SpringBoot

Maven pom.xml 文件中增加依赖

```
        <dependency>
            <groupId>io.micrometer</groupId>
            <artifactId>micrometer-registry-prometheus</artifactId>
        </dependency>
```

打包后运行 Springboot 项目，然后使用 /actuator/prometheus 地址测试是否有监控数据输出。https://api.netkiller.cn/actuator/prometheus

/etc/prometheus/prometheus.yml 增加如下配置：

```
 - job_name: 'springboot'
   scrape_interval: 5s
   metrics_path: '/actuator/prometheus'
   static_configs:
     - targets: ['127.0.0.1:8080']
```

Grafana 面板ID：4701

2.4. PromQL 自定义查询语言

Metrics 格式

Metric 的格式: metric 名称 {标签名=标签值} 监控样本

```
<metric name>{<label name>=<label value>, ...} <sample>
```

指标的名称(metric name)用于定义监控样本的含义，名称只能由ASCII字符、数字、下划线以及冒号组成并必须符合正则表达式[a-zA-Z_:][a-zA-Z0-9_:]*

标签(label)反映了当前样本的特征维度，通过这些维度Prometheus可以对样本数据进行过滤，聚合等。标签的名称只能由ASCII字符、数字以及下划线组成并满足正则表达式[a-zA-Z_][a-zA-Z0-9_]*

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9100/metrics | grep
node_cpu_seconds_total
# HELP node_cpu_seconds_total Seconds the cpus spent in each mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 16761.9
node_cpu_seconds_total{cpu="0",mode="iowait"} 2.91
node_cpu_seconds_total{cpu="0",mode="irq"} 0
node_cpu_seconds_total{cpu="0",mode="nice"} 0
node_cpu_seconds_total{cpu="0",mode="softirq"} 5.76
node_cpu_seconds_total{cpu="0",mode="steal"} 0
node_cpu_seconds_total{cpu="0",mode="system"} 440.28
node_cpu_seconds_total{cpu="0",mode="user"} 135.58
node_cpu_seconds_total{cpu="1",mode="idle"} 16851.16
node_cpu_seconds_total{cpu="1",mode="iowait"} 1.81
node_cpu_seconds_total{cpu="1",mode="irq"} 0
node_cpu_seconds_total{cpu="1",mode="nice"} 0
node_cpu_seconds_total{cpu="1",mode="softirq"} 1.33
node_cpu_seconds_total{cpu="1",mode="steal"} 0
node_cpu_seconds_total{cpu="1",mode="system"} 440.52
node_cpu_seconds_total{cpu="1",mode="user"} 125.7
node_cpu_seconds_total{cpu="2",mode="idle"} 16792.57
node_cpu_seconds_total{cpu="2",mode="iowait"} 2.52
node_cpu_seconds_total{cpu="2",mode="irq"} 0
node_cpu_seconds_total{cpu="2",mode="nice"} 0
node_cpu_seconds_total{cpu="2",mode="softirq"} 1.36
```

```
node_cpu_seconds_total{cpu="2",mode="steal"} 0
node_cpu_seconds_total{cpu="2",mode="system"} 445.29
node_cpu_seconds_total{cpu="2",mode="user"} 129.73
node_cpu_seconds_total{cpu="3",mode="idle"} 16844.57
node_cpu_seconds_total{cpu="3",mode="iowait"} 1.16
node_cpu_seconds_total{cpu="3",mode="irq"} 0
node_cpu_seconds_total{cpu="3",mode="nice"} 0
node_cpu_seconds_total{cpu="3",mode="softirq"} 1.24
node_cpu_seconds_total{cpu="3",mode="steal"} 0
node_cpu_seconds_total{cpu="3",mode="system"} 430.82
node_cpu_seconds_total{cpu="3",mode="user"} 135.15
```

metric 类型

Prometheus 定义了4种不同的指标类型(metric type)：

- Counter（计数器）
- Gauge（仪表盘）
- Histogram（直方图）
- Summary（摘要）

Counter：只增不减的计数器

Counter 例子

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9100/metrics | grep
node_cpu_seconds_total
# HELP node_cpu_seconds_total Seconds the cpus spent in each mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 16761.9
```

Gauge：可增可减的仪表盘

Gauge 类型的指标侧重于反应系统的当前状态，指标的样本数据可增可减。常用于内存容量的监控。

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9100/metrics | grep node_memory_MemFree
# HELP node_memory_MemFree_bytes Memory information field MemFree_bytes.
# TYPE node_memory_MemFree_bytes gauge
node_memory_MemFree_bytes 2.933243904e+09
```

Histogram

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9090/metrics | grep
prometheus_tsdb_compaction_chunk_range
```

```
# HELP prometheus_tsdb_compaction_chunk_range_seconds Final time range of chunks on
their first compaction
# TYPE prometheus_tsdb_compaction_chunk_range_seconds histogram
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="100"} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="400"} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="1600"} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="6400"} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="25600"} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="102400"} 3
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="409600"} 1506
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="1.6384e+06"} 1558
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="6.5536e+06"} 4564
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="2.62144e+07"} 4564
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="+Inf"} 4564
prometheus_tsdb_compaction_chunk_range_seconds_sum 5.85524936e+09
prometheus_tsdb_compaction_chunk_range_seconds_count 4564
```

Summary

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9090/metrics | grep
prometheus_tsdb_wal_fsync_duration_seconds
# HELP prometheus_tsdb_wal_fsync_duration_seconds Duration of WAL fsync.
# TYPE prometheus_tsdb_wal_fsync_duration_seconds summary
prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.5"} NaN
prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.9"} NaN
prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.99"} NaN
prometheus_tsdb_wal_fsync_duration_seconds_sum 1.63e-05
prometheus_tsdb_wal_fsync_duration_seconds_count 1
```

查询时间序列

标签查询

查询 instance="node-exporter:9100"

```
node_cpu_seconds_total{instance="node-exporter:9100"}
```

mode!="irq" 排出 irq

```
node_cpu_seconds_total{mode!="irq"}
```

查询所有 mode="user"

```
{mode="user"}
```

正则查询

```
node_cpu_seconds_total{mode=~"user|system|nice"}
restful_api_requests_total{environment=~"staging|testing|development",method!="GET"}

{instance =~"n.*"}
```

正则排除

```
node_cpu_seconds_total{mode!~"steal|softirq|irq|iowait|idle"}
```

范围查询

PromQL的时间范围选择器支持时间单位:

1. s - 秒
2. m - 分钟
3. h - 小时
4. d - 天
5. w - 周
6. y - 年

该表达式将会查询返回时间序列中最近5分钟的所有样本数据:

```
rate(node_memory_MemAvailable_bytes{}[5m])
```

可以使用offset时间位移操作:

```
node_memory_MemAvailable_bytes{} offset 5m
rate(node_load1{}[5m] offset 1m)
```

数学运算

PromQL 支持: 数学运算符, 逻辑运算符, 布尔运算符

PromQL操作符中优先级由高到低依次为:

- ^
- *, /, %
- +, -
- ==, !=, <=, <, >=, >
- and, unless
- or

Bytes 转 MB 的例子

```
node_memory_MemFree_bytes /  (1024 * 1024)
```

计算磁盘读写总量

```
(node_disk_read_bytes_total{device="vda"} + node_disk_written_bytes_total{device="vda"})
/ (1024 * 1024)
```

内存使用率计算

```
(node_memory_MemTotal_bytes - node_memory_MemFree_bytes) / node_memory_MemTotal_bytes *
100

# 查询出内存使用率到达 80% 的节点
(node_memory_MemTotal_bytes - node_memory_MemFree_bytes) / node_memory_MemTotal_bytes >
0.8

node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes * 100 > 80
```

聚合操作

PromQL内置的聚合操作和函数可以让用户对这些数据进行进一步的分析

rate()

通过rate()函数计算HTTP请求量的增长率：

```
rate(http_requests_total[5m])
```

topk() 和 bottomk()

查询当前访问量前10的HTTP地址：

```
topk(10, http_requests_total)
```

delta()

通过PromQL内置函数delta()可以获取样本在一段时间返回内的变化情况。例如，计算CPU温度在两个小时内的差异：

```
delta(cpu_temp_celsius{host="zeus"}[2h])
```

delta 适用于 Gauge 类型的监控指标

predict_linear()

使用predict_linear()对数据的变化趋势进行预测。例如，预测系统磁盘空间在4个小时之后的剩余情况：

```
predict_linear(node_filesystem_free{job="node"}[1h], 4 * 3600)
```

deriv()

deriv()计算样本的线性回归模型

```

```

sum()

求和操作

```
sum(node_cpu_seconds_total)
sum(node_cpu_seconds_total) by (mode)
```

```
Element                 Value
{mode="steal"}  0
{mode="system"} 2632.2400000000002
{mode="user"}   768.49
```

```
{mode="idle"}   93899.19
{mode="iowait"} 8.85
{mode="irq"}    0
{mode="nice"}   0
{mode="softirq"}        13.35
```

```
sum(node_cpu_seconds_total) without (instance)
```

```
sum(node_cpu_seconds_total) by (mode,cpu)
```

```
sum(sum(irate(node_cpu{mode!='idle'}[5m]))  / sum(irate(node_cpu[5m]))) by (instance)
```

avg()

计算平均数

```
avg(node_cpu_seconds_total) by (mode)
```

```
Element                 Value
{mode="nice"}   0
{mode="softirq"}        3.3374999999999995
{mode="steal"}  0
{mode="system"} 658.06
{mode="user"}   192.1225
{mode="idle"}   23474.7975
{mode="iowait"} 2.2125
{mode="irq"}    0
```

min (最小值)，max (最大值)

count_values()

quantile()

3. Prometheus Exporter

3.1. 监控 Docker

Collect Docker metrics with Prometheus

配置 docker /etc/docker/daemon.json

指定metrics采集端口， Prometheus 会定时从该端口拉取数据

```
{
  "metrics-addr" : "127.0.0.1:9323",
  "experimental" : true
}
```

查看 Docker 状态信息

```
iMac:prometheus neo$ curl http://localhost:9323/metrics
# HELP builder_builds_failed_total Number of failed image builds
# TYPE builder_builds_failed_total counter
builder_builds_failed_total{reason="build_canceled"} 0
builder_builds_failed_total{reason="build_target_not_reachable_error"} 0
builder_builds_failed_total{reason="command_not_supported_error"} 0
builder_builds_failed_total{reason="dockerfile_empty_error"} 0
builder_builds_failed_total{reason="dockerfile_syntax_error"} 0
builder_builds_failed_total{reason="error_processing_commands_error"} 0
builder_builds_failed_total{reason="missing_onbuild_arguments_error"} 0
builder_builds_failed_total{reason="unknown_instruction_error"} 0
# HELP builder_builds_triggered_total Number of triggered image builds
# TYPE builder_builds_triggered_total counter
builder_builds_triggered_total 0
# HELP engine_daemon_container_actions_seconds The number of seconds it
takes to process each container action
# TYPE engine_daemon_container_actions_seconds histogram
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.00
5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.01
"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.02
5"} 1
```

```
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.05
"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.1"
} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.25
"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.5"
} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="1"}
1
engine_daemon_container_actions_seconds_bucket{action="changes",le="2.5"
} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="5"}
1
engine_daemon_container_actions_seconds_bucket{action="changes",le="10"}
1
engine_daemon_container_actions_seconds_bucket{action="changes",le="+Inf
"} 1
engine_daemon_container_actions_seconds_sum{action="changes"} 0
engine_daemon_container_actions_seconds_count{action="changes"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.005
"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.01"
} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.025
"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.05"
} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.1"}
1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.25"
} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.5"}
1
engine_daemon_container_actions_seconds_bucket{action="commit",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="2.5"}
1
engine_daemon_container_actions_seconds_bucket{action="commit",le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="10"}
1
engine_daemon_container_actions_seconds_bucket{action="commit",le="+Inf"
} 1
engine_daemon_container_actions_seconds_sum{action="commit"} 0
engine_daemon_container_actions_seconds_count{action="commit"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.005
"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.01"
} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.025
"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.05"
} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.1"}
1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.25"
} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.5"}
1
engine_daemon_container_actions_seconds_bucket{action="create",le="1"} 2
engine_daemon_container_actions_seconds_bucket{action="create",le="2.5"}
2
engine_daemon_container_actions_seconds_bucket{action="create",le="5"} 2
engine_daemon_container_actions_seconds_bucket{action="create",le="10"}
2
engine_daemon_container_actions_seconds_bucket{action="create",le="+Inf"
} 2
engine_daemon_container_actions_seconds_sum{action="create"} 0.552623576
engine_daemon_container_actions_seconds_count{action="create"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",le="0.005
"} 1
engine_daemon_container_actions_seconds_bucket{action="delete",le="0.01"
} 1
engine_daemon_container_actions_seconds_bucket{action="delete",le="0.025
"} 1
engine_daemon_container_actions_seconds_bucket{action="delete",le="0.05"
} 1
engine_daemon_container_actions_seconds_bucket{action="delete",le="0.1"}
2
engine_daemon_container_actions_seconds_bucket{action="delete",le="0.25"
} 2
engine_daemon_container_actions_seconds_bucket{action="delete",le="0.5"}
2
engine_daemon_container_actions_seconds_bucket{action="delete",le="1"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",le="2.5"}
2
engine_daemon_container_actions_seconds_bucket{action="delete",le="5"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",le="10"}
2
engine_daemon_container_actions_seconds_bucket{action="delete",le="+Inf"
} 2
engine_daemon_container_actions_seconds_sum{action="delete"} 0.097789156
engine_daemon_container_actions_seconds_count{action="delete"} 2
engine_daemon_container_actions_seconds_bucket{action="start",le="0.005"
} 1
engine_daemon_container_actions_seconds_bucket{action="start",le="0.01"}
1
engine_daemon_container_actions_seconds_bucket{action="start",le="0.025"
} 1
engine_daemon_container_actions_seconds_bucket{action="start",le="0.05"}
1
engine_daemon_container_actions_seconds_bucket{action="start",le="0.1"}
1
```

```
engine_daemon_container_actions_seconds_bucket{action="start",le="0.25"}
1
engine_daemon_container_actions_seconds_bucket{action="start",le="0.5"}
1
engine_daemon_container_actions_seconds_bucket{action="start",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="start",le="2.5"}
3
engine_daemon_container_actions_seconds_bucket{action="start",le="5"} 3
engine_daemon_container_actions_seconds_bucket{action="start",le="10"} 3
engine_daemon_container_actions_seconds_bucket{action="start",le="+Inf"}
3
engine_daemon_container_actions_seconds_sum{action="start"} 2.804409176
engine_daemon_container_actions_seconds_count{action="start"} 3
# HELP engine_daemon_container_states_containers The count of containers
in various states
# TYPE engine_daemon_container_states_containers gauge
engine_daemon_container_states_containers{state="paused"} 0
engine_daemon_container_states_containers{state="running"} 2
engine_daemon_container_states_containers{state="stopped"} 2
# HELP engine_daemon_engine_cpus_cpus The number of cpus that the host
system of the engine has
# TYPE engine_daemon_engine_cpus_cpus gauge
engine_daemon_engine_cpus_cpus 2
# HELP engine_daemon_engine_info The information related to the engine
and the OS it is running on
# TYPE engine_daemon_engine_info gauge
engine_daemon_engine_info{architecture="x86_64",commit="ff3fbc9d55",daem
on_id="JXJ2:2434:PD5N:4UXM:POXB:ANLF:HHOE:G25W:Y3AG:UFUO:CBZP:H7K4",grap
hdriver="overlay2",kernel="4.19.76-linuxkit",os="Docker
Desktop",os_type="linux",version="19.03.13-beta2"} 1
# HELP engine_daemon_engine_memory_bytes The number of bytes of memory
that the host system of the engine has
# TYPE engine_daemon_engine_memory_bytes gauge
engine_daemon_engine_memory_bytes 2.088206336e+09
# HELP engine_daemon_events_subscribers_total The number of current
subscribers to events
# TYPE engine_daemon_events_subscribers_total gauge
engine_daemon_events_subscribers_total 7
# HELP engine_daemon_events_total The number of events logged
# TYPE engine_daemon_events_total counter
engine_daemon_events_total 11
# HELP engine_daemon_health_checks_failed_total The total number of
failed health checks
# TYPE engine_daemon_health_checks_failed_total counter
engine_daemon_health_checks_failed_total 0
# HELP engine_daemon_health_checks_total The total number of health
checks
# TYPE engine_daemon_health_checks_total counter
engine_daemon_health_checks_total 0
# HELP engine_daemon_network_actions_seconds The number of seconds it
takes to process each network action
```

```
# TYPE engine_daemon_network_actions_seconds histogram
engine_daemon_network_actions_seconds_bucket{action="allocate",le="0.005
"} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",le="0.01"
} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",le="0.025
"} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",le="0.05"
} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",le="0.1"}
0
engine_daemon_network_actions_seconds_bucket{action="allocate",le="0.25"
} 1
engine_daemon_network_actions_seconds_bucket{action="allocate",le="0.5"}
1
engine_daemon_network_actions_seconds_bucket{action="allocate",le="1"} 2
engine_daemon_network_actions_seconds_bucket{action="allocate",le="2.5"}
2
engine_daemon_network_actions_seconds_bucket{action="allocate",le="5"} 2
engine_daemon_network_actions_seconds_bucket{action="allocate",le="10"}
2
engine_daemon_network_actions_seconds_bucket{action="allocate",le="+Inf"
} 2
engine_daemon_network_actions_seconds_sum{action="allocate"} 0.721134186
engine_daemon_network_actions_seconds_count{action="allocate"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",le="0.005"
} 0
engine_daemon_network_actions_seconds_bucket{action="connect",le="0.01"}
0
engine_daemon_network_actions_seconds_bucket{action="connect",le="0.025"
} 0
engine_daemon_network_actions_seconds_bucket{action="connect",le="0.05"}
0
engine_daemon_network_actions_seconds_bucket{action="connect",le="0.1"}
0
engine_daemon_network_actions_seconds_bucket{action="connect",le="0.25"}
1
engine_daemon_network_actions_seconds_bucket{action="connect",le="0.5"}
1
engine_daemon_network_actions_seconds_bucket{action="connect",le="1"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",le="2.5"}
2
engine_daemon_network_actions_seconds_bucket{action="connect",le="5"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",le="10"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",le="+Inf"}
2
engine_daemon_network_actions_seconds_sum{action="connect"} 0.70473929
engine_daemon_network_actions_seconds_count{action="connect"} 2
# HELP etcd_debugging_snap_save_marshalling_duration_seconds The
marshalling cost distributions of save called by snapshot.
# TYPE etcd_debugging_snap_save_marshalling_duration_seconds histogram
```

```
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.001"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.002"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.004"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.008"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.016"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.032"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.064"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.128"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.256"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="0.512"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="1.024"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="2.048"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="4.096"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="8.192"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le="+Inf"}
0
etcd_debugging_snap_save_marshalling_duration_seconds_sum 0
etcd_debugging_snap_save_marshalling_duration_seconds_count 0
# HELP etcd_debugging_snap_save_total_duration_seconds The total latency
distributions of save called by snapshot.
# TYPE etcd_debugging_snap_save_total_duration_seconds histogram
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.001"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.002"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.004"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.008"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.016"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.032"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.064"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.128"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.256"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.512"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="1.024"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="2.048"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="4.096"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="8.192"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="+Inf"} 0
etcd_debugging_snap_save_total_duration_seconds_sum 0
```

```
etcd_debugging_snap_save_total_duration_seconds_count 0
# HELP etcd_disk_wal_fsync_duration_seconds The latency distributions of
fsync called by wal.
# TYPE etcd_disk_wal_fsync_duration_seconds histogram
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.001"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.002"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.004"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.008"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.016"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.032"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.064"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.128"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.256"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.512"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="1.024"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="2.048"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="4.096"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="8.192"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="+Inf"} 0
etcd_disk_wal_fsync_duration_seconds_sum 0
etcd_disk_wal_fsync_duration_seconds_count 0
# HELP etcd_snap_db_fsync_duration_seconds The latency distributions of
fsyncing .snap.db file
# TYPE etcd_snap_db_fsync_duration_seconds histogram
etcd_snap_db_fsync_duration_seconds_bucket{le="0.001"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.002"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.004"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.008"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.016"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.032"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.064"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.128"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.256"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.512"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="1.024"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="2.048"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="4.096"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="8.192"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="+Inf"} 0
etcd_snap_db_fsync_duration_seconds_sum 0
etcd_snap_db_fsync_duration_seconds_count 0
# HELP etcd_snap_db_save_total_duration_seconds The total latency
distributions of v3 snapshot save
# TYPE etcd_snap_db_save_total_duration_seconds histogram
etcd_snap_db_save_total_duration_seconds_bucket{le="0.1"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="0.2"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="0.4"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="0.8"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="1.6"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="3.2"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="6.4"} 0
```

```
etcd_snap_db_save_total_duration_seconds_bucket{le="12.8"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="25.6"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="51.2"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="+Inf"} 0
etcd_snap_db_save_total_duration_seconds_sum 0
etcd_snap_db_save_total_duration_seconds_count 0
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.1441e-05
go_gc_duration_seconds{quantile="0.25"} 1.7381e-05
go_gc_duration_seconds{quantile="0.5"} 4.7132e-05
go_gc_duration_seconds{quantile="0.75"} 8.847e-05
go_gc_duration_seconds{quantile="1"} 0.000336452
go_gc_duration_seconds_sum 0.000573966
go_gc_duration_seconds_count 7
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 124
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in
use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.3152408e+07
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated,
even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 3.7942088e+07
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the
profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.458259e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 239116
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage
collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 2.4064e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and
still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 1.3152408e+07
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be
used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 4.8480256e+07
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in
use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 1.67936e+07
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
```

```
go_memstats_heap_objects 134382
# HELP go_memstats_heap_released_bytes_total Total number of heap bytes
released to OS.
# TYPE go_memstats_heap_released_bytes_total counter
go_memstats_heap_released_bytes_total 4.6186496e+07
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from
system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 6.5273856e+07
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of
last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.6024955900357985e+09
# HELP go_memstats_lookups_total Total number of pointer lookups.
# TYPE go_memstats_lookups_total counter
go_memstats_lookups_total 0
# HELP go_memstats_mallocs_total Total number of mallocs.
# TYPE go_memstats_mallocs_total counter
go_memstats_mallocs_total 373498
# HELP go_memstats_mcache_inuse_bytes Number of bytes in use by mcache
structures.
# TYPE go_memstats_mcache_inuse_bytes gauge
go_memstats_mcache_inuse_bytes 3472
# HELP go_memstats_mcache_sys_bytes Number of bytes used for mcache
structures obtained from system.
# TYPE go_memstats_mcache_sys_bytes gauge
go_memstats_mcache_sys_bytes 16384
# HELP go_memstats_mspan_inuse_bytes Number of bytes in use by mspan
structures.
# TYPE go_memstats_mspan_inuse_bytes gauge
go_memstats_mspan_inuse_bytes 215424
# HELP go_memstats_mspan_sys_bytes Number of bytes used for mspan
structures obtained from system.
# TYPE go_memstats_mspan_sys_bytes gauge
go_memstats_mspan_sys_bytes 229376
# HELP go_memstats_next_gc_bytes Number of heap bytes when next garbage
collection will take place.
# TYPE go_memstats_next_gc_bytes gauge
go_memstats_next_gc_bytes 1.8665712e+07
# HELP go_memstats_other_sys_bytes Number of bytes used for other system
allocations.
# TYPE go_memstats_other_sys_bytes gauge
go_memstats_other_sys_bytes 542885
# HELP go_memstats_stack_inuse_bytes Number of bytes in use by the stack
allocator.
# TYPE go_memstats_stack_inuse_bytes gauge
go_memstats_stack_inuse_bytes 1.835008e+06
# HELP go_memstats_stack_sys_bytes Number of bytes obtained from system
for stack allocator.
# TYPE go_memstats_stack_sys_bytes gauge
go_memstats_stack_sys_bytes 1.835008e+06
```

```
# HELP go_memstats_sys_bytes Number of bytes obtained by system. Sum of
all system allocations.
# TYPE go_memstats_sys_bytes gauge
go_memstats_sys_bytes 7.1762168e+07
# HELP http_request_duration_microseconds The HTTP request latencies in
microseconds.
# TYPE http_request_duration_microseconds summary
http_request_duration_microseconds{handler="prometheus",quantile="0.5"}
5785.224
http_request_duration_microseconds{handler="prometheus",quantile="0.9"}
18160.443
http_request_duration_microseconds{handler="prometheus",quantile="0.99"}
18160.443
http_request_duration_microseconds_sum{handler="prometheus"} 27367.838
http_request_duration_microseconds_count{handler="prometheus"} 3
# HELP http_request_size_bytes The HTTP request sizes in bytes.
# TYPE http_request_size_bytes summary
http_request_size_bytes{handler="prometheus",quantile="0.5"} 232
http_request_size_bytes{handler="prometheus",quantile="0.9"} 232
http_request_size_bytes{handler="prometheus",quantile="0.99"} 232
http_request_size_bytes_sum{handler="prometheus"} 696
http_request_size_bytes_count{handler="prometheus"} 3
# HELP http_requests_total Total number of HTTP requests made.
# TYPE http_requests_total counter
http_requests_total{code="200",handler="prometheus",method="get"} 3
# HELP http_response_size_bytes The HTTP response sizes in bytes.
# TYPE http_response_size_bytes summary
http_response_size_bytes{handler="prometheus",quantile="0.5"} 4145
http_response_size_bytes{handler="prometheus",quantile="0.9"} 4171
http_response_size_bytes{handler="prometheus",quantile="0.99"} 4171
http_response_size_bytes_sum{handler="prometheus"} 12422
http_response_size_bytes_count{handler="prometheus"} 3
# HELP logger_log_entries_size_greater_than_buffer_total Number of log
entries which are larger than the log buffer
# TYPE logger_log_entries_size_greater_than_buffer_total counter
logger_log_entries_size_greater_than_buffer_total 0
# HELP logger_log_read_operations_failed_total Number of log reads from
container stdio that failed
# TYPE logger_log_read_operations_failed_total counter
logger_log_read_operations_failed_total 0
# HELP logger_log_write_operations_failed_total Number of log write
operations that failed
# TYPE logger_log_write_operations_failed_total counter
logger_log_write_operations_failed_total 0
# HELP process_cpu_seconds_total Total user and system CPU time spent in
seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 1.36
# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds 1.048576e+06
```

```
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 88
# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 6.0104704e+07
# HELP process_start_time_seconds Start time of the process since unix
epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.6024954353e+09
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 1.223262208e+09
# HELP swarm_dispatcher_scheduling_delay_seconds Scheduling delay is the
time a task takes to go from NEW to RUNNING state.
# TYPE swarm_dispatcher_scheduling_delay_seconds histogram
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.005"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.01"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.025"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.05"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.1"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.25"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.5"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="1"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="2.5"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="5"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="10"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="+Inf"} 0
swarm_dispatcher_scheduling_delay_seconds_sum 0
swarm_dispatcher_scheduling_delay_seconds_count 0
# HELP swarm_manager_configs_total The number of configs in the cluster
object store
# TYPE swarm_manager_configs_total gauge
swarm_manager_configs_total 0
# HELP swarm_manager_leader Indicates if this manager node is a leader
# TYPE swarm_manager_leader gauge
swarm_manager_leader 0
# HELP swarm_manager_networks_total The number of networks in the
cluster object store
# TYPE swarm_manager_networks_total gauge
swarm_manager_networks_total 0
# HELP swarm_manager_nodes The number of nodes
# TYPE swarm_manager_nodes gauge
swarm_manager_nodes{state="disconnected"} 0
swarm_manager_nodes{state="down"} 0
swarm_manager_nodes{state="ready"} 0
swarm_manager_nodes{state="unknown"} 0
# HELP swarm_manager_secrets_total The number of secrets in the cluster
object store
# TYPE swarm_manager_secrets_total gauge
swarm_manager_secrets_total 0
```

```
# HELP swarm_manager_services_total The number of services in the
cluster object store
# TYPE swarm_manager_services_total gauge
swarm_manager_services_total 0
# HELP swarm_manager_tasks_total The number of tasks in the cluster
object store
# TYPE swarm_manager_tasks_total gauge
swarm_manager_tasks_total{state="accepted"} 0
swarm_manager_tasks_total{state="assigned"} 0
swarm_manager_tasks_total{state="complete"} 0
swarm_manager_tasks_total{state="failed"} 0
swarm_manager_tasks_total{state="new"} 0
swarm_manager_tasks_total{state="orphaned"} 0
swarm_manager_tasks_total{state="pending"} 0
swarm_manager_tasks_total{state="preparing"} 0
swarm_manager_tasks_total{state="ready"} 0
swarm_manager_tasks_total{state="rejected"} 0
swarm_manager_tasks_total{state="remove"} 0
swarm_manager_tasks_total{state="running"} 0
swarm_manager_tasks_total{state="shutdown"} 0
swarm_manager_tasks_total{state="starting"} 0
# HELP swarm_node_manager Whether this node is a manager or not
# TYPE swarm_node_manager gauge
swarm_node_manager 0
# HELP swarm_raft_snapshot_latency_seconds Raft snapshot create latency.
# TYPE swarm_raft_snapshot_latency_seconds histogram
swarm_raft_snapshot_latency_seconds_bucket{le="0.005"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.01"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.025"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.05"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.1"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.25"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.5"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="1"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="2.5"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="5"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="10"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="+Inf"} 0
swarm_raft_snapshot_latency_seconds_sum 0
swarm_raft_snapshot_latency_seconds_count 0
# HELP swarm_raft_transaction_latency_seconds Raft transaction latency.
# TYPE swarm_raft_transaction_latency_seconds histogram
swarm_raft_transaction_latency_seconds_bucket{le="0.005"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.01"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.025"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.05"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.1"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.25"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.5"} 0
swarm_raft_transaction_latency_seconds_bucket{le="1"} 0
swarm_raft_transaction_latency_seconds_bucket{le="2.5"} 0
```

```
swarm_raft_transaction_latency_seconds_bucket{le="5"} 0
swarm_raft_transaction_latency_seconds_bucket{le="10"} 0
swarm_raft_transaction_latency_seconds_bucket{le="+Inf"} 0
swarm_raft_transaction_latency_seconds_sum 0
swarm_raft_transaction_latency_seconds_count 0
# HELP swarm_store_batch_latency_seconds Raft store batch latency.
# TYPE swarm_store_batch_latency_seconds histogram
swarm_store_batch_latency_seconds_bucket{le="0.005"} 0
swarm_store_batch_latency_seconds_bucket{le="0.01"} 0
swarm_store_batch_latency_seconds_bucket{le="0.025"} 0
swarm_store_batch_latency_seconds_bucket{le="0.05"} 0
swarm_store_batch_latency_seconds_bucket{le="0.1"} 0
swarm_store_batch_latency_seconds_bucket{le="0.25"} 0
swarm_store_batch_latency_seconds_bucket{le="0.5"} 0
swarm_store_batch_latency_seconds_bucket{le="1"} 0
swarm_store_batch_latency_seconds_bucket{le="2.5"} 0
swarm_store_batch_latency_seconds_bucket{le="5"} 0
swarm_store_batch_latency_seconds_bucket{le="10"} 0
swarm_store_batch_latency_seconds_bucket{le="+Inf"} 0
swarm_store_batch_latency_seconds_sum 0
swarm_store_batch_latency_seconds_count 0
# HELP swarm_store_lookup_latency_seconds Raft store read latency.
# TYPE swarm_store_lookup_latency_seconds histogram
swarm_store_lookup_latency_seconds_bucket{le="0.005"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.01"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.025"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.05"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.1"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.25"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.5"} 0
swarm_store_lookup_latency_seconds_bucket{le="1"} 0
swarm_store_lookup_latency_seconds_bucket{le="2.5"} 0
swarm_store_lookup_latency_seconds_bucket{le="5"} 0
swarm_store_lookup_latency_seconds_bucket{le="10"} 0
swarm_store_lookup_latency_seconds_bucket{le="+Inf"} 0
swarm_store_lookup_latency_seconds_sum 0
swarm_store_lookup_latency_seconds_count 0
# HELP swarm_store_memory_store_lock_duration_seconds Duration for which
the raft memory store lock was held.
# TYPE swarm_store_memory_store_lock_duration_seconds histogram
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.005"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.01"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.025"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.05"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.1"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.25"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.5"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="1"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="2.5"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="5"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="10"} 0
```

```
swarm_store_memory_store_lock_duration_seconds_bucket{le="+Inf"} 0
swarm_store_memory_store_lock_duration_seconds_sum 0
swarm_store_memory_store_lock_duration_seconds_count 0
# HELP swarm_store_read_tx_latency_seconds Raft store read tx latency.
# TYPE swarm_store_read_tx_latency_seconds histogram
swarm_store_read_tx_latency_seconds_bucket{le="0.005"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.01"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.025"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.05"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.1"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.25"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.5"} 0
swarm_store_read_tx_latency_seconds_bucket{le="1"} 0
swarm_store_read_tx_latency_seconds_bucket{le="2.5"} 0
swarm_store_read_tx_latency_seconds_bucket{le="5"} 0
swarm_store_read_tx_latency_seconds_bucket{le="10"} 0
swarm_store_read_tx_latency_seconds_bucket{le="+Inf"} 0
swarm_store_read_tx_latency_seconds_sum 0
swarm_store_read_tx_latency_seconds_count 0
# HELP swarm_store_write_tx_latency_seconds Raft store write tx latency.
# TYPE swarm_store_write_tx_latency_seconds histogram
swarm_store_write_tx_latency_seconds_bucket{le="0.005"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.01"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.025"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.05"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.1"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.25"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.5"} 0
swarm_store_write_tx_latency_seconds_bucket{le="1"} 0
swarm_store_write_tx_latency_seconds_bucket{le="2.5"} 0
swarm_store_write_tx_latency_seconds_bucket{le="5"} 0
swarm_store_write_tx_latency_seconds_bucket{le="10"} 0
swarm_store_write_tx_latency_seconds_bucket{le="+Inf"} 0
swarm_store_write_tx_latency_seconds_sum 0
swarm_store_write_tx_latency_seconds_count 0
```

配置 /etc/prometheus/prometheus.yml

```
# my global config
global:
  scrape_interval:     15s # Set the scrape interval to every 15
seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The
default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

  # Attach these labels to any time series or alerts when communicating
```

```
with
  # external systems (federation, remote storage, Alertmanager).
  external_labels:
      monitor: 'netkiller-monitor'

# Load rules once and periodically evaluate them according to the global
'evaluation_interval'.
rule_files:
  # - "first.rules"
  # - "second.rules"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries
scraped from this config.
  - job_name: 'prometheus'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ['host.docker.internal:9090'] # Only works on Docker
Desktop for Mac

  - job_name: 'docker'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ['docker.for.mac.host.internal:9323']

  - job_name: 'node-exporter'
    static_configs:
          - targets: ['node-exporter:9100']
```

```
$ docker service create --replicas 1 --name my-prometheus \
    --mount
type=bind,source=/tmp/prometheus.yml,destination=/etc/prometheus/prometh
eus.yml \
    --publish published=9090,target=9090,protocol=tcp \
    prom/prometheus
```

docker-compress

```
version: '3.9'
```

```
services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    volumes:
      - ./mac/prometheus.yml:/etc/prometheus/prometheus.yml
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - "--
web.console.libraries=/usr/share/prometheus/console_libraries"
      - "--web.console.templates=/usr/share/prometheus/consoles"
    ports:
      - '9090:9090'

  node-exporter:
    image: prom/node-exporter:latest
    container_name: node-exporter
    ports:
      - '9100:9100'
```

3.2. node-exporter

https://grafana.com/grafana/dashboards/8919

```
version: '3.9'
services:
  node-exporter:
    image: prom/node-exporter:latest
    container_name: node-exporter
    hostname: node-exporter
    restart: always
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
    ports:
      - '9100:9100'
    command:
      - '--path.procfs=/host/proc'
      - '--path.sysfs=/host/sys'
      - --collector.filesystem.ignored-mount-points
      -
"^/(sys|proc|dev|host|etc|rootfs/var/lib/docker/containers|rootfs/var/li
b/docker/overlay2|rootfs/run/docker/netns|rootfs/var/lib/docker/aufs)
($$|/)"
```

3.3. cadvisor

```
docker run                                    \
--volume=/:/rootfs:ro                         \
--volume=/var/run:/var/run:rw                 \
--volume=/sys:/sys:ro                         \
--volume=/var/lib/docker/:/var/lib/docker:ro  \
--publish=8080:8090                           \
--detach=true                                 \
--name=cadvisor                               \
google/cadvisor:latest
```

修改 prometheus.yml 添加 cadvisor 监控

```
- job_name: cadvisor1
    static_configs:
      - targets: ['cadvisor:8090']
```

3.4. Nginx Prometheus Exporter

Nginx 配置，开启状态

/etc/nginx/conf.d/status.conf:

```
server {
        listen 80;
        server_name 127.0.0.1;
        location = /status {
                stub_status;
                access_log off;
            allow 127.0.0.1;
                deny all;
        }
}
```

如果 nginx 是 docker 运行需要设置 server_name，实体机不需要指定 server_name。

docker-compose.yml 编排脚本

```
version: '3.9'
services:
  nginx-prometheus-exporter:
    image: nginx/nginx-prometheus-exporter:latest
    command: -nginx.scrape-uri http://your_ipaddress_or_domain/status
    ports:
      - "9113:9113"
```

nginx-prometheus-exporter 官方下载地址：https://github.com/nginxinc/nginx-prometheus-exporter

调试方法

```
$ nginx-prometheus-exporter -nginx.scrape-uri http://<nginx>/status

neo@MacBook-Pro-Neo ~/workspace/Linux % curl
http://localhost:9113/metrics
# HELP nginx_connections_accepted Accepted client connections
# TYPE nginx_connections_accepted counter
nginx_connections_accepted 53
# HELP nginx_connections_active Active client connections
# TYPE nginx_connections_active gauge
nginx_connections_active 10
# HELP nginx_connections_handled Handled client connections
# TYPE nginx_connections_handled counter
nginx_connections_handled 53
# HELP nginx_connections_reading Connections where NGINX is reading the
request header
# TYPE nginx_connections_reading gauge
nginx_connections_reading 0
# HELP nginx_connections_waiting Idle client connections
# TYPE nginx_connections_waiting gauge
nginx_connections_waiting 9
# HELP nginx_connections_writing Connections where NGINX is writing the
response back to the client
```

```
# TYPE nginx_connections_writing gauge
nginx_connections_writing 1
# HELP nginx_http_requests_total Total http requests
# TYPE nginx_http_requests_total counter
nginx_http_requests_total 390
# HELP nginx_up Status of the last metric scrape
# TYPE nginx_up gauge
nginx_up 1
# HELP nginxexporter_build_info Exporter build information
# TYPE nginxexporter_build_info gauge
nginxexporter_build_info{commit="5f88afbd906baae02edfbab4f5715e06d88538a
0",date="2021-03-22T20:16:09Z",version="0.9.0"} 1
```

配置 prometheus.yml 加入 job

```
  - job_name: 'nginx_exporter'
    static_configs:
     - targets: ['nginx-exporter:9113']
```

NGINX exporter dashboard: https://grafana.com/grafana/dashboards/12708

Official dashboard for NGINX Prometheus exporter for https://github.com/nginxinc/nginx-prometheus-exporter

3.5. Redis

https://github.com/oliver006/redis_exporter

```
version: '3.9'
services:
  redis-exporter:
        image: oliver006/redis_exporter
        container_name: redis-exporter
        hostname: redis-exporter
        restart: always
        ports:
            - "9121:9121"
        command:
            - '--redis.addr=redis://:passw0rd@redis.netkiller.cn:6379'
```

使用下面命令确认 redis-exporter 是否工作正常

```
root@production:~/prometheus# curl -s
http://redis.netkiller.cn:9121/metrics | head
# HELP go_gc_duration_seconds A summary of the pause duration of garbage
collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
```

修改配置文件 prometheus.yml 加入下面配置

```
scrape_configs:
  - job_name: redis_exporter
    static_configs:
    - targets: ['<<REDIS-EXPORTER-HOSTNAME>>:9121']
```

Grafana 面板：https://grafana.com/grafana/dashboards/763

3.6. MongoDB

https://github.com/percona/mongodb_exporter

docker-compose.yml 构建脚本

```
version: '3.9'
services:
  mongodb_exporter:
    image: noenv/mongo-exporter:latest
    container_name: mongodb_exporter
    hostname: mongodb_exporter
    restart: always
    ports:
        - "9216:9216"
    command:
        - '--
mongodb.uri=mongodb://admin:admin@mongo.netkiller.cn:27017/admin'
```

检查 exporter 数据采集状态

```
root@production:~/prometheus# curl -s http://localhost:9216/metrics |
head
# HELP go_gc_duration_seconds A summary of the pause duration of garbage
collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.4908e-05
go_gc_duration_seconds{quantile="0.25"} 2.7779e-05
go_gc_duration_seconds{quantile="0.5"} 2.9463e-05
go_gc_duration_seconds{quantile="0.75"} 3.736e-05
go_gc_duration_seconds{quantile="1"} 0.000120332
go_gc_duration_seconds_sum 0.001014832
go_gc_duration_seconds_count 26
# HELP go_goroutines Number of goroutines that currently exist.
```

修改配置文件 prometheus.yml 加入下面配置

```
  - job_name: mongo_exporter
    static_configs:
    - targets: ['mongo.netkiller.cn:9216']
```

Dashboard for Grafana (ID: 2583)

3.7. MySQL

https://github.com/prometheus/mysqld_exporter

创建 MySQL 监控用户

```
mysql> CREATE USER 'exporter'@'%' IDENTIFIED BY 'exporterpassword' WITH
MAX_USER_CONNECTIONS 3;
mysql> GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO
'exporter'@'%';
```

```
version: '3.9'
services:
  mysqld_exporter:
    image: prom/mysqld-exporter:latest
    container_name: mysqld_exporter
    hostname: mysqld_exporter
    restart: always
    ports:
        - "9104:9104"
    environment:
      - DATA_SOURCE_NAME=exporter:passw0rd@(db.netkiller.cn:3306)/neo
    # command:
    #   --collect.info_schema.processlist
    #   --collect.info_schema.innodb_metrics
    #   --collect.info_schema.tablestats
    #   --collect.info_schema.tables
    #   --collect.info_schema.userstats
    #   --collect.engine_innodb_status
```

检查 exporter 数据采集状态

```
root@production:~# curl -s http://db.netkiller.cn:9104/metrics | head
# HELP go_gc_duration_seconds A summary of the pause duration of garbage
collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.9298e-05
go_gc_duration_seconds{quantile="0.25"} 2.846e-05
go_gc_duration_seconds{quantile="0.5"} 3.8975e-05
go_gc_duration_seconds{quantile="0.75"} 6.0157e-05
go_gc_duration_seconds{quantile="1"} 0.000150234
go_gc_duration_seconds_sum 0.007067359
go_gc_duration_seconds_count 145
# HELP go_goroutines Number of goroutines that currently exist.
```

修改配置文件 prometheus.yml 加入下面配置

```
  - job_name: mysql_exporter
    static_configs:
    - targets: ['db.netkiller.cn:9104']
```

https://grafana.com/oss/prometheus/exporters/mysql-exporter/

14057

3.8. Blackbox Exporter(blackbox-exporter)

默认配置文件

```
version: '3.9'
services:
  blackbox_exporter:
    image: prom/blackbox-exporter:latest
    container_name: blackbox_exporter
    hostname: blackbox-exporter
    restart: always
    ports:
      - "9115:9115"
    # environment:
    volumes:
      - ${PWD}/blackbox-exporter/config.yml:/etc/blackbox_exporter/config.yml
```

/etc/blackbox_exporter/config.yml

```
modules:
  http_2xx:
    prober: http
    timeout: 10s
    http:
      method: GET
  http_post_2xx:
    prober: http
    http:
      method: POST
  tcp_connect:
    prober: tcp
    timeout: 10s
  pop3s_banner:
    prober: tcp
    timeout: 10s
    tcp:
      query_response:
      - expect: "^+OK"
      tls: true
      tls_config:
        insecure_skip_verify: false
  ssh_banner:
    prober: tcp
    tcp:
      query_response:
      - expect: "^SSH-2.0-"
      - send: "SSH-2.0-blackbox-ssh-check"
  irc_banner:
    prober: tcp
    tcp:
      query_response:
      - send: "NICK prober"
      - send: "USER prober prober prober :prober"
      - expect: "PING :([^ ]+)"
        send: "PONG ${1}"
      - expect: "^:[^ ]+ 001"
  icmp:
    prober: icmp
    timeout: 2s
```

配置 Prometheus 在配置文件 prometheus.yml 中增加如下内容

```
scrape_configs:
  - job_name: blackbox_exporter
    static_configs:
    - targets: ['blackbox-exporter:9115']

  - job_name: blackbox-http
    metrics_path: /probe
    params:
      module: [http_2xx]
    static_configs:
      - targets:
        - http://192.168.30.10
        - http://192.168.30.11
        - http://192.168.3.15
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement:  blackbox-exporter:9115

  - job_name: 'blackbox-ping'
    metrics_path: /probe
    params:
      modelus: [icmp]
    static_configs:
      - targets:
        - 8.8.8.8
        labels:
          instance: Google DNS
      - targets:
        - 247.192.129.167
        labels:
          instance: test
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: blackbox-exporter:9115

  - job_name: 'blackbox_tcp_connect'
    scrape_interval: 30s
    metrics_path: /probe
    params:
      module: [tcp_connect]
    static_configs:
      - targets:
        - 127.0.0.1:3306
        - 127.0.0.1:6379
        - 127.0.0.1:27017
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: blackbox-exporter:9115
```

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % mkdir blackbox-
exporter
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % docker-compose cp
blackbox_exporter:/etc/blackbox_exporter/config.yml blackbox-exporter
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % cat blackbox-
exporter/config.yml
modules:
  http_2xx:
    prober: http
  http_post_2xx:
    prober: http
    http:
      method: POST
  tcp_connect:
    prober: tcp
  pop3s_banner:
    prober: tcp
    tcp:
      query_response:
      - expect: "^+OK"
      tls: true
      tls_config:
        insecure_skip_verify: false
  ssh_banner:
    prober: tcp
    tcp:
      query_response:
      - expect: "^SSH-2.0-"
      - send: "SSH-2.0-blackbox-ssh-check"
  irc_banner:
    prober: tcp
    tcp:
      query_response:
      - send: "NICK prober"
      - send: "USER prober prober prober :prober"
      - expect: "PING :([^ ]+)"
        send: "PONG ${1}"
      - expect: "^:[^ ]+ 001"
  icmp:
    prober: icmp
```

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9115/metrics | head
# HELP blackbox_exporter_build_info A metric with a constant '1' value
labeled by version, revision, branch, and goversion from which
blackbox_exporter was built.
# TYPE blackbox_exporter_build_info gauge
blackbox_exporter_build_info{branch="HEAD",goversion="go1.16.4",revision
="5d575b88eb12c65720862e8ad2c5890ba33d1ed0",version="0.19.0"} 1
# HELP blackbox_exporter_config_last_reload_success_timestamp_seconds
Timestamp of the last successful configuration reload.
# TYPE blackbox_exporter_config_last_reload_success_timestamp_seconds
gauge
blackbox_exporter_config_last_reload_success_timestamp_seconds
1.6298732380407274e+09
# HELP blackbox_exporter_config_last_reload_successful Blackbox exporter
config loaded successfully.
# TYPE blackbox_exporter_config_last_reload_successful gauge
blackbox_exporter_config_last_reload_successful 1
# HELP blackbox_module_unknown_total Count of unknown modules requested
by probes
```

Prometheus Blackbox Exporter: 12275

手工发起请求

Ping

```
curl -s http://127.0.0.1:9115/probe?target=127.0.0.1&module=icmp
```

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?target\=127.0.0.1\&module\=icmp | grep
^\probe_success
probe_success 1
```

默认超时时间太长，使用一个错误IP地址13.13.13.13测试，会等待很长时间

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?target\=13.13.13.13\&module\=icmp | grep
^\probe_success
probe_success 0
```

优化方法是设置 timeout，编辑 /etc/blackbox_exporter/config.yml 配置设置为2秒，这样2秒立即反馈IP地址PING结果。

```
  icmp:
    prober: icmp
    timeout: 2s
```

TCP 检查端口号

```
curl -s http://127.0.0.1:9115/probe?
target=127.0.0.1:8080&module=tcp_connect&debug=true
```

HTTP/HTTPS URL

```
curl -s http://127.0.0.1:9115/probe?
target=http://www.netkiller.cn&module=http_2xxx
```

HTTP 不能仅仅看 probe_success 状态，还要看 probe_http_status_code，这是HTTP服务器返回的状态码，通常是 200

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?
target\=http://192.168.30.11\&module\=http_2xx | grep -v ^#
probe_dns_lookup_time_seconds 0.000241511
probe_duration_seconds 0.011169367
probe_failed_due_to_regex 0
probe_http_content_length -1
probe_http_duration_seconds{phase="connect"} 0.003367677
probe_http_duration_seconds{phase="processing"} 0.006039874
probe_http_duration_seconds{phase="resolve"} 0.000241511
probe_http_duration_seconds{phase="tls"} 0
probe_http_duration_seconds{phase="transfer"} 0.000451174
probe_http_redirects 0
probe_http_ssl 0
probe_http_status_code 200
```

```
probe_http_uncompressed_body_length 407
probe_http_version 1.1
probe_ip_addr_hash 2.66977244e+08
probe_ip_protocol 4
probe_success 1
```

HTTPS

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?
target\=https://www.netkiller.cn/api/captcha\&module\=http_2xx | grep -v
^#
probe_dns_lookup_time_seconds 0.023551527
probe_duration_seconds 0.054094864
probe_failed_due_to_regex 0
probe_http_content_length -1
probe_http_duration_seconds{phase="connect"} 0.005037651
probe_http_duration_seconds{phase="processing"} 0.009932338
probe_http_duration_seconds{phase="resolve"} 0.023551527
probe_http_duration_seconds{phase="tls"} 0.011010897
probe_http_duration_seconds{phase="transfer"} 0.0009768
probe_http_redirects 0
probe_http_ssl 1
probe_http_status_code 200
probe_http_uncompressed_body_length 2604
probe_http_version 2
probe_ip_addr_hash 7.14414465e+08
probe_ip_protocol 4
probe_ssl_earliest_cert_expiry 1.661299199e+09
probe_ssl_last_chain_expiry_timestamp_seconds 1.661299199e+09
probe_ssl_last_chain_info{fingerprint_sha256="fd49505ad2ab79ef02070a2017
2ae56acbe525195ae0ddbe18359ce4144fea6b"} 1
probe_success 1
probe_tls_version_info{version="TLS 1.2"} 1
```

⚠注意这几项，probe_http_ssl 1，probe_http_version 2，probe_tls_version_info{version="TLS 1.2"} 1

```
probe_dns_lookup_time_seconds #DNS解析时间,单位s
probe_duration_seconds #探测从开始到结束的时间,单位 s,请求这个页面响应时间
probe_failed_due_to_regex 0
probe_http_content_length #HTTP 内容响应的长度
```

```
#按照阶段统计每阶段的时间
probe_http_duration_seconds{phase="connect"} 0.050388884   #连接时间
probe_http_duration_seconds{phase="processing"} 0.45868667 #处理请求的时间
probe_http_duration_seconds{phase="resolve"} 0.040037612  #响应时间
probe_http_duration_seconds{phase="tls"} 0.145433254    #校验证书的时间
probe_http_duration_seconds{phase="transfer"} 0.000566269
probe_http_redirects 1 #是否重定向的
probe_http_ssl 1 SSL证书可用
probe_http_status_code 200      #返回的状态码
probe_http_uncompressed_body_length #未压缩的响应主体长度
probe_http_version 2 #http 协议的版本
probe_ip_protocol 4  #IP协议的版本号，4是ipv4，6是 ipv6
probe_ssl_earliest_cert_expiry SSL证书过期时间
probe_success 1 #是否探测成功，1表示成功，0表示失败
probe_tls_version_info{version="TLS 1.2"} 1  #TLS 的版本号
```

自定义

restful

```
http_post_2xx:
    prober: http
    timeout: 5s
    http:
      method: POST
      headers:
        Content-Type: application/json
      body: '{}'
```

http auth

```
http_basic_auth_example:
    prober: http
    timeout: 5s
    http:
      method: POST
      headers:
        Host: "login.example.com"
      basic_auth:
        username: "username"
        password: "mysecret"
```

```
http_2xx_example:
    prober: http
    timeout: 5s
    http:
      valid_http_versions: ["HTTP/1.1", "HTTP/2"]
      valid_status_codes: [200,301,302]
```

SSL证书检查

```
  http_2xx_example:
    prober: http
    timeout: 5s
    http:
      valid_status_codes: []
      method: GET
      no_follow_redirects: false
      fail_if_ssl: false
      fail_if_not_ssl: false
```

检测返回内容

```
  http_2xx_example:
    prober: http
    timeout: 5s
    http:
      method: GET
      fail_if_matches_regexp:
        - "Could not connect to database"
      fail_if_not_matches_regexp:
        - "Download the latest version here"
```

3.9. SNMP Exporter

```
% docker-compose cp snmp_exporter:/etc/snmp_exporter/snmp.yml snmp-
exporter
% vim snmp-exporter/snmp.yml
  auth:
    community: public
```

确认交换机或路由器的SNMP已经开启，如何开启交换机和路由器的SNMP请参考 《Netkiller Network 手札》

```
neo@MacBook-Pro-Neo ~/workspace % snmpwalk -v2c -c public 172.16.254.254
| more
SNMPv2-MIB::sysDescr.0 = STRING: H3C Series Router MSR26-00
H3C Comware Platform Software
Comware Software Version 5.20, Release 2516P15
Copyright(c) 2004-..}> New H3C Technologies Co., Ltd.

SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.25506.1.913
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (794793008) 91 days,
23:45:30.08
SNMPv2-MIB::sysContact.0 = STRING: R&D Hangzhou, New H3C Technologies
Co., Ltd.
SNMPv2-MIB::sysName.0 = STRING: MSR2610
SNMPv2-MIB::sysLocation.0 = STRING: Hangzhou, China
SNMPv2-MIB::sysServices.0 = INTEGER: 78
IF-MIB::ifNumber.0 = INTEGER: 24
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifIndex.6 = INTEGER: 6
IF-MIB::ifIndex.7 = INTEGER: 7
IF-MIB::ifIndex.8 = INTEGER: 8
IF-MIB::ifIndex.9 = INTEGER: 9
IF-MIB::ifIndex.10 = INTEGER: 10
```

测试网站 http://localhost:9116

或者使用 curl 命令，确保你监控的社会能读取到 SNMP 数据。

```
neo@MacBook-Pro-Neo ~/workspace % curl -s http://localhost:9116/snmp\?
target\=172.16.254.254 | more
# HELP ifAdminStatus The desired state of the interface -
1.3.6.1.2.1.2.2.1.7
# TYPE ifAdminStatus gauge
ifAdminStatus{ifAlias="Aux0
Interface",ifDescr="Aux0",ifIndex="1",ifName="Aux0"} 1
ifAdminStatus{ifAlias="Cellular0/0
Interface",ifDescr="Cellular0/0",ifIndex="2",ifName="Cellular0/0"} 1
ifAdminStatus{ifAlias="Dialer1
Interface",ifDescr="Dialer1",ifIndex="14",ifName="Dialer1"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/0
Interface",ifDescr="GigabitEthernet0/0",ifIndex="3",ifName="GigabitEther
net0/0"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/1
Interface",ifDescr="GigabitEthernet0/1",ifIndex="4",ifName="GigabitEther
net0/1"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/2
Interface",ifDescr="GigabitEthernet0/2",ifIndex="5",ifName="GigabitEther
net0/2"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/3
Interface",ifDescr="GigabitEthernet0/3",ifIndex="6",ifName="GigabitEther
net0/3"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/4
Interface",ifDescr="GigabitEthernet0/4",ifIndex="7",ifName="GigabitEther
net0/4"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/5
Interface",ifDescr="GigabitEthernet0/5",ifIndex="8",ifName="GigabitEther
net0/5"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/6
Interface",ifDescr="GigabitEthernet0/6",ifIndex="9",ifName="GigabitEther
net0/6"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/7
Interface",ifDescr="GigabitEthernet0/7",ifIndex="10",ifName="GigabitEthe
rnet0/7"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/8
Interface",ifDescr="GigabitEthernet0/8",ifIndex="11",ifName="GigabitEthe
rnet0/8"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/9
Interface",ifDescr="GigabitEthernet0/9",ifIndex="12",ifName="GigabitEthe
rnet0/9"} 1
ifAdminStatus{ifAlias="NULL0
Interface",ifDescr="NULL0",ifIndex="13",ifName="NULL0"} 1
```

snmp 的监控 Dashboard ID 为：10523

4. Alertmanager

4.1. Docker 安装

```
 alertmanager:
  image: prom/alertmanager:latest
  container_name: alertmanager
  hostname: alertmanager
  restart: always
  volumes:
   - ${PWD}/alertmanager/config.yml:/etc/alertmanager/config.yml
   - alertmanager:/alertmanager
  ports:
   - "9093:9093"
  depends_on:
   - prometheus
  command:
   --config.file=/etc/alertmanager/config.yml
   --cluster.advertise-address=0.0.0.0:9093
```

配置 prometheus.yml

```
alerting:
 alertmanagers:
  - static_configs:
   - targets: ["alertmanager:9093"]

scrape_configs:
 - job_name: 'alertmanager'
  metrics_path: "/metrics"
```

检查 Alertmanager 是否正常工作

```
root@production:~# curl -s http://localhost:9093/metrics | head
# HELP alertmanager_alerts How many alerts by state.
# TYPE alertmanager_alerts gauge
alertmanager_alerts{state="active"} 0
alertmanager_alerts{state="suppressed"} 0
# HELP alertmanager_alerts_invalid_total The total number of received alerts
that were invalid.
# TYPE alertmanager_alerts_invalid_total counter
alertmanager_alerts_invalid_total{version="v1"} 0
alertmanager_alerts_invalid_total{version="v2"} 0
```

```
# HELP alertmanager_alerts_received_total The total number of received alerts.
# TYPE alertmanager_alerts_received_total counter
```

解决时区问题，默认 docker 镜像使用 UTC，我们需要改为GMT+8

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % docker exec -it alertmanager
sh
/alertmanager $ cat /etc/localtime
TZif2UTCTZif2?UTC
UTC0
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % docker-compose cp
alertmanager:/usr/share/zoneinfo/PRC Shanghai
```

查看反馈信息

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -X OPTIONS
127.0.0.1:9093/api/v1/alerts -v
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 9093 (#0)
> OPTIONS /api/v1/alerts HTTP/1.1
> Host: 127.0.0.1:9093
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Access-Control-Allow-Headers: Accept, Authorization, Content-Type, Origin
< Access-Control-Allow-Methods: GET, POST, DELETE, OPTIONS
< Access-Control-Allow-Origin: *
< Access-Control-Expose-Headers: Date
< Cache-Control: no-cache, no-store, must-revalidate
< Date: Mon, 23 Aug 2021 12:18:20 GMT
< Content-Length: 0
<
* Connection #0 to host 127.0.0.1 left intact
* Closing connection 0
```

4.2. alertmanager.yml 配置文件

amtool 配置文件检查工具

```
amtool check-config alertmanager.yml
```

global 全局配置项

SMTP 配置

```
global:
  resolve_timeout: 5m                                           #处理超时时间，默认
为5min
  smtp_smarthost: 'smtp.nejtkiller.cn:25'       # 邮箱smtp服务器代理
  smtp_from: 'monitor@netkiller.cn'             # 发送邮箱名称
  smtp_auth_username: 'monitor@netkiller.cn'    # 邮箱名称
  smtp_auth_password: '******'                          #邮箱密码
```

route 路由配置

```
route:
  group_by: ['alertname']       # 报警分组名称
  group_wait: 10s                       # 最初即第一次等待多久时间发送一组警报的通知
  group_interval: 10s           # 在发送新警报前的等待时间
  repeat_interval: 1m           # 发送重复警报的周期
  receiver: 'email'             # 发送警报的接收者的名称，以下receivers name的名称
```

receivers 定义警报接收者

```
receivers:
  - name: 'email'                                       # 警报
    email_configs:                                      # 邮箱配置
    - to: 'monitor@netkiller.cn'        # 接收警报的email配置
```

Webhook 配置

通过 webhook 触发手机短信发送程序

```
global:
```

```
route:
  group_by: ["alertname"]
  group_wait: 10s
  group_interval: 10s
  repeat_interval: 1h
  receiver: webhook

receivers:
- name: 'webhook'
  webhook_configs:
    - url: 'http://alertmanager-webhook:8080/webhook'
```

```
docker-compose.yaml 容器编排文件

version: '3.9'
services:
  alertmanager-webhook:
    image: netkiller/alertmanager
    container_name: alertmanager-webhook
    restart: always
    hostname: alertmanager-webhook
    extra_hosts:
      - dysmsapi.aliyuncs.com:106.11.45.35
    environment:
      TZ: Asia/Shanghai
      JAVA_OPTS: -Xms256m -Xmx1024m -XX:MetaspaceSize=128m -
XX:MaxMetaspaceSize=512m
    ports:
      - 8080:8080
    volumes:
      - ${PWD}/alertmanager/application.properties:/app/application.properties
      - /tmp/alertmanager:/tmp
    working_dir: /app
    command:
      --spring.config.location=/app/application.properties
```

application.properties 配置文件

4.3. 触发测试

```
alerts_message='[
  {
    "labels": {
       "alertname": "磁盘满",
       "dev": "sda1",
       "instance": "example",
       "msgtype": "testing"
     },
     "annotations": {
        "info": "/dev/vdb1 磁盘空间满",
        "summary": "/dev/vdb1 磁盘空间满"
      }
  }
]'
curl -XPOST -d"$alerts_message" http://127.0.0.1:9093/api/v1/alerts
```

```
#!/usr/bin/env bash

alerts_message='[
  {
    "labels": {
       "alertname": "DiskRunningFull",
       "dev": "sda1",
       "instance": "example1",
       "msgtype": "testing"
     },
     "annotations": {
        "info": "The disk sda1 is running full",
        "summary": "please check the instance example1"
      }
  },
  {
    "labels": {
       "alertname": "DiskRunningFull",
       "dev": "sda2",
       "instance": "example1",
       "msgtype": "testing"
     },
     "annotations": {
        "info": "The disk sda2 is running full",
        "summary": "please check the instance example1",
        "runbook": "the following link http://test-url should be clickable"
      }
  }
]'

curl -XPOST -d"$alerts_message" http://127.0.0.1:9093/api/v1/alerts
```

4.4. 警报状态

- firing：警报已被激活，而且超出设置的持续时间。
- pending：警报被激活，但是低于配置的持续即rule里的FOR字段设置的时间。
- inactive：既不是pending也不是firing的时候状态变为inactive
- resolved: 故障恢复

5. Grafana

Installing and Configuring Graphite

5.1. cadvisor

https://grafana.com/grafana/dashboards/11277

5.2. Docker - container summary (Prometheus)

https://grafana.com/grafana/dashboards/11467

This is a visualization of the Docker container metrics provided by the prometheus-net/docker_exporter project.

第 2 章 Zabbix

1. Installing and Configuring Zabbix

1.1. Ubuntu

```
neo@monitor:~$ apt-cache search zabbix
zabbix-agent - network monitoring solution - agent
zabbix-frontend-php - network monitoring solution - PHP front-
end
zabbix-proxy-mysql - network monitoring solution - proxy (using
MySQL)
zabbix-proxy-pgsql - network monitoring solution - proxy (using
PostgreSQL)
zabbix-server-mysql - network monitoring solution - server
(using MySQL)
zabbix-server-pgsql - network monitoring solution - server
(using PostgreSQL)
```

```
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost'
IDENTIFIED BY 'chen' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

```
sudo apt-get install zabbix-server-mysql zabbix-frontend-php
```

如果上述过程中遇到一些问题，可以手工安装数据库

```
$ sudo mysql -uroot -p -e"create database zabbix;"
$ sudo mysql -uroot -p -e"grant all privileges on zabbix.* to
zabbix@localhost identified by 'enter-password-here';"
$ mysql -uzabbix -p zabbix < /usr/share/zabbix-server/mysql.sql
$ mysql -uzabbix -p zabbix < /usr/share/zabbix-server/data.sql
$ sudo dpkg-reconfigure zabbix-server-mysql
```

```
cat >> /etc/services <<EOF

zabbix-agent    10050/tcp                       #Zabbix Agent
zabbix-agent    10050/udp                       #Zabbix Agent
zabbix-trapper  10051/tcp                       #Zabbix Trapper
zabbix-trapper  10051/udp                       #Zabbix Trapper
EOF
```

1.2. CentOS Zabbix 2.4

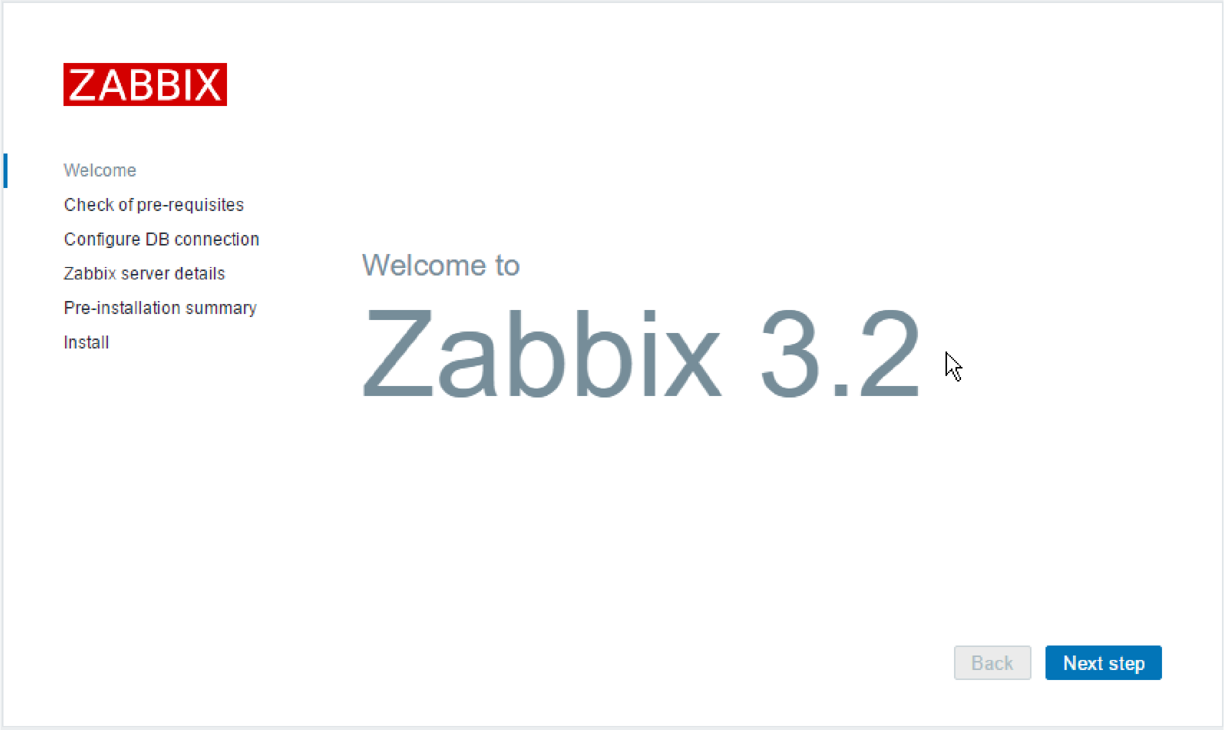
```
yum localinstall -y
http://repo.zabbix.com/zabbix/2.4/rhel/7/x86_64/zabbix-release-
2.4-1.el7.noarch.rpm

yum install -y zabbix-server-mysql zabbix-web-mysql

cd /usr/share/doc/zabbix-server-mysql-2.4.0/create/

mysql -uzabbix -p zabbix < schema.sql
mysql -uzabbix -p zabbix < images.sql
mysql -uzabbix -p zabbix < data.sql

cp /etc/zabbix/zabbix_server.conf{,.original}
vim /etc/zabbix/zabbix_server.conf <<EOF > /dev/null 2>&1
:%s/# DBPassword=/DBPassword=your_password/
:wq
EOF

systemctl start zabbix-server
systemctl restart httpd
```

1.3. Zabbix 3.x CentOS 7

安装脚本

```
#!/bin/bash
##################################################
# Author: Neo <netkiller@msn.com>
# Website http://netkiller.github.io
##################################################
yum localinstall -y
http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-
3.2-1.el7.noarch.rpm

yum install -y zabbix-server-mysql zabbix-web-mysql

# CREATE DATABASE `zabbix` /*!40100 COLLATE 'utf8_general_ci' */

zcat /usr/share/doc/zabbix-server-mysql-3.2.1/create.sql.gz |
mysql -uzabbix -p zabbix

cp /etc/zabbix/zabbix_server.conf{,.original}
vim /etc/zabbix/zabbix_server.conf <<EOF > /dev/null 2>&1
:%s/# DBPassword=/DBPassword=your_password/
:wq
EOF

systemctl enable httpd
systemctl enable zabbix-server

systemctl start zabbix-server
systemctl restart httpd
```

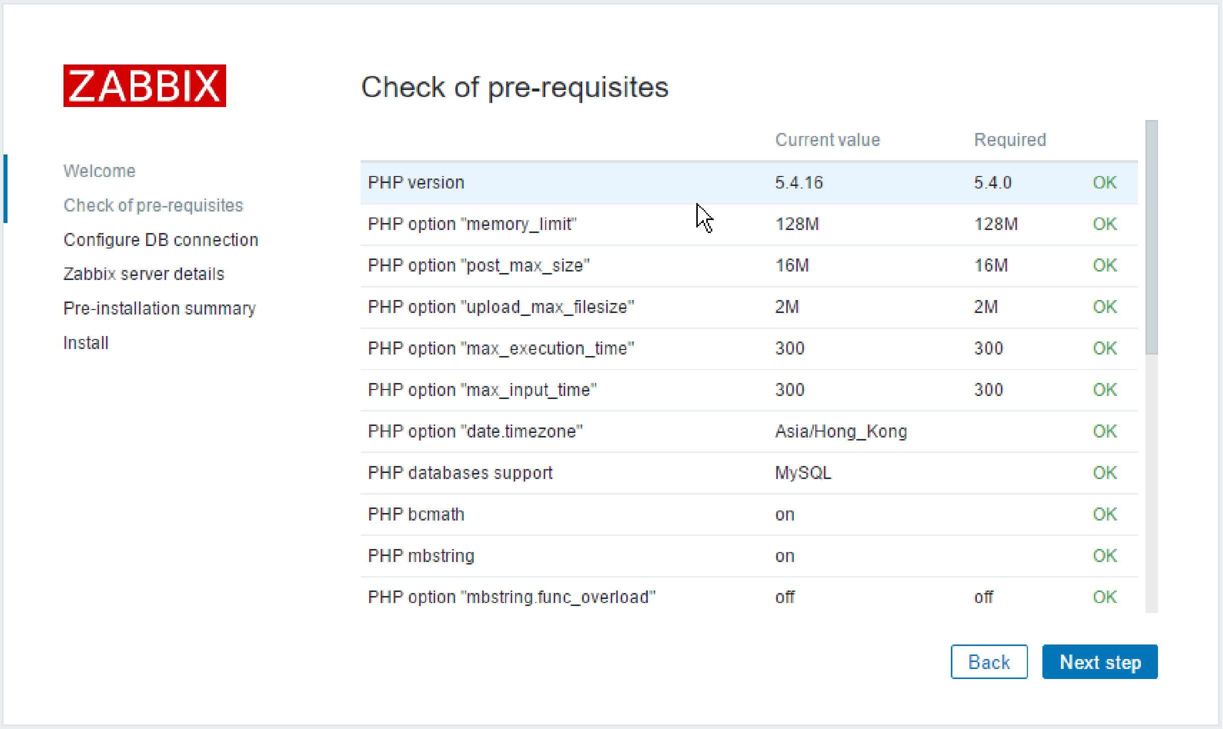
配置php.ini文件 date.timezone = Asia/Hong_Kong



Welcome to

Zabbix 3.2

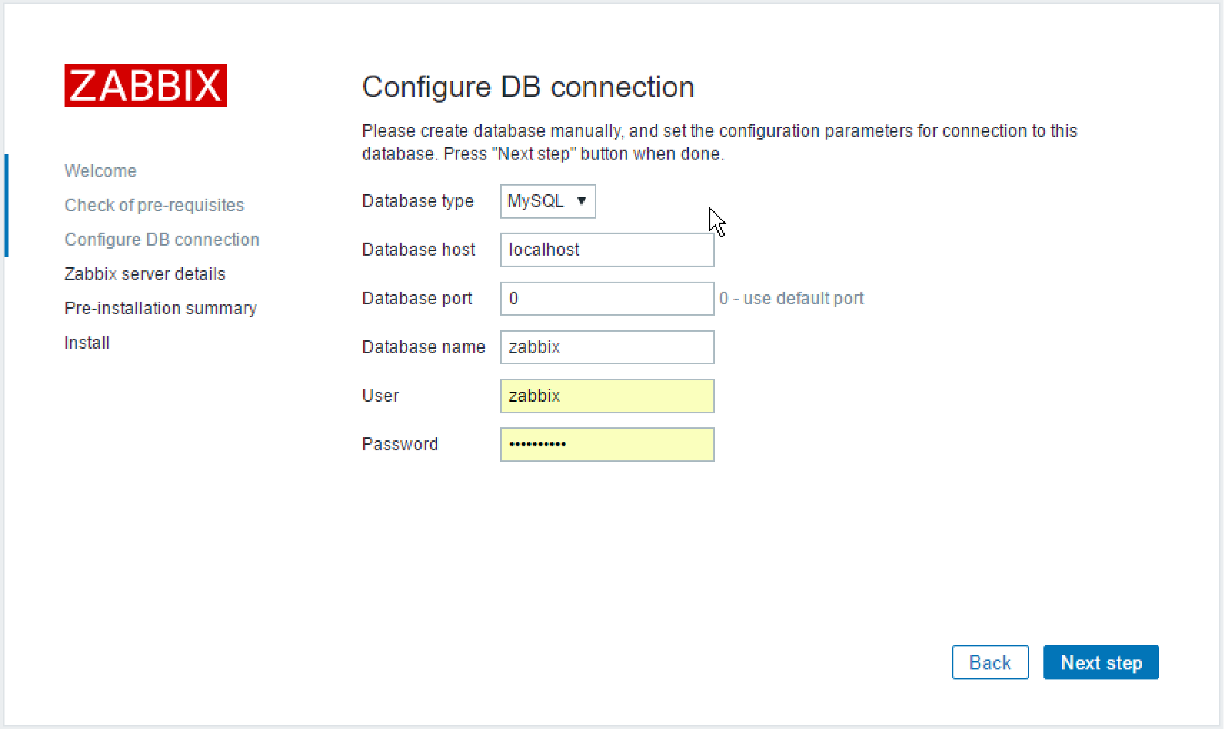
下一步



Check of pre-requisites

	Current value	Required	
PHP version	5.4.16	5.4.0	OK
PHP option "memory_limit"	128M	128M	OK
PHP option "post_max_size"	16M	16M	OK
PHP option "upload_max_filesize"	2M	2M	OK
PHP option "max_execution_time"	300	300	OK
PHP option "max_input_time"	300	300	OK
PHP option "date.timezone"	Asia/Hong_Kong		OK
PHP databases support	MySQL		OK
PHP bcmath	on		OK
PHP mbstring	on		OK
PHP option "mbstring.func_overload"	off	off	OK

检查PHP模块与配置，如果未提示错误信息点击下一步按钮

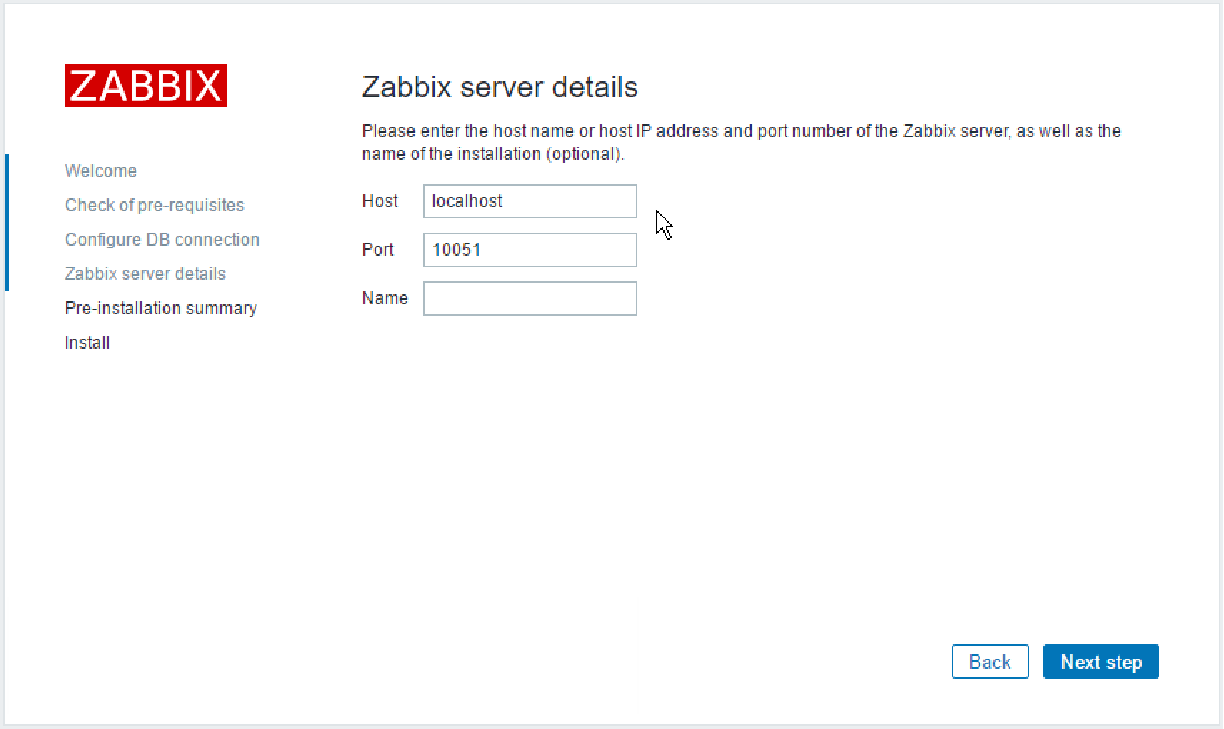


Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type	MySQL ▼
Database host	localhost
Database port	0 0 - use default port
Database name	zabbix
User	zabbix
Password	••••••••••

填写数据主机名，用户与密码，然后下一步

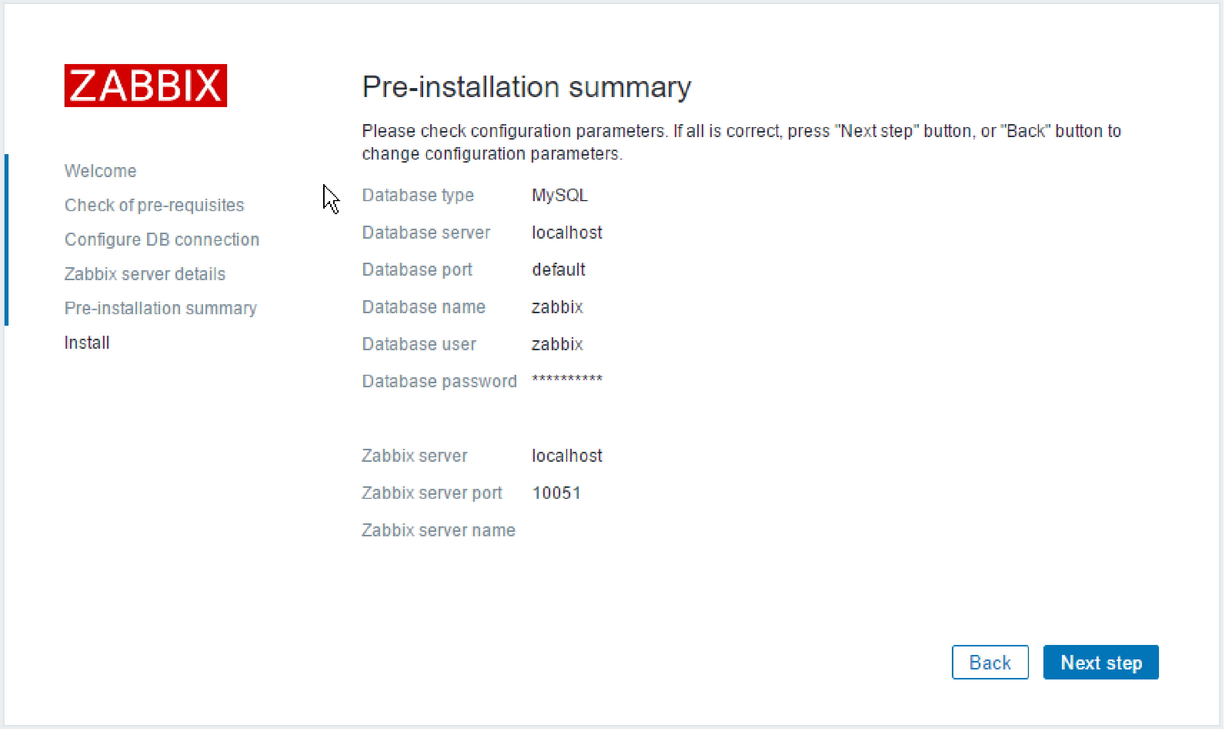


Zabbix server details

Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).

Host	localhost
Port	10051
Name	

Zabbix Server 直接点击下一步

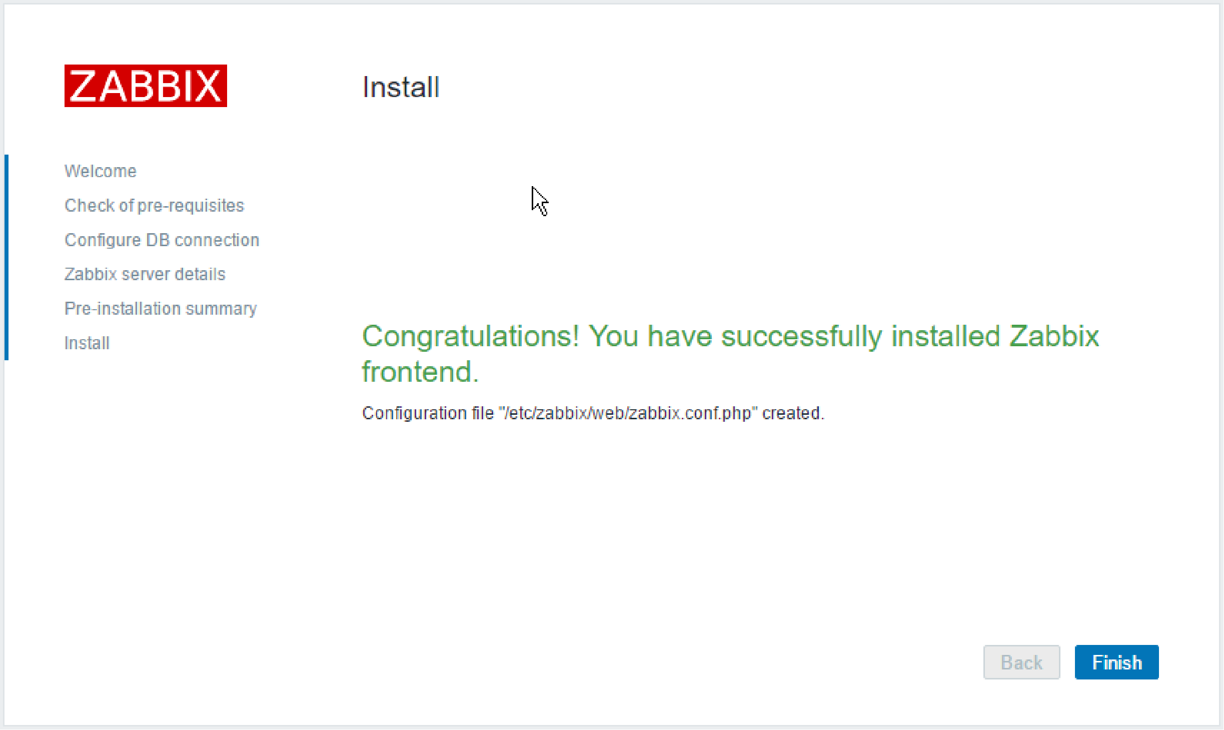


Pre-installation summary

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

Database type	MySQL
Database server	localhost
Database port	default
Database name	zabbix
Database user	zabbix
Database password	**********
Zabbix server	localhost
Zabbix server port	10051
Zabbix server name	

确认填写信息，如果不正确可以返回重新填写，确认安装点击下一步

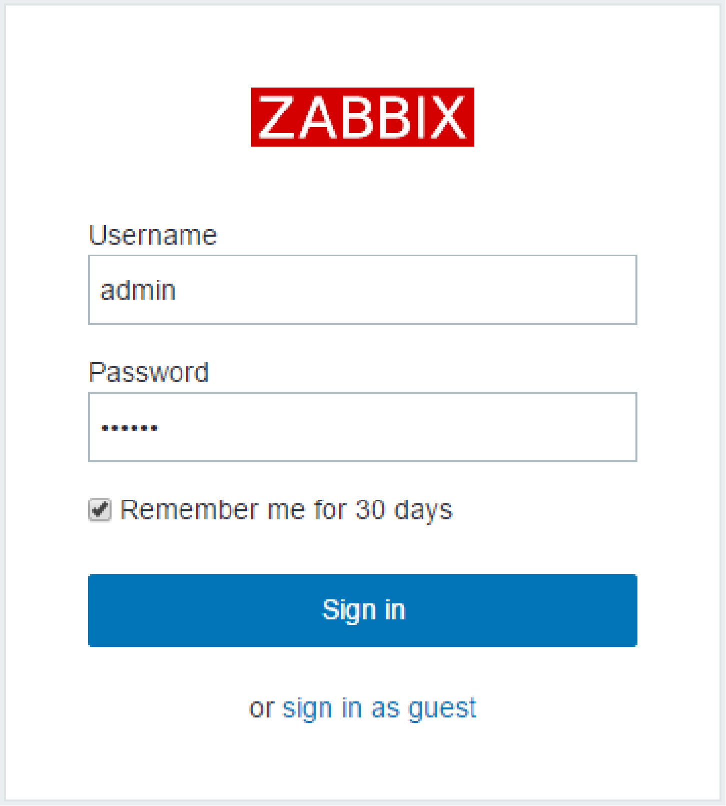


Install

Congratulations! You have successfully installed Zabbix frontend.

Configuration file "/etc/zabbix/web/zabbix.conf.php" created.

完成安装



ZABBIX

Username

admin

Password

••••••

☑ Remember me for 30 days

登陆Zabbix 默认用户名admin 密码 zabbix ， 请务必登陆后修改密码

2. web ui

http://localhost/zabbix/

user: admin

passwd: zabbix

2.1. 警告脚本

下面实现一个通过短信网关发送短信的警告脚本

首先查询 AlertScriptsPath，这是放置脚本的路径

```
# grep AlertScriptsPath /etc/zabbix/zabbix_server.conf | grep -v
^#
AlertScriptsPath=/usr/lib/zabbix/alertscripts
```

创建脚本文件/usr/lib/zabbix/alertscripts/sms.sh

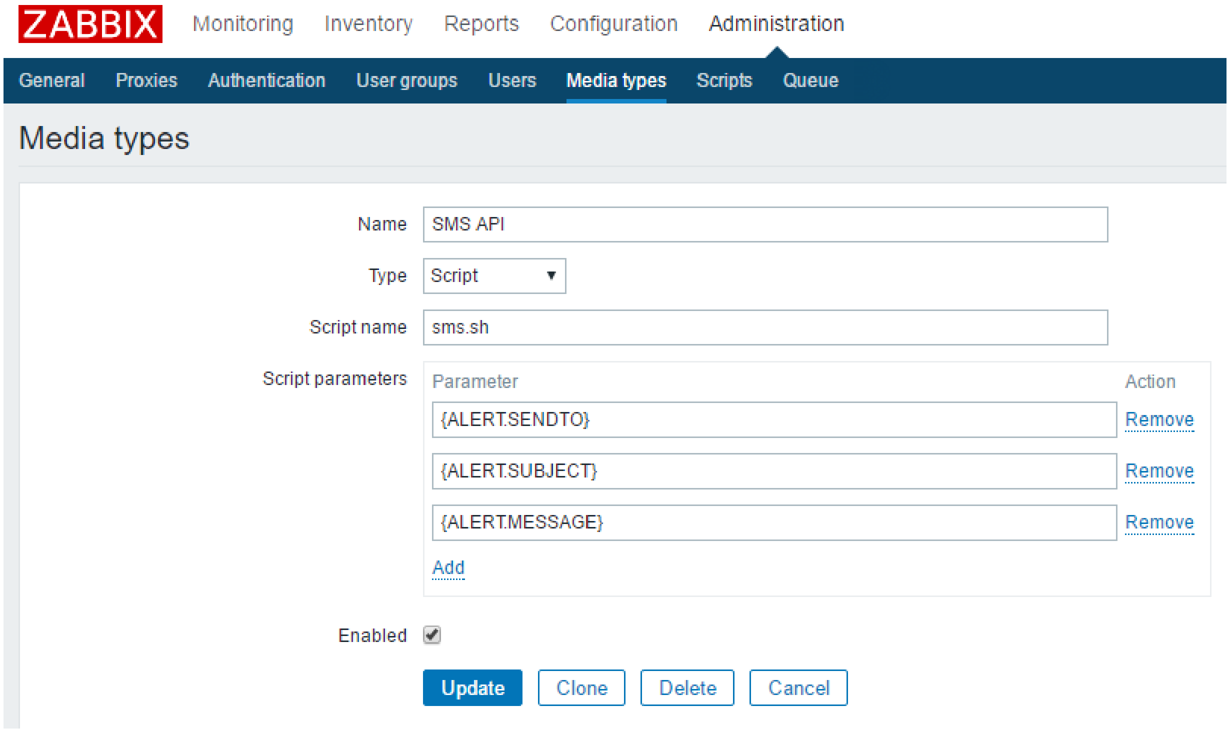
```
vim /usr/lib/zabbix/alertscripts/sms.sh

#!/bin/bash
########################################
# Author:       Neo Chen <netkiller@msn.com>
# Website:      http://www.netkiller.cn/
# Description:  zabbix alert script
# Notes:                https://github.com/oscm/zabbix
# Date:         2016-11-24
########################################
TIMEOUT=10
MOBILE=$1
MSG="$2 - $3"
########################################
LOGFILE="/tmp/sms.log"
```

```
:>"$LOGFILE"
exec 1>"$LOGFILE"
exec 2>&1

CURL="curl -s --connect-timeout ${TIMEOUT}"
URL="http://xxx.xxx.xxx.xxx/sms.php?to=${MOBILE}&msg=${MSG}"

set -x
${CURL} "${URL}"
```

测试

```
# chmod +x /usr/lib/zabbix/alertscripts/sms.sh
# /usr/lib/zabbix/alertscripts/sms.sh 13013668890 Test
Helloworld
```

进入 WEB UI 配置媒体类型，Administration/Media types/Create media type



Name	SMS API
Type	Script
Script name	sms.sh

Script parameters	{ALERT.SENDTO}
	{ALERT.SUBJECT}
	{ALERT.MESSAGE}

Enabled	☑

向脚本传递三个参数

```
{ALERT.SENDTO}
{ALERT.SUBJECT}
{ALERT.MESSAGE}
```

3. zabbix-java-gateway - Zabbix java gateway

```
yum install -y zabbix-java-gateway
```

zabbix-java-gateway 包所含内容如下

```
# rpm -ql zabbix-java-gateway
/etc/zabbix/zabbix_java_gateway.conf
/usr/lib/systemd/system/zabbix-java-gateway.service
/usr/sbin/zabbix_java_gateway
/usr/share/zabbix-java-gateway
/usr/share/zabbix-java-gateway/bin
/usr/share/zabbix-java-gateway/bin/zabbix-java-gateway-2.4.4.jar
/usr/share/zabbix-java-gateway/lib
/usr/share/zabbix-java-gateway/lib/android-json-4.3_r3.1.jar
/usr/share/zabbix-java-gateway/lib/logback-classic-0.9.27.jar
/usr/share/zabbix-java-gateway/lib/logback-console.xml
/usr/share/zabbix-java-gateway/lib/logback-core-0.9.27.jar
/usr/share/zabbix-java-gateway/lib/logback.xml
/usr/share/zabbix-java-gateway/lib/slf4j-api-1.6.1.jar
```

配置/etc/zabbix/zabbix_server.conf文件

```
# vim /etc/zabbix/zabbix_server.conf
### Option: JavaGateway
#       IP address (or hostname) of Zabbix Java gateway.
#       Only required if Java pollers are started.
#
# Mandatory: no
# Default:
JavaGateway=127.0.0.1

### Option: JavaGatewayPort
#       Port that Zabbix Java gateway listens on.
#
# Mandatory: no
# Range: 1024-32767
# Default:
JavaGatewayPort=10052

### Option: StartJavaPollers
#       Number of pre-forked instances of Java pollers.
#
# Mandatory: no
# Range: 0-1000
# Default:
StartJavaPollers=5
```

配置 /etc/zabbix/zabbix_java_gateway.conf 文件

```
# vim /etc/zabbix/zabbix_java_gateway.conf
# This is a configuration file for Zabbix Java Gateway.
# It is sourced by startup.sh and shutdown.sh scripts.

### Option: zabbix.listenIP
#       IP address to listen on.
#
# Mandatory: no
# Default:
LISTEN_IP="0.0.0.0"

### Option: zabbix.listenPort
#       Port to listen on.
#
# Mandatory: no
# Range: 1024-32767
# Default:
LISTEN_PORT=10052

### Option: zabbix.pidFile
#       Name of PID file.
#       If omitted, Zabbix Java Gateway is started as a console
application.
#
# Mandatory: no
# Default:
# PID_FILE=

PID_FILE="/var/run/zabbix/zabbix_java.pid"
```

```
### Option: zabbix.startPollers
#       Number of worker threads to start.
#
# Mandatory: no
# Range: 1-1000
# Default:
START_POLLERS=5
```

启动 zabbix-java-gateway

```
# systemctl enable zabbix-java-gateway.service
ln -s '/usr/lib/systemd/system/zabbix-java-gateway.service'
'/etc/systemd/system/multi-user.target.wants/zabbix-java-
gateway.service'

# systemctl start zabbix-java-gateway.service

systemctl restart zabbix-server
```

4. zabbix-agent

4.1. Ubuntu

```
# sudo apt-get install zabbix-agent
```

/etc/zabbix/zabbix_agent.conf

```
#Server=localhost
Server=your_server_ip_address
```

```
# vim /etc/services

zabbix-agent    10050/tcp                       #Zabbix Agent
zabbix-agent    10050/udp                       #Zabbix Agent
```

```
# sudo /etc/init.d/zabbix-agent restart
```

4.2. CentOS 7

```
yum localinstall -y http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-3.2-
1.el7.noarch.rpm

yum install -y zabbix-agent

cp /etc/zabbix/zabbix_agentd.conf{,.original}

sed -i "s/# SourceIP=/SourceIP=zabbix_server_ip/" /etc/zabbix/zabbix_agentd.conf
sed -i "s/Server=127.0.0.1/Server=zabbix_server_ip/" /etc/zabbix/zabbix_agentd.conf
sed -i "s/ServerActive=127.0.0.1/ServerActive=zabbix_server_ip/"
/etc/zabbix/zabbix_agentd.conf
sed -i "s/Hostname=Zabbix server/Hostname=Alpha Testing/" /etc/zabbix/zabbix_agentd.conf

systemctl enable zabbix-agent.service
systemctl start zabbix-agent.service

iptable -A INPUT -s zabbix_server_ip -p tcp -m state --state NEW -m tcp --dport 10050 -j
ACCEPT
```

例 2.1. zabbix-agent 配置实例

```
# grep -v "^#" /etc/zabbix/zabbix_agentd.conf | grep -v "^$"
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
SourceIP=147.90.4.87
Server=147.90.4.87
```

```
ServerActive=147.90.4.87
Hostname=Alpha Testing
Include=/etc/zabbix/zabbix_agentd.d/*.conf
```

配置完成

4.3. zabbix_agentd 命令

测试工具

```
# zabbix_agentd --test dependency.discovery
dependency.discovery                          [t|{"data":[
{"{#NAME}":"UCWEB","{#IP}":"115.84.241.16","{#PORT}":"6666"},{"{#NAME}":"Redis","
{#IP}":"115.84.241.16","{#PORT}":"6379"},{"{#NAME}":"Binary","{#IP}":"223.197.79.114","
{#PORT}":"80"},{"{#NAME}":"SMS","{#IP}":"192.230.90.194","{#PORT}":"80"},{"
{#NAME}":"CF1","{#IP}":"192.168.42.153","{#PORT}":"8080"},{"{#NAME}":"CF2","
{#IP}":"192.168.42.134","{#PORT}":"8008"},{"{#NAME}":"CF3","{#IP}":"192.168.42.177","
{#PORT}":"8080"},{"{#NAME}":"EDM","{#IP}":"47.89.27.78","{#PORT}":"80"}
]}]
```

4.4. Nginx status 监控

nginx status 监控扩展包 https://github.com/oscm/zabbix/tree/master/nginx

从 localhost 收集 nginx 状态信息

```
server {
    listen       80;
    server_name  localhost;

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

配置 zabbix_agentd

创建配置文件 /etc/zabbix/zabbix_agentd.d/userparameter_nginx.conf 内容如下:

```
############################################################
# Redis - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
############################################################
```

```
# Discovery

# Return Redis statistics
UserParameter=nginx.status[*],/srv/zabbix/libexec/nginx.sh $1
```

安装数据采集脚本，请使用 nginx.sh

```
mkdir -p /srv/zabbix/libexec
vim /srv/zabbix/libexec/nginx.sh

chmod +x /srv/zabbix/libexec/nginx.sh

# /srv/zabbix/libexec/nginx.sh
Usage /srv/zabbix/libexec/nginx.sh
{check|active|accepts|handled|requests|reading|writing|waiting}
# /srv/zabbix/libexec/nginx.sh accepts
82

# systemctl restart zabbix-agent.service
```

使用 zabbix-get 工具从 Zabbix Server 链接 Zabbix Agent 测试是否正常工作

```
Test Agent

# yum install -y zabbix-get

# zabbix_get -s <agent_ip_address> -k 'nginx.status[accepts]'
109
```

最后进入Zabbix Web界面导入模板 zbx_export_templates.xml

```
Import file: choice xml file
click "import" button

Imported successfully 表示成功导入
```

4.5. redis

获取最新模板以及脚本请访问 https://github.com/oscm/zabbix/tree/master/redis

创建代理配置文件

```
cat > /etc/zabbix/zabbix_agentd.d/userparameter_redis.conf <<'EOF'
############################################################
# Redis - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
############################################################

# Discovery

# Return Redis statistics
UserParameter=redis.status[*],redis-cli -h 127.0.0.1 -p 6379 info|grep $1|cut -d : -f2
UserParameter=redis.proc,pidof redis-server | wc -l

EOF
```

重启代理服务

```
systemctl restart zabbix-agent.service
```

测试

```
# zabbix_get -s www.netkiller.cn -k redis.status[redis_version]
2.8.19
```

导入模板文件

4.6. MongoDB

获取最新模板以及脚本请访问 https://github.com/oscm/zabbix/tree/master/mongodb

创建 Mongo 监控用户

创建监控用户

```
[root@netkiller www.netkiller.cn]# mongo -u admin -p D90YVqwmUATUeFSxfRo14  admin

> use admin
switched to db admin

> db.createUser(
   {
     user: "monitor",
     pwd: "chen",
     roles: [ "clusterMonitor"]
   }
)

Successfully added user: { "user" : "monitor", "roles" : [ "clusterMonitor" ] }
```

```
> db.auth("monitor", "netkiller")
1

> exit
bye
```

```
# echo "db.stats();" | mongo -u monitor -p chen admin
MongoDB shell version: 2.6.12
connecting to: test
{
        "db" : "test",
        "collections" : 0,
        "objects" : 0,
        "avgObjSize" : 0,
        "dataSize" : 0,
        "storageSize" : 0,
        "numExtents" : 0,
        "indexes" : 0,
        "indexSize" : 0,
        "fileSize" : 0,
        "dataFileVersion" : {

        },
        "ok" : 1
}
bye

[root@iZ62sreab5qZ www.cf88.com]# echo "db.serverStatus()" | mongo -u monitor -p chen
admin | more
MongoDB shell version: 2.6.12
connecting to: admin
{
        "host" : "iZ62sreab5qZ",
        "version" : "2.6.12",
        "process" : "mongod",
        "pid" : NumberLong(612),
        "uptime" : 852982,
        "uptimeMillis" : NumberLong(852982589),
        "uptimeEstimate" : 845317,
        "localTime" : ISODate("2016-11-23T07:02:42.899Z"),
        "asserts" : {
                "regular" : 0,
                "warning" : 0,
                "msg" : 0,
                "user" : 26,
                "rollovers" : 0
        },
        "backgroundFlushing" : {
                "flushes" : 14216,
                "total_ms" : 251465,
                "average_ms" : 17.688871693866066,
                "last_ms" : 7,
                "last_finished" : ISODate("2016-11-23T07:02:23.283Z")
        },
        "connections" : {
                "current" : 16,
                "available" : 51184,
                "totalCreated" : NumberLong(566)
        },
        "cursors" : {
                "note" : "deprecated, use server status metrics",
                "clientCursors_size" : 0,
                "totalOpen" : 0,
                "pinned" : 0,
                "totalNoTimeout" : 0,
                "timedOut" : 8
        },
        "dur" : {
                "commits" : 30,
                "journaledMB" : 0,
                "writeToDataFilesMB" : 0,
                "compression" : 0,
                "commitsInWriteLock" : 0,
                "earlyCommits" : 0,
                "timeMs" : {
                        "dt" : 3068,
                        "prepLogBuffer" : 0,
                        "writeToJournal" : 0,
                        "writeToDataFiles" : 0,
                        "remapPrivateView" : 0
                }
        },
--More--
```

Zabbix agentd 配置

```
cat > /etc/zabbix/zabbix_agentd.d/userparameter_mongodb.conf <<'EOF'
############################################################
# MongoDB - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
############################################################

# Discovery

# Return Redis statistics
UserParameter=mongodb.status[*],/srv/zabbix/libexec/mongodb.sh $1 $2 $3 $4 $5

EOF
```

安装采集脚本，创建 /srv/zabbix/libexec/mongodb.sh 文件

```
cat /srv/zabbix/libexec/mongodb.sh
#!/bin/bash
##################################################
# AUTHOR: Neo <netkiller@msn.com>
# WEBSITE: http://www.netkiller.cn
# Description：zabbix mongodb monitor
# Note：Zabbix 3.2
# DateTime: 2016-11-23
##################################################
HOST=localhost
PORT=27017
USER=monitor
PASS=chen

index=$(echo $@ | tr " " ".")

status=$(echo "db.serverStatus().${index}" |mongo -u ${USER} -p ${PASS} admin --port
${PORT}|sed -n '3p')

#check if the output contains "NumberLong"
if [[ "$status" =~ "NumberLong"   ]];then
        echo $status|sed -n 's/NumberLong(//p'|sed -n 's/)//p'
else
        echo $status
fi


# chmod +x /srv/zabbix/libexec/mongodb.sh

# /srv/zabbix/libexec/mongodb.sh version
2.6.12

# systemctl restart zabbix-agent.service
```

Zabbix server 测试

```
[root@netkiller ~]# zabbix_get -s www.netkiller.cn -k mongodb.status[ok]
1
[root@netkiller ~]# zabbix_get -s www.netkiller.cn -k mongodb.status[version]
2.6.12
```

测试成功后导入模板

监控内容如下

```
链接数监控(当前连接数和可用连接数)
mongodb current mongodb.status[connections,current]
mongodb available mongodb.status[connections,available]

流量监控(每秒请求数,出站流量,入站流量)
mongodb mongodb.status[network,numRequests]
mongodb mongodb.status[network,bytesOut]
mongodb mongodb.status[network,bytesIn]

命令统计(查询，更新，插入，删除......)
mongodb query/s mongodb.status[opcounters,query]
mongodb update/s mongodb.status[opcounters,update]
mongodb insert/s mongodb.status[opcounters,insert]
mongodb getmore/s mongodb.status[opcounters,getmore]
mongodb delete/s mongodb.status[opcounters,delete]
mongodb command/s mongodb.status[opcounters,command]

内存监控
mongodb mem virtual mongodb.status[mem,virtual]
mongodb mem resident mongodb.status[mem,resident]
mongodb mem mapped mongodb.status[mem,mapped]
mongodb mem mappedWithJournal mongodb.status[mem,mappedWithJournal]

复制监控
mongodb repl mongodb.status[repl,ismaster]

锁监控
# zabbix_get -s www.chuangfu24.net -k mongodb.status[locks,admin,timeAcquiringMicros,r]
```

4.7. PHP-FPM

获取最新模板以及脚本请访问 https://github.com/oscm/zabbix/tree/master/php-fpm

启用 php-fpm status 功能

这里假设你是采用 yum install php-fpm 方式安装的

```
sed -i "s/;pm.status_path/pm.status_path/" /etc/php-fpm.d/www.conf
sed -i "s/;ping/ping/" /etc/php-fpm.d/www.conf

systemctl reload php-fpm
```

配置 nginx

```
server {
    listen       80;
    server_name  localhost;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

        location /stub_status {
        stub_status on;
        access_log   off;
        allow 127.0.0.1;
        deny all;
    }
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 127.0.0.1;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
                fastcgi_param SCRIPT_FILENAME $fastcgi_script_name;
        include fastcgi_params;
    }
}
```

配置 Zabbix 代理

采集脚本 /srv/zabbix/libexec/php-fpm.xml.sh

```
#!/bin/bash
##################################################
# AUTHOR: Neo <netkiller@msn.com>
# WEBSITE: http://www.netkiller.cn
# Description：zabbix 通过 status 模块监控 php-fpm
# Note：Zabbix 3.2
# DateTime: 2016-11-22
##################################################

HOST="localhost"
PORT="80"
status="status"

function query() {
        curl -s http://${HOST}:${PORT}/${status}?xml | grep "$1" | awk -F'>|<' '{ print
$3}'
}

if [ $# == 0 ]; then
                echo $"Usage $0 {pool|process-manager|start-time|start-since|accepted-
conn|listen-queue|max-listen-queue|listen-queue-len|idle-processes|active-
processes|total-processes|max-active-processes|max-children-reached|slow-requests}"
                exit
else
        query "$1"
fi
```

创建zabbix代理配置文件 /etc/zabbix/zabbix_agentd.d/userparameter_php-fpm.conf

```
############################################################
# Netkiller PHP-FPM - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
############################################################
```

```
# Discovery

# Return statistics
UserParameter=php-fpm.status[*],/srv/zabbix/libexec/php-fpm.xml.sh $1
```

从zabbix server 运行下面命令测试是否可以正确获得数据

```
# zabbix_get -s node.netkiller.cn -k 'php-fpm.status[listen-queue-len]'
128
```

php-fpm 监控参数

php-fpm 可以带参数json、xml、html并且前面三个参数可以分别和full做一个组合。

```
status 详解
-----
pool – fpm池子名称，大多数为www
process manager – 进程管理方式,值：static, dynamic or ondemand. dynamic
start time – 启动日期,如果reload了php-fpm，时间会更新
start since – 运行时长
accepted conn – 当前池子接受的请求数
listen queue – 请求等待队列，如果这个值不为0，那么要增加FPM的进程数量
max listen queue – 请求等待队列最高的数量
listen queue len – socket等待队列长度
idle processes – 空闲进程数量
active processes – 活跃进程数量
total processes – 总进程数量
max active processes – 最大的活跃进程数量（FPM启动开始算）
max children reached - 大道进程最大数量限制的次数，如果这个数量不为0，那说明你的最大进程数量太小了,
请改大一点。
slow requests – 启用了php-fpm slow-log，缓慢请求的数量

full详解
-----
pid – 进程PID，可以单独kill这个进程.
state – 当前进程的状态 (Idle, Running, …)
start time – 进程启动的日期
start since – 当前进程运行时长
requests – 当前进程处理了多少个请求
request duration – 请求时长（微妙）
request method – 请求方法 (GET, POST, …)
request URI – 请求URI
content length – 请求内容长度 (仅用于 POST)
user – 用户 (PHP_AUTH_USER) (or ‘-’ 如果没设置)
script – PHP脚本 (or ‘-’ if not set)
last request cpu – 最后一个请求CPU使用率。
last request memorythe - 上一个请求使用的内存
```

```
[root@netkiller tmp]# curl http://localhost/status
pool:                 www
process manager:      dynamic
start time:           25/Nov/2016:10:31:32 +0800
start since:          2337
accepted conn:        191
listen queue:         0
max listen queue:     0
listen queue len:     128
idle processes:       5
active processes:     1
total processes:      6
max active processes: 1
max children reached: 0
slow requests:        0
[root@netkiller tmp]# curl http://localhost/status?full
pool:                 www
process manager:      dynamic
start time:           25/Nov/2016:10:31:32 +0800
start since:          2343
accepted conn:        192
listen queue:         0
max listen queue:     0
listen queue len:     128
idle processes:       5
active processes:     1
total processes:      6
max active processes: 1
max children reached: 0
slow requests:        0

************************
pid:                  27329
state:                Running
start time:           25/Nov/2016:10:31:32 +0800
start since:          2343
requests:             33
request duration:     140
request method:       GET
request URI:          /status?full
content length:       0
user:                 -
script:               -
last request cpu:     0.00
last request memory:  0

************************
pid:                  27330
state:                Idle
start time:           25/Nov/2016:10:31:32 +0800
start since:          2343
requests:             32
request duration:     111
request method:       GET
request URI:          /status?xml
content length:       0
user:                 -
script:               -
last request cpu:     0.00
last request memory:  262144

************************
pid:                  27331
state:                Idle
start time:           25/Nov/2016:10:31:32 +0800
start since:          2343
```

```
requests:               32
request duration:       110
request method:         GET
request URI:            /status?xml
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144

************************
pid:                    27332
state:                  Idle
start time:             25/Nov/2016:10:31:32 +0800
start since:            2343
requests:               32
request duration:       106
request method:         GET
request URI:            /status?xml
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144

************************
pid:                    27333
state:                  Idle
start time:             25/Nov/2016:10:31:32 +0800
start since:            2343
requests:               32
request duration:       90
request method:         GET
request URI:            /status
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144

************************
pid:                    27557
state:                  Idle
start time:             25/Nov/2016:10:33:43 +0800
start since:            2212
requests:               31
request duration:       131
request method:         GET
request URI:            /status?xml
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144
```

```
[root@netkiller tmp]# curl http://localhost/status?json
```

```
{"pool":"www","process manager":"dynamic","start time":1480041092,"start
since":2308,"accepted conn":181,"listen queue":0,"max listen queue":0,"listen queue
len":128,"idle processes":5,"active processes":1,"total processes":6,"max active
processes":1,"max children reached":0,"slow requests":0}
```

```
[root@netkiller tmp]# curl http://localhost/status?xml
<?xml version="1.0" ?>
<status>
<pool>www</pool>
<process-manager>dynamic</process-manager>
<start-time>1480041092</start-time>
<start-since>2520</start-since>
<accepted-conn>226</accepted-conn>
<listen-queue>0</listen-queue>
<max-listen-queue>0</max-listen-queue>
<listen-queue-len>128</listen-queue-len>
<idle-processes>5</idle-processes>
<active-processes>1</active-processes>
<total-processes>6</total-processes>
<max-active-processes>1</max-active-processes>
<max-children-reached>0</max-children-reached>
<slow-requests>0</slow-requests>
```

```
[root@netkiller tmp]# curl http://localhost/status?html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head><title>PHP-FPM Status Page</title></head>
<body>
<table>
<tr><th>pool</th><td>www</td></tr>
<tr><th>process manager</th><td>dynamic</td></tr>
<tr><th>start time</th><td>25/Nov/2016:10:31:32 +0800</td></tr>
<tr><th>start since</th><td>2486</td></tr>
<tr><th>accepted conn</th><td>216</td></tr>
<tr><th>listen queue</th><td>0</td></tr>
<tr><th>max listen queue</th><td>0</td></tr>
<tr><th>listen queue len</th><td>128</td></tr>
<tr><th>idle processes</th><td>5</td></tr>
<tr><th>active processes</th><td>1</td></tr>
<tr><th>total processes</th><td>6</td></tr>
<tr><th>max active processes</th><td>1</td></tr>
<tr><th>max children reached</th><td>0</td></tr>
<tr><th>slow requests</th><td>0</td></tr>
</table>
</body></html>
```

4.8. Elasticsearch

获取最新模板以及脚本请访问 https://github.com/oscm/zabbix/tree/master/elasticsearch

首先导入模板 https://github.com/oscm/zabbix/blob/master/elasticsearch/zbx_export_templates.xml

安装采集脚本

一步步运行下面脚本即可

```
# yum install -y python34
# wget https://raw.githubusercontent.com/oscm/zabbix/master/elasticsearch/elasticsearch
-P /srv/zabbix/libexec
# chmod +x /srv/zabbix/libexec/elasticsearch
# /srv/zabbix/libexec/elasticsearch indices _all.total.flush.total_time_in_millis
25557
```

配置Zabbix代理

运行脚本安装代理配置文件

```
# wget
https://raw.githubusercontent.com/oscm/zabbix/master/elasticsearch/userparameter_elastic
search.conf -P /etc/zabbix/zabbix_agentd.d/
# systemctl restart zabbix-agent
```

测试Zabbix Agent 工作是否正常

```
# zabbix_get -s 10.47.33.14 -k
'elasticsearch.status[indices,_all.total.flush.total_time_in_millis]'
25557
```

4.9. Postfix

获取最新模板以及脚本请访问 https://github.com/oscm/zabbix/tree/master/postfix

首先导入模板 https://github.com/oscm/zabbix/blob/master/postfix/zbx_export_templates.xml

安装采集脚本

一步步运行下面脚本即可

```
# chmod +r /var/log/maillog
# mkdir -p /srv/zabbix/libexec
# yum install -y logcheck
# wget https://raw.githubusercontent.com/oscm/zabbix/master/postfix/postfix -P
/srv/zabbix/libexec
# chmod +x /srv/zabbix/libexec/postfix
```

测试脚本

```
# /srv/zabbix/libexec/postfix queue active
1418
```

userparameter_postfix.conf

```
# wget
https://raw.githubusercontent.com/oscm/zabbix/master/postfix/userparameter_postfix.conf
-P /etc/zabbix/zabbix_agentd.d/
# systemctl restart zabbix-agent
```

```
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k 'agent.ping'
1
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k 'postfix[queue,active]'
1140
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k 'postfix[queue,deferred]'
149
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k 'postfix[log,sent]'
10931
```

4.10. TCP stats

```
curl -s https://raw.githubusercontent.com/oscm/shell/master/monitor/zabbix/zabbix-
agent/tcpstats.sh | bash
```

采集脚本

```
# zabbix_agentd --test tcp.stats[FIN-WAIT-2]
tcp.stats[FIN-WAIT-2]                         [t|130]
```

Zabbix

```
zabbix_get -s 10.24.15.18 -k 'tcp.stats[LISTEN]'
```

4.11. 应用依赖检查

```
curl -s https://raw.githubusercontent.com/oscm/shell/master/monitor/zabbix/zabbix-
agent/dependency.sh | bash
```

4.12. Oracle

采集脚本

创建JDBC配置文件　/srv/zabbix/conf/jdbc.properties

```
# Oracle 单机环境
jdbc.url=jdbc:oracle:thin:@//172.16.0.10:1521/oral
# Oracle RAC 环境
# jdbc.url=jdbc\:oracle\:thin\:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=172.16.0.5)
(PORT=1521))(LOAD_BALANCE=yes)(FAILOVER=ON)(CONNECT_DATA=(SERVER=DEDICATED)
(SERVICE_NAME=oral)(FAILOVER_MODE=(TYPE=SESSION)(METHOD=BASIC))))
jdbc.username=neo
jdbc.password=netkiller
```

```

```

第 3 章 ElasticSearch + Logstash + Kibana

官方网站 https://www.elastic.co

环境准备:

操作系统： CentOS 7

Java 1.8

Redis

ElasticSearch + Logstash + Kibana 均使用 5.2 版本

以下安装均使用 Netkiller OSCM 脚本一键安装

1. 安装

1.1. 6.x

```
curl -s https://raw.githubusercontent.com/oscm/shell/master/search/elastic/elastic-6.x.sh | bash
```

1.2. ElasticSearch + Logstash + Kibana 安装

ElasticSearch 安装

粘贴下面命令到Linux控制台即可一键安装

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elas
ticsearch/elasticsearch-5.x.sh | bash
```

Kibana 安装

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/log/kibana/
kibana-5.x.sh | bash
```

Logstash 安装

```
                                        curl -s
https://raw.githubusercontent.com/oscm/shell/master/log/kibana/
logstash-5.x.sh | bash
```

从 5.x 升级到 6.x

升级仓库

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elas
tic/elastic-6.x.sh | bash
```

```
yum update logstash
```

2. logstash 命令简单应用

2.1. -e 命令行运行

logstash -e "input {stdin{}} output {stdout{}}"

```
/usr/share/logstash/bin/logstash  -e 'input{file {path =>
"/etc/centos-release" start_position => "beginning"}} output {
stdout {}}'
```

2.2. -f 指定配置文件

```
/usr/share/logstash/bin/logstash -f stdin.conf

/usr/share/logstash/bin/logstash -f jdbc.conf --path.settings
/etc/logstash --path.data /tmp
```

2.3. -t：测试配置文件是否正确，然后退出。

```
root@netkiller ~/logstash % /usr/share/logstash/bin/logstash -t
-f test.conf
WARNING: Default JAVA_OPTS will be overridden by the JAVA_OPTS
defined in the environment. Environment JAVA_OPTS are -server -
Xms2048m -Xmx4096m
WARNING: Could not find logstash.yml which is typically located
in $LS_HOME/config or /etc/logstash. You can specify the path
using --path.settings. Continuing using the defaults
Could not find log4j2 configuration at path
/usr/share/logstash/config/log4j2.properties. Using default
config which logs errors to the console
Configuration OK
```

2.4. -l：日志输出的地址

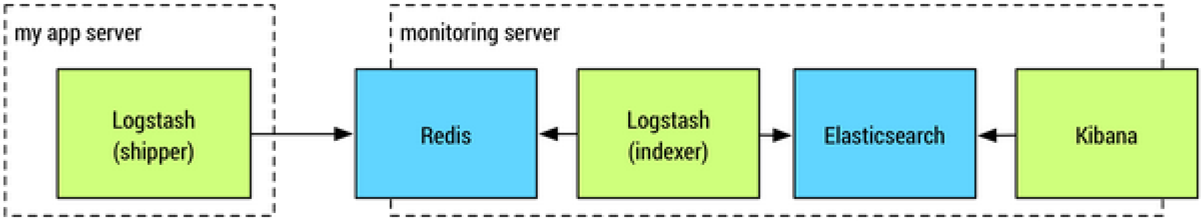
默认就是stdout直接在控制台中输出

2.5. log.level 启动Debug模式

```
% /usr/share/logstash/bin/logstash -f nginx.conf --path.settings
/etc/logstash --log.level debug
```

3. 配置 Broker(Redis)

3.1. indexer



/etc/logstash/conf.d/indexer.conf

```
input {
  redis {
    host => "127.0.0.1"
    port => "6379"
    key => "logstash:demo"
    data_type => "list"
    codec  => "json"
    type => "logstash-redis-demo"
    tags => ["logstashdemo"]
  }
}

output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
}
```

测试

```
# redis-cli
```

```
127.0.0.1:6379> RPUSH logstash:demo "{\"time\": \"2012-01-01T10:20:00\", \"message\": \"logstash demo message\"}"
(integer) 1
127.0.0.1:6379> exit
```

如果执行成功日志如下

```
# cat /var/log/logstash/logstash-plain.log
[2017-03-22T15:54:36,491][INFO ]
[logstash.outputs.elasticsearch] Elasticsearch pool URLs
updated {:changes=>{:removed=>[], :added=>
[http://127.0.0.1:9200/]}}
[2017-03-22T15:54:36,496][INFO ]
[logstash.outputs.elasticsearch] Running health check to see if
an Elasticsearch connection is working
{:healthcheck_url=>http://127.0.0.1:9200/, :path=>"/"}
[2017-03-22T15:54:36,600][WARN ]
[logstash.outputs.elasticsearch] Restored connection to ES
instance {:url=>#<URI::HTTP:0x20dae6aa
URL:http://127.0.0.1:9200/>}
[2017-03-22T15:54:36,601][INFO ]
[logstash.outputs.elasticsearch] Using mapping template from
{:path=>nil}
[2017-03-22T15:54:36,686][INFO ]
[logstash.outputs.elasticsearch] Attempting to install template
{:manage_template=>{"template"=>"logstash-*", "version"=>50001,
"settings"=>{"index.refresh_interval"=>"5s"}, "mappings"=>
{"_default_"=>{"_all"=>{"enabled"=>true, "norms"=>false},
"dynamic_templates"=>[{"message_field"=>
{"path_match"=>"message", "match_mapping_type"=>"string",
"mapping"=>{"type"=>"text", "norms"=>false}}},
{"string_fields"=>{"match"=>"*",
"match_mapping_type"=>"string", "mapping"=>{"type"=>"text",
"norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword"}}}}}],
"properties"=>{"@timestamp"=>{"type"=>"date",
"include_in_all"=>false}, "@version"=>{"type"=>"keyword",
"include_in_all"=>false}, "geoip"=>{"dynamic"=>true,
"properties"=>{"ip"=>{"type"=>"ip"}, "location"=>
{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_float"},
"longitude"=>{"type"=>"half_float"}}}}}}}}
[2017-03-22T15:54:36,693][INFO ]
```

```
[logstash.outputs.elasticsearch] Installing elasticsearch
template to _template/logstash
[2017-03-22T15:54:36,780][INFO ]
[logstash.outputs.elasticsearch] New Elasticsearch output
{:class=>"LogStash::Outputs::ElasticSearch", :hosts=>[#
<URI::Generic:0x2f9efc89 URL://127.0.0.1>]}
[2017-03-22T15:54:36,787][INFO ][logstash.pipeline        ]
Starting pipeline {"id"=>"main", "pipeline.workers"=>8,
"pipeline.batch.size"=>125, "pipeline.batch.delay"=>5,
"pipeline.max_inflight"=>1000}
[2017-03-22T15:54:36,792][INFO ][logstash.inputs.redis    ]
Registering Redis {:identity=>"redis://@127.0.0.1:6379/0
list:logstash:demo"}
[2017-03-22T15:54:36,793][INFO ][logstash.pipeline        ]
Pipeline main started
[2017-03-22T15:54:36,838][INFO ][logstash.agent           ]
Successfully started Logstash API endpoint {:port=>9600}
[2017-03-22T15:55:10,018][WARN ][logstash.runner          ]
SIGTERM received. Shutting down the agent.
[2017-03-22T15:55:10,024][WARN ][logstash.agent           ]
stopping pipeline {:id=>"main"}
```

3.2. shipper

```
input {
  file {
    path => [ "/var/log/nginx/access.log" ]
    start_position => "beginning"
  }
}

filter {
  grok {
    match => { "message" => "%{NGINXACCESS}" }
    add_field => { "type" => "access" }
  }
  date {
    match => [ "timestamp" , "dd/MMM/YYYY:HH:mm:ss Z" ]
  }
  geoip {
    source => "clientip"
  }
}

output {
  redis {
    host => "127.0.0.1"
    port => 6379
    data_type => "list"
    key => "logstash:demo"
  }
}
```

4. logstash 配置项

4.1. input

标准输入输出

```
                                        root@netkiller ~ %
/usr/share/logstash/bin/logstash -e "input {stdin{}} output {stdout{}}"
                                        Helloworld
                                        ERROR StatusLogger No log4j2 configuration file
found. Using default configuration: logging only errors to the console.
                                        WARNING: Could not find logstash.yml which is
typically located in $LS_HOME/config or /etc/logstash. You can specify the path using --
path.settings. Continuing using the defaults
                                        Could not find log4j2 configuration at path
//usr/share/logstash/config/log4j2.properties. Using default config which logs to
console
                                        18:03:38.340 [[main]-pipeline-manager] INFO
logstash.pipeline - Starting pipeline {"id"=>"main", "pipeline.workers"=>8,
"pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max_inflight"=>1000}
                                        18:03:38.356 [[main]-pipeline-manager] INFO
logstash.pipeline - Pipeline main started
                                        The stdin plugin is now waiting for input:
                                        2017-08-03T10:03:38.375Z localhost Helloworld
                                        18:03:38.384 [Api Webserver] INFO logstash.agent
- Successfully started Logstash API endpoint {:port=>9601}
```

rubydebug

rubydebug提供以json格式输出到屏幕

```
                                        root@netkiller ~ %
/usr/share/logstash/bin/logstash -e 'input{stdin{}}output{stdout{codec=>rubydebug}}'
                                        My name is neo
                                        ERROR StatusLogger No log4j2 configuration file
found. Using default configuration: logging only errors to the console.
                                        WARNING: Could not find logstash.yml which is
typically located in $LS_HOME/config or /etc/logstash. You can specify the path using --
path.settings. Continuing using the defaults
                                        Could not find log4j2 configuration at path
//usr/share/logstash/config/log4j2.properties. Using default config which logs to
console
                                        18:05:02.734 [[main]-pipeline-manager] INFO
logstash.pipeline - Starting pipeline {"id"=>"main", "pipeline.workers"=>8,
"pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max_inflight"=>1000}
                                        18:05:02.747 [[main]-pipeline-manager] INFO
logstash.pipeline - Pipeline main started
                                        The stdin plugin is now waiting for input:
                                        {
                                        "@timestamp" => 2017-08-03T10:05:02.764Z,
                                        "@version" => "1",
                                        "host" => "localhost",
                                        "message" => "My name is neo"
                                        }
```

```
                    18:05:02.782 [Api Webserver] INFO logstash.agent
- Successfully started Logstash API endpoint {:port=>9601}
```

本地文件

```
input {
 file {
  type => "syslog"
  path => [ "/var/log/maillog", "/var/log/messages", "/var/log/secure" ]
  start_position => "beginning"
 }
}
output {
 stdout { codec => rubydebug }
 elasticsearch {
  hosts => ["127.0.0.1:9200"]
 }
}
```

start_position => "beginning" 从头开始读，如果没有这个选项，只会读取最后更新的数据。

指定文件类型

```
input {
 file { path =>"/var/log/messages" type =>"syslog"}
 file { path =>"/var/log/apache/access.log" type =>"apache"}
}
```

Nginx

```
input {
    file {
        type => "nginx_access"
        path => ["/usr/share/nginx/logs/test.access.log"]
    }
}
output {
    redis {
        host => "localhost"
        data_type => "list"
        key => "logstash:redis"
    }
}
```

TCP/UDP

```
input {
  file {
    type => "syslog"
    path => [ "/var/log/secure", "/var/log/messages", "/var/log/syslog" ]
  }
  tcp {
    port => "5145"
    type => "syslog-network"
  }
  udp {
    port => "5145"
    type => "syslog-network"
  }
}
output {
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
}
```

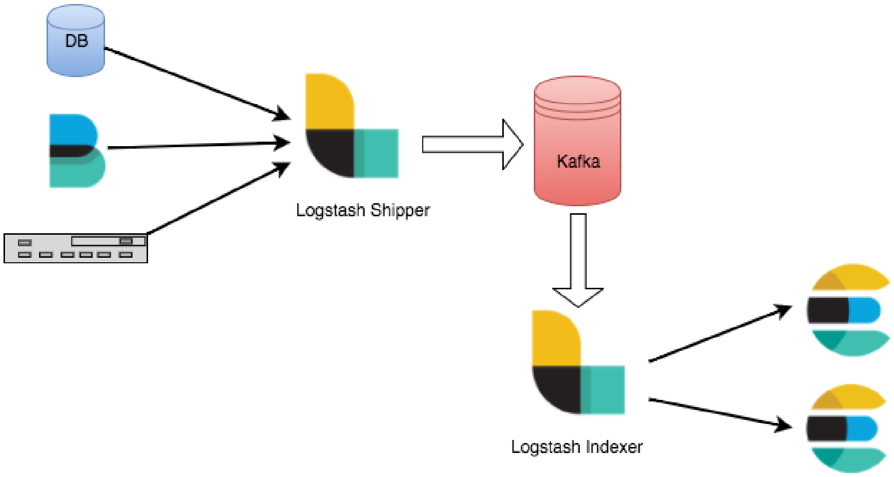
Redis

```
input {
  redis {
    host => "127.0.0.1"
    port => "6379"
    key => "logstash:demo"
    data_type => "list"
    codec  => "json"
    type => "logstash-redis-demo"
    tags => ["logstashdemo"]
  }
}

output {
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
}
```

指定 Database 10

```
root@netkiller /etc/logstash/conf.d % cat spring-boot-redis.conf
input {
 redis {
  codec => json
  host => "localhost"
  port => 6379
  db => 10
  key => "logstash:redis"
  data_type => "list"
 }
}

output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logstash-api"
  }
}
```

Kafka



```
input {
  kafka {
   zk_connect => "kafka:2181"
   group_id => "logstash"
   topic_id => "apache_logs"
   consumer_threads => 16
  }
}
```

jdbc

```
root@netkiller /etc/logstash/conf.d % cat jdbc.conf
input {
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "123456"
    schedule => "* * * * *"
    statement => "select * from article where id > :sql_last_value"
    use_column_value => true
    tracking_column => "id"
    tracking_column_type => "numeric"
    record_last_run => true
    last_run_metadata_path => "/var/tmp/article.last"
  }
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "123456"
    schedule => "* * * * *"     #定时cron的表达式,这里是每分钟执行一次
    statement => "select * from article where ctime > :sql_last_value"
    use_column_value => true
    tracking_column => "ctime"
    tracking_column_type => "timestamp"
    record_last_run => true
    last_run_metadata_path => "/var/tmp/article-ctime.last"
  }

}
output {
    elasticsearch {
        hosts => "localhost:9200"
        index => "information"
        document_type => "article"
        document_id => "%{id}"
        action => "update"
        doc_as_upsert => true
    }
}
```

4.2. filter

日期格式化

系统默认是 ISO8601 如果需要转换为 yyyy-MM-dd-HH:mm:ss 参考:

```
filter {
  date {
    match => [ "ctime", "yyyy-MM-dd HH:mm:ss" ]
    locale => "cn"
  }
 date {
    match => [ "mtime", "yyyy-MM-dd HH:mm:ss" ]
    locale => "cn"
  }
}
```

patterns

创建匹配文件 /usr/share/logstash/patterns

```
mkdir /usr/share/logstash/patterns
vim /usr/share/logstash/patterns

NGUSERNAME [a-zA-Z\.\@\-\+_%]+
NGUSER %{NGUSERNAME}
NGINXACCESS %{IPORHOST:clientip} %{NGUSER:ident} %{NGUSER:auth} \[%
{HTTPDATE:timestamp}\] "%{WORD:verb} %{URIPATHPARAM:request} HTTP/%{NUMBER:httpversion}"
%{NUMBER:response} (?:%{NUMBER:bytes}|-) (?:"(?:%{URI:referrer}|-)"|%{QS:referrer}) %
{QS:agent}
```

```
filter {
  if [type] == "nginx-access" {
    grok {
      match => { "message" => "%{NGINXACCESS}" }
    }
  }
}
```

syslog

```
input {
  file {
    type => "syslog"
    path => [ "/var/log/*.log", "/var/log/messages", "/var/log/syslog" ]
    sincedb_path => "/opt/logstash/sincedb-access"
  }
  syslog {
    type => "syslog"
    port => "5544"
  }
}

filter {
  grok {
    type => "syslog"
    match => [ "message", "%{SYSLOGBASE2}" ]
    add_tag => [ "syslog", "grokked" ]
  }
}

output {
 elasticsearch { host => "elk.netkiller.cn" }
}
```

csv

```
input {
    file {
        type => "SSRCode"
        path => "/SD/2015*/01*/*.csv"
        start_position => "beginning"
    }
}

filter {
        csv {
                columns => ["Code","Source"]
                separator => ","
        }
        kv {
                source => "uri"
                field_split => "&?"
                value_split => "="
        }

}

# output logs to console and to elasticsearch
output {
    stdout {}
    elasticsearch {
        hosts => ["172.16.1.1:9200"]
    }
}
```

使用ruby 处理 CSV文件

```
input {
    stdin {}
}
filter {
    ruby {
        init => "
            begin
                @@csv_file    = 'output.csv'
                @@csv_headers = ['A','B','C']
                if File.zero?(@@csv_file) || !File.exist?(@@csv_file)
                    CSV.open(@@csv_file, 'w') do |csv|
                        csv << @@csv_headers
                    end
                end
            end
        "
        code => "
            begin
                event['@metadata']['csv_file']    = @@csv_file
                event['@metadata']['csv_headers'] = @@csv_headers
            end
        "
    }
    csv {
        columns => ["a", "b", "c"]
    }
}
output {
    csv {
        fields => ["a", "b", "c"]
        path   => "%{[@metadata][csv_file]}"
    }
    stdout {
        codec => rubydebug {
            metadata => true
        }
    }
}
```

测试

```
echo "1,2,3\n4,5,6\n7,8,9" | ./bin/logstash -f csv-headers.conf
```

输出结果

```
A,B,C
1,2,3
4,5,6
7,8,9
```

执行 ruby 代码

日期格式化, 将ISO 8601日期格式转换为 %Y-%m-%d %H:%M:%S

保存下面内容到配置文件data.conf

```
input {
        stdin{}
}
filter {

        ruby {
                init => "require 'time'"
        code => "event.set('ctime', event.get('ctime').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }

        ruby {
                init => "require 'time'"
        code => "event.set('mtime', event.get('mtime').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }
}
output {

        stdout {
                codec => rubydebug
        }

}
```

/usr/share/logstash/bin/logstash -f date.conf

grok debug 工具

http://grokdebug.herokuapp.com

4.3. output

stdout

```
                                        output {
                                        stdout { codec => rubydebug }
                                        }
```

file 写入文件

```
output {
    file {
        path => "/path/to/%{host}/%{+yyyy}/%{+MM}/%{+dd}.log.gz"
        message_format => "%{message}"
        gzip => true
    }
}
```

elasticsearch

```
output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logging"
  }
}
```

自定义 index

配置实现每日切割一个 index

```
index => "logstash-%{+YYYY.MM.dd}"

"_index" : "logstash-2017.03.22"
```

index 自定义 logstash-%{type}-%{+YYYY.MM.dd}

```
input {

    redis {
        data_type => "list"
        key => "logstash:redis"
        host => "127.0.0.1"
        port => 6379
        threads => 5
        codec => "json"
    }
}
filter {

}
output {

    elasticsearch {
        hosts => ["127.0.0.1:9200"]
        index => "logstash-%{type}-%{+YYYY.MM.dd}"
        document_type => "%{type}"
        workers => 1
        flush_size => 20
        idle_flush_time => 1
        template_overwrite => true
    }
    stdout{}
}
```

exec 执行脚本

```
output {
    exec {
        command => "sendsms.php \"%{message}\" -t %{user}"
    }
}
```

5. Example

https://github.com/kmtong/logback-redis-appender

5.1. Spring boot logback

例 3.1. spring boot logback

```
root@netkiller /etc/logstash/conf.d % cat spring-boot-
redis.conf
input {
 redis {
  codec => json
  host => "localhost"
  port => 6379
  key => "logstash:redis"
  data_type => "list"
 }
}

output {
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logstash-api"
  }
}
```

src/main/resources/logback.xml

```
neo@MacBook-Pro ~/deployment % cat
api.netkiller.cn/src/main/resources/logback.xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
        <include
```

```
resource="org/springframework/boot/logging/logback/defaults.xml
" />
        <include
resource="org/springframework/boot/logging/logback/file-
appender.xml" />
        <property name="type.name" value="test" />
        <appender name="LOGSTASH"
class="com.cwbase.logback.RedisAppender">
                <source>mySource</source>
                <sourcePath>mySourcePath</sourcePath>
                <type>myApplication</type>
                <tags>production</tags>
                <host>localhost</host>
                <port>6379</port>
                <database>0</database>
                <key>logstash:api</key>
        </appender>
        <appender name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
                <encoder>
                        <pattern>%date{yyyy-MM-dd HH:mm:ss}
%-4relative [%thread] %-5level %logger{35} : %msg %n</pattern>
                </encoder>
        </appender>
        <root level="INFO">
                <appender-ref ref="STDOUT" />
                <appender-ref ref="FILE" />
                <appender-ref ref="LOGSTASH" />
        </root>
</configuration>
```

5.2. 索引切割实例

例 3.2. Elasticsearch 索引切割示例

```
root@netkiller /opt/api.netkiller.cn % cat
/etc/logstash/conf.d/spring-boot-redis.conf
input {
 redis {
  codec => json
  host => "localhost"
  port => 6379
  db => 10
  key => "logstash:redis"
  data_type => "list"
 }
}

output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logstash-%{type}-%{+YYYY.MM.dd}"
  }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
        <include
resource="org/springframework/boot/logging/logback/defaults.xml
" />
        <include
resource="org/springframework/boot/logging/logback/file-
appender.xml" />
        <property name="logstash.type" value="api" />
        <property name="logstash.tags" value="springboot" />
        <appender name="LOGSTASH"
class="com.cwbase.logback.RedisAppender">
                <source>application.properties</source>
                <type>${logstash.type}</type>
                <tags>${logstash.tags}</tags>

                <host>localhost</host>
                <database>10</database>
                <key>logstash:redis</key>

                <mdc>true</mdc>
                <location>true</location>
                <callerStackIndex>0</callerStackIndex>
        </appender>
        <appender name="ASYNC"
class="ch.qos.logback.classic.AsyncAppender">
                <appender-ref ref="LOGSTASH" />
        </appender>

        <appender name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
                <encoder>
                        <pattern>%date{yyyy-MM-dd HH:mm:ss}
%-4relative [%thread] %-5level %logger{35} : %msg %n</pattern>
                </encoder>
        </appender>
        <root level="INFO">
                <appender-ref ref="STDOUT" />
                <appender-ref ref="FILE" />
                <appender-ref ref="LOGSTASH" />
        </root>
</configuration>
```

5.3.

input { file { path => ["/home/test/data.csv"] start_position => "beginning" #从什么位置读取，beginnig时导入原有数据 sincedb_path => "/test/111" type => "csv" tags => ["optical", "gather"] } } filter { if [type] == "csv" { #多个配置文件同时执行的区分 csv { columns =>["name","device_id"] separator => "^" quote_char => "‰" remove_field => ["device_id","branch_id","area_type"] } } output{ }

6. Beats

6.1. 安装 Beta

Beats 6.x 安装

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elas
tic/elastic-6.x.sh | bash
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elas
tic/beats/beats.sh | bash
```

Beats 5.x 安装

```
                                        curl -s
https://raw.githubusercontent.com/oscm/shell/master/log/beats/b
eats-5.x.sh | bash
```

6.2. Filebeat

7. FAQ

7.1. 查看 Kibana 数据库

```
# curl 'http://localhost:9200/_search?pretty'
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : ".kibana",
        "_type" : "config",
        "_id" : "5.2.2",
        "_score" : 1.0,
        "_source" : {
          "buildNum" : 14723
        }
      }
    ]
  }
}
```

7.2. logstash 无法写入 elasticsearch

elasticsearch 的配置不能省略 9200 端口，否则将无法链接 elasticsearch

```
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
```

7.3. 标准输出

```
#cd /etc/logstash/conf.d
#vim logstash_server.conf
input {
    redis {
        port => "6379"
        host => "127.0.0.1"
        data_type => "list"
        key => "logstash-redis"
        type => "redis-input"
   }
}
output {
    stdout {
        codec => rubydebug
    }
}
```

7.4. 5.x 升级至 6.x 的变化

5.x type类型如果是date，那么系统默认使用 ISO8601 格式。 6.x 修复了这个问题。"ctime": "2017-12-18 11:21:57"

第 4 章 fluentd

OS Linux/FreeBSD

Web Apache/Lighttpd/Nginx

DB MySQL/PostgreSQL

1. 收集 Docker 日志

1.1.

```
<source>
  @type forward
</source>

<match **>
  @type file
  path               /var/log/fluentd/${tag}
  append             true
  <format>
    @type            single_value
    message_key      log
  </format>
  <buffer tag,time>
    @type             file
    timekey           1d
    timekey_wait      10m
    flush_mode        interval
    flush_interval    30s
  </buffer>
</match>
```

docker-compose.yaml

```
version: '3.9'
services:
  fluentd:
    image: fluent/fluentd:latest
    container_name: fluentd
    hostname: fluentd.netkiller.cn
    restart: always
    volumes:
      -
/opt/netkiller.cn/ops.netkiller.cn/fluentd/conf:/fluentd/etc
      - /var/log/fluentd:/var/log/fluentd
    ports:
      - "24224:24224"
      - "24224:24224/udp"
    environment:
      FLUENTD_CONF: fluentd.conf
  api:
    image: openjdk:8
    container_name: api
    restart: always
    hostname: api.netkiller.cn
    extra_hosts:
      - cfca.netkiller.cn:139.196.170.132
      - raweb.netkiller.cn:139.196.170.132
      - eos.netkiller.cn:192.168.30.120
    environment:
      TZ: Asia/Shanghai
      JAVA_OPTS: -Xms1024m -Xmx4096m -XX:MetaspaceSize=128m -
XX:MaxMetaspaceSize=512m
    ports:
      - 8088:8080
    volumes:
      - /opt/netkiller.cn/api.netkiller.cn:/app
      - /opt/netkiller.cn/api.netkiller.cn/logs:/app/logs
    working_dir: /app
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: api.netkiller.cn
    entrypoint: java -jar /app/api.netkiller.cn.jar
    command:
      --spring.profiles.active=test
      --server.port=8080
```

1.2. 标准输出

```
<source>
  @type udp
  tag docker
  format json
  port 5160
</source>

<match docker>
  @type stdout
</match>
```

2. temporarily failed to flush the buffer

```
2020-10-19 03:22:24 +0000 [warn]: temporarily failed to flush the
buffer. next_retry=2020-10-19 03:22:26 +0000
error_class="Elasticsearch::Transport::Transport::Errors::NotAcceptable"
error="[406] {\"error\":\"Content-Type header [] is not
supported\",\"status\":406}" plugin_id="object:2b246e6b2084"
2020-10-19 03:22:24 +0000 [warn]: suppressed same stacktrace
```

第 5 章 监控命令

System Monitoring & Utility

1. User

1.1. last, lastb - show listing of last logged in users

```
[neo@linux ~]$ last reboot
reboot   system boot  2.6.18-164.15.1. Wed Apr 28 23:43
(6+21:31)
reboot   system boot  2.6.18-164.15.1. Fri Apr 16 04:07
(12+19:23)
reboot   system boot  2.6.18-164.15.1. Fri Apr 16 02:19
(01:46)
reboot   system boot  2.6.18-164.el5   Thu Apr 15 18:52
(07:25)

wtmp begins Thu Apr 15 18:52:15 2010
```

2. Memory

2.1. Memory

free - Display amount of free and used memory in the system

```
$ free
             total       used       free     shared    buffers
cached
Mem:       2053440     522028    1531412          0      87076
265952
-/+ buffers/cache:     169000    1884440
Swap:      2441840          0    2441840
```

5秒监控一次

```
neo@neo-OptiPlex-780:~/workspace/Document$ free -s 5
             total       used       free     shared    buffers
cached
Mem:       2054224    1708876     345348          0      58908
696404
-/+ buffers/cache:     953564    1100660
Swap:      2077692      81948    1995744

             total       used       free     shared    buffers
cached
Mem:       2054224    1708876     345348          0      58908
696404
-/+ buffers/cache:     953564    1100660
Swap:      2077692      81948    1995744

             total       used       free     shared    buffers
cached
Mem:       2054224    1709000     345224          0      58908
696404
-/+ buffers/cache:     953688    1100536
Swap:      2077692      81948    1995744
```

2.2. vmstat - Report virtual memory statistics

vmstat

```
# vmstat
procs -----------memory---------- ---swap-- -----io---- --
system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in
cs us sy id wa
 0  0      0 203668  53352 2878928    0    0     0     2    4
6  0  0 100  0
```

```
procs:
r                ;在运行队列中等待的进程数
b                ;在等待io的进程数
w                ;可以进入运行队列但被替换的进程

memoy
swap    ;现时可用的交换内存（k表示）
free    ;空闲的内存（k表示）

pages
re          回收的页面
mf          非严重错误的页面
pi          进入页面数（k表示）
po          出页面数（k表示）
fr          空余的页面数（k表示）
de          提前读入的页面中的未命中数
sr          通过时钟算法扫描的页面

disk 显示每秒的磁盘操作。 s表示scsi盘，0表示盘号

fault 显示每秒的中断数
in          设备中断
sy          系统中断
cy          cpu交换

cpu 表示cpu的使用状态
```

```
cs        用户进程使用的时间
sy        系统进程使用的时间
id        cpu空闲的时间
```

```
$ vmstat 1
procs -----------memory---------- ---swap-- -----io---- -
system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in
cs us sy id wa
 2  0      0 2692472 347884 442576    0    0     0    54   11
7 99  1  0  0
 2  0      0 2692420 347884 442600    0    0     0     0    6
87 100  0  0  0
 2  1      0 2692320 347884 442600    0    0     0  2568   26
121 100  0  0  0
 2  0      0 2687872 347884 442600    0    0     0    72   28
129 100  1  0  0
 2  0      0 2684716 347884 442600    0    0     0     0   16
91 100  0  0  0
 2  0      0 2680528 347884 442600    0    0     0     0   12
88 100  1  0  0

vmstat 参数详解

procs:
r-->在运行队列中等待的进程数
b-->在等待io的进程数
w-->可以进入运行队列但被替换的进程

memoy
swap-->现时可用的交换内存（k表示）
free-->空闲的内存（k表示）

pages
re－－》回收的页面
mf－－》非严重错误的页面
pi－－》进入页面数（k表示）
po－－》出页面数（k表示）
fr－－》空余的页面数（k表示）
de－－》提前读入的页面中的未命中数
sr－－》通过时钟算法扫描的页面

disk 显示每秒的磁盘操作。 s表示scsi盘，0表示盘号
```

```
fault 显示每秒的中断数
in－－》设备中断
sy－－》系统中断
cy－－》cpu交换

cpu 表示cpu的使用状态
cs－－》用户进程使用的时间
sy－－》系统进程使用的时间
id－－》cpu空闲的时间
```

2.3. mpstat

```
# mpstat -P ALL
Linux 2.6.18-194.el5 (cms)      08/30/2010

07:30:56 PM  CPU   %user   %nice    %sys %iowait    %irq
%soft  %steal   %idle    intr/s
07:30:56 PM  all    0.73    0.00    3.91    0.61    0.02
0.11    0.00   94.62   1380.14
07:30:56 PM    0    1.62    0.00    5.40    1.82    0.08
0.42    0.00   90.65   1375.30
07:30:56 PM    1    0.35    0.00    3.78    0.21    0.00
0.00    0.00   95.66      0.00
07:30:56 PM    2    0.44    0.00    2.74    0.22    0.00
0.00    0.00   96.59      0.00
07:30:56 PM    3    0.50    0.00    3.72    0.20    0.00
0.00    0.00   95.59      0.00
```

2.4. pmap - report memory map of a process

```
# pmap -d PID
```

```
[root@development ~]# pmap -d 3817
3817:   /sbin/mingetty tty3
Address           Kbytes Mode  Offset           Device
Mapping
```

```
0000000000400000      12 r-x-- 0000000000000000 008:00002
mingetty
0000000000602000       8 rw--- 0000000000002000 008:00002
mingetty
000000001b9f8000     132 rw--- 000000001b9f8000 000:00000   [
anon ]
0000003fd8200000     112 r-x-- 0000000000000000 008:00002 ld-
2.5.so
0000003fd841b000       4 r---- 000000000001b000 008:00002 ld-
2.5.so
0000003fd841c000       4 rw--- 000000000001c000 008:00002 ld-
2.5.so
0000003fd9200000    1332 r-x-- 0000000000000000 008:00002 libc-
2.5.so
0000003fd934d000    2048 ----- 000000000014d000 008:00002 libc-
2.5.so
0000003fd954d000      16 r---- 000000000014d000 008:00002 libc-
2.5.so
0000003fd9551000       4 rw--- 0000000000151000 008:00002 libc-
2.5.so
0000003fd9552000      20 rw--- 0000003fd9552000 000:00000   [
anon ]
00002ba6fbb68000       8 rw--- 00002ba6fbb68000 000:00000   [
anon ]
00002ba6fbb7d000       8 rw--- 00002ba6fbb7d000 000:00000   [
anon ]
00007fff2ba17000      84 rw--- 00007fffffffea000 000:00000   [
stack ]
ffffffffff600000    8192 ----- 0000000000000000 000:00000   [
anon ]
mapped: 11984K    writeable/private: 268K    shared: 0K
```

3. CPU

3.1. uptime - Tell how long the system has been running.

uptime

```
# uptime
 21:26:06 up 15 days, 58 min,  1 user,  load average: 0.85,
1.16, 2.21
```

3.2. top - display Linux tasks

5 秒监控一次

```
top -d 5
```

3.3. atop - AT Computing's System & Process Monitor

```


ATOP - ubuntu                        2013/03/12  16:09:34
------                        10s elapsed
PRC |  sys    0.03s  |  user   0.01s |  #proc    104  |  #tslpi
184  |  #tslpu     0  | #zombie    0  |  #exit      0  |
CPU |  sys       0%  |  user      0% |  irq       0%  |  idle
399%  |  wait      1%  | curf 2.13GHz  |  curscal   ?%  |
cpu |  sys       0%  |  user      0% |  irq       0%  |  idle
100%  |  cpu000 w  0%  | curf 2.13GHz  |  curscal   ?%  |
cpu |  sys       0%  |  user      0% |  irq       0%  |  idle
100%  |  cpu002 w  0%  | curf 2.13GHz  |  curscal   ?%  |
CPL |  avg1    0.00  |  avg5    0.01 |  avg15   0.05  |  csw
694  |  intr     351  |               |  numcpu     4  |
MEM |  tot     1.9G  |  free    1.4G |  cache 219.7M  |  dirty
0.0M  |  buff   93.5M  | slab   39.3M  |                |
SWP |  tot     2.0G  |  free    2.0G |                |
|                | vmcom 338.8M  |  vmlim   2.9G  |
LVM |   ubuntu-root  |  busy      1% |  read       0  |  write
9  |  MBr/s   0.00  | MBw/s   0.00  |  avio 8.44 ms  |
NET |  transport      |  tcpi      11 |  tcpo       9  |  udpi
2  |  udpo       2  | tcpao      0  |  tcppo      0  |
NET |  network        |  ipi       16 |  ipo       12  |  ipfrw
0  |  deliv     15  | icmpi      0  |  icmpo      0  |
NET |  eth0    ----  |  pcki      23 |  pcko      14  |  si
2 Kbps  |  so    2 Kbps  | erri       0  |  erro       0  |

  PID  RUID      EUID       THR   SYSCPU   USRCPU   VGROW
RGROW   RDDSK   WRDSK  ST  EXC  S  CPUNR   CPU  CMD        1/1
 5571  root      root         1    0.01s    0.00s      0K
0K       0K      0K  --    -  R      0    0%  atop
 1188  postgres  postgres     1    0.01s    0.00s      0K
0K       0K      0K  --    -  S      1    0%  postgres
 1256  redis     redis        3    0.00s    0.01s      0K
0K       0K      0K  --    -  S      2    0%  redis-server
  247  root      root         1    0.01s    0.00s      0K
0K       0K      0K  --    -  S      0    0%  kworker/0:1
 1229  ntop      ntop        11    0.00s    0.00s      0K
0K       0K      0K  --    -  S      3    0%  ntop
  920  whoopsie  whoopsie     2    0.00s    0.00s      0K
0K       0K      0K  --    -  S      3    0%  whoopsie
  914  root      root         1    0.00s    0.00s      0K
0K       0K      0K  --    -  S      1    0%  irqbalance
  265  root      root         1    0.00s    0.00s      0K
0K       0K     16K  --    -  S      1    0%  jbd2/dm-0-8
```

3.4. htop - interactive process viewer

```
```

4. Processes

4.1. strace - trace system calls and signals

```
$ strace -f -F lighttpd
```

5. lsof - list open files 文件监控

lsof - list open files

```

Command、PID 和 User 列分别表示进程的名称 进程标识符 (PID) 和所有者名称.

FD: 文件描述符,应用程序通过文件描述符识别该文件.如cwd txt等
  (1) cwd : current working directory
      应用程序的当前工作目录,这是该应用程序启动的目录,除非它本身对这个目录进行更改
  (2) txt : program text (code and data)
      该类型的文件是程序代码,如应用程序二进制文件本身或共享库,如上列表中显示的 /sbin/init 程序
  (3) lnn : library references (AIX)
      库引用
  (4) er  : FD information error (see NAME column)
      FD错误信息
  (5) jld : jail directory (FreeBSD)
      安全目录
  (6) ltx : shared library text (code and data)
      共享库文本
  (7) mxx : hex memory-mapped type number xx
      十六进制内存映射型号码xx
  (8) m86 : DOS Merge mapped file
      DOS的合并映射文件
  (9) mem : memory-mapped file
      文件内存映射
 (10) mmap : memory-mapped device
      设备内存映射
 (11) pd  : parent directory
      父目录
 (12) rtd : root directory
      root目录
 (13)  tr : kernel trace file (OpenBSD)
     内核跟踪文件
 (14) v86 : VP/ix mapped file
      VP/ix映射文件

 (15) 0 : 表示标准输出
 (16) 1 : 表示标准输入
 (17) 2 : 表示标准错误
     初始打开每个应用程序时,都具有三个文件描述符,从 0 到 2,分别表示 标准输入 标准输出 和 错误流. 正因
为如此,大多数应用程序所打开的文件的 FD 都是从3开始.
     一般在标准输出 标准错误 标准输入 后还跟着文件状态模式: r w u等
  (1) u : 表示该文件被打开并处于读取/写入模式
  (2) r : 表示该文件被打开并处于只读模式
  (3) w : 表示该文件被打开并处于
  (4) 空格 : 表示该文件的状态模式为unknow,且没有锁定
  (5) - : 表示该文件的状态模式为unknow,且被锁定
      同时在文件状态模式后面,还跟着相关的锁
  (1) N : for a Solaris NFS lock of unknown type;
  (2) r : for read lock on part of the file;
  (3) R : for a read lock on the entire file;
  (4) w : for a write lock on part of the file;
          文件的部分写锁
  (5) W : for a write lock on the entire file
          整个文件的写锁
  (6) u : for a read and write lock of any length;
  (7) U : for a lock of unknown type;

  (8) x : for an SCO OpenServer Xenix lock on part of the file;
  (9) X : for an SCO OpenServer Xenix lock on the      entire file;
  (10) space : if there is no lock.

TYPE :  文件类型,与 FD 列相比,Type 列则比较直观.
     根据具体操作系统的不同,您会发现将文件和目录称为REG 和 DIR（在 Solaris 中，称为 VREG 和
VDIR）.
     其他可能的取值为 CHR 和 BLK,分别表示字符和块设备;
     或者 UNIX、FIFO 和 IPv4,分别表示 UNIX 域套接字 先进先出 (FIFO) 队列和网际协议 (IP) 套接字.
  (1) DIR : 表示目录
  (2) CHR : 表示字符类型
  (3) BLK : 块设备类型
  (4) UNIX : UNIX 域套接字
  (5) FIFO :先进先出 (FIFO) 队列
  (6) IPv4 :网际协议 (IP) 套接字

Device  SIZE/OFF Node 和 NA
      列涉及到文件本身的信息,分别表示
          指定磁盘的名称
          文件的大小
          索引节点(文件在磁盘上的标识)
          该文件的确切名称
```

```
$ sudo lsof -c lighttpd
```

5.1. $$

```
neo@netkiller:~/workspace/Document$ lsof -p $$
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF   NODE NAME
zsh     4536  neo  cwd    DIR    8,6     4096     30 /home/neo/workspace/Document
zsh     4536  neo  rtd    DIR    8,1     4096      2 /
zsh     4536  neo  txt    REG    8,1   675792   6907 /bin/zsh4
zsh     4536  neo  mem    REG    8,1    68824  56594 /usr/lib/zsh/4.3.10/zsh/computil.so
zsh     4536  neo  mem    REG    8,1    41000  30570
/usr/lib/zsh/4.3.10/zsh/parameter.so
zsh     4536  neo  mem    REG    8,1    31512  53350 /usr/lib/zsh/4.3.10/zsh/zutil.so
zsh     4536  neo  mem    REG    8,1   153096  53354 /usr/lib/zsh/4.3.10/zsh/complete.so
zsh     4536  neo  mem    REG    8,1   290888  56596 /usr/lib/zsh/4.3.10/zsh/zle.so
zsh     4536  neo  mem    REG    8,1    10544  30579 /usr/lib/zsh/4.3.10/zsh/terminfo.so
zsh     4536  neo  mem    REG    8,1    51712  19594 /lib/libnss_files-2.11.1.so
zsh     4536  neo  mem    REG    8,1    43552  23798 /lib/libnss_nis-2.11.1.so
zsh     4536  neo  mem    REG    8,1    97256  15503 /lib/libnsl-2.11.1.so
zsh     4536  neo  mem    REG    8,1    35712  16431 /lib/libnss_compat-2.11.1.so
zsh     4536  neo  mem    REG    8,1    18704   1902 /lib/libattr.so.1.1.0
zsh     4536  neo  mem    REG    8,1  1568136   7583 /lib/libc-2.11.1.so
zsh     4536  neo  mem    REG    8,1   534832  11379 /lib/libm-2.11.1.so
zsh     4536  neo  mem    REG    8,1   323640   7295 /lib/libncursesw.so.5.7
zsh     4536  neo  mem    REG    8,1    14696  11378 /lib/libdl-2.11.1.so
zsh     4536  neo  mem    REG    8,1    18888   5099 /lib/libcap.so.2.17
zsh     4536  neo  mem    REG    8,1   136936   7487 /lib/ld-2.11.1.so
zsh     4536  neo  mem    REG    8,1   256324 145156 /usr/lib/locale/en_US.utf8/LC_CTYPE
zsh     4536  neo  mem    REG    8,1       54 131099
/usr/lib/locale/en_US.utf8/LC_NUMERIC
zsh     4536  neo  mem    REG    8,1     2454 145158 /usr/lib/locale/en_US.utf8/LC_TIME
zsh     4536  neo  mem    REG    8,1  1170770 145157
/usr/lib/locale/en_US.utf8/LC_COLLATE
```

```
zsh     4536  neo  mem    REG    8,1      286 145159
/usr/lib/locale/en_US.utf8/LC_MONETARY
zsh     4536  neo  mem    REG    8,1       57 145160
/usr/lib/locale/en_US.utf8/LC_MESSAGES/SYS_LC_MESSAGES
zsh     4536  neo  mem    REG    8,1    26048  73711 /usr/lib/gconv/gconv-modules.cache
zsh     4536  neo  mem    REG    8,1       34 131105 /usr/lib/locale/en_US.utf8/LC_PAPER
zsh     4536  neo  mem    REG    8,1       77 131106 /usr/lib/locale/en_US.utf8/LC_NAME
zsh     4536  neo  mem    REG    8,1      155 145161
/usr/lib/locale/en_US.utf8/LC_ADDRESS
zsh     4536  neo  mem    REG    8,1       59 145162
/usr/lib/locale/en_US.utf8/LC_TELEPHONE
zsh     4536  neo  mem    REG    8,1       23 131109
/usr/lib/locale/en_US.utf8/LC_MEASUREMENT
zsh     4536  neo  mem    REG    8,1      373 145163
/usr/lib/locale/en_US.utf8/LC_IDENTIFICATION
zsh     4536  neo    0u   CHR  136,0      0t0      3 /dev/pts/0
zsh     4536  neo    1u   CHR  136,0      0t0      3 /dev/pts/0
zsh     4536  neo    2u   CHR  136,0      0t0      3 /dev/pts/0
zsh     4536  neo   10u   CHR  136,0      0t0      3 /dev/pts/0
```

5.2. 监控文件系统

谁打开了该文件? 显示打开文件filename的进程

```
lsof filename
```

列出某个目录下被打开的文件

```
# lsof /tmp/
COMMAND     PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
seahorse- 4158  neo  cwd    DIR    8,2    53248 1310721 /tmp
```

递归子目录列出文件状态

```
$ sudo lsof +D /srv/
COMMAND  PID USER  FD   TYPE DEVICE SIZE/OFF    NODE NAME
match   5227 root txt    REG  252,0  1351616 1966083 /srv/match

[root@netkiller ~]# lsof +D /proc/1/
COMMAND PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd   1 root    9r   REG    0,3        0 8401 /proc/1/mountinfo
```

```
>1 查看某个文件被哪个进程/命令正在使用


在一个窗口执行
[root@netkiller ~]# less /etc/passwd
在另外一个窗口执行
[root@netkiller ~]# lsof /etc/passwd
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
```

```
less     14493 root    4r   REG    8,2      2676 4466070 /etc/passwd

递归查看某个目录中文件被哪些命令/程序使用
    使用了+D，对应目录下的所有子目录和文件都会被列出
开两个窗口分别执行如下命令
[root@netkiller ~]# less test/logs/access/2013-05-22.access
[root@netkiller ~]# less test/11
再第三个窗口执行
[root@netkiller ~]# lsof +D test/
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
less     14840 root    4r   REG    8,2       252 6166856 test/11
less     14877 root    4r   REG    8,2         0 6166852 test/logs/access/2013-05-22.access
```

5.3. 设备文件

```
$ lsof /dev/tty1
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
bash     17187  neo    0u   CHR    4,1      0t0 1057 /dev/tty1
bash     17187  neo    1u   CHR    4,1      0t0 1057 /dev/tty1
bash     17187  neo    2u   CHR    4,1      0t0 1057 /dev/tty1
bash     17187  neo  255u   CHR    4,1      0t0 1057 /dev/tty1
```

5.4. 用户监控

用户显示打开的文件

```
# lsof -u apache |more
COMMAND   PID   USER   FD    TYPE DEVICE SIZE/OFF   NODE NAME
httpd    4374 apache  cwd     DIR  252,1     4096      2 /
httpd    4374 apache  rtd     DIR  252,1     4096      2 /
httpd    4374 apache  txt     REG  252,1   354816 408099 /usr/sbin/httpd
httpd    4374 apache  mem     REG  252,1     9488 408013 /usr/lib64/apr-util-1/apr_ldap-1.so
httpd    4374 apache  mem     REG  252,1    27424    907 /lib64/libnss_dns-2.12.so
httpd    4374 apache  mem     REG  252,1    65928    909 /lib64/libnss_files-2.12.so
httpd    4374 apache  mem     REG  252,1    10416 408095 
/usr/lib64/httpd/modules/mod_version.so
httpd    4374 apache  mem     REG  252,1    27312 408054 
/usr/lib64/httpd/modules/mod_cgi.so
httpd    4374 apache  mem     REG  252,1    22992 408061 
/usr/lib64/httpd/modules/mod_disk_cache.so

[root@netkiller ~]# lsof -u www
COMMAND   PID USER   FD   TYPE             DEVICE SIZE/OFF     NODE NAME
httpd    2412  www  DEL    REG                0,4             12653 /dev/zero
httpd    2412  www  mem    REG                8,2    90784  5636110 /lib64/libgcc_s-4.4.7-20120601.so.1
```

列出被打开的文件信息,排除root用户

```
[root@netkiller neo]# lsof -u ^root |more

COMMAND     PID   TID          USER   FD      TYPE             DEVICE   SIZE/OFF
NODE NAME
dbus-daem   448                dbus  cwd       DIR              253,1       4096
2 /
dbus-daem   448                dbus  rtd       DIR              253,1       4096
2 /
dbus-daem   448                dbus  txt       REG              253,1     441256
141406 /usr/bin/dbus-daemon;56822cb8 (deleted)
dbus-daem   448                dbus  DEL       REG              253,1
146439 /usr/lib64/libnss_sss.so.2;56822cb8
dbus-daem   448                dbus  DEL       REG              253,1
151203 /usr/lib64/libnss_files-2.17.so;56822cb8
dbus-daem   448                dbus  DEL       REG              253,1
151199 /usr/lib64/libdl-2.17.so;56822cb8
dbus-daem   448                dbus  DEL       REG              253,1
133002 /usr/lib64/liblzma.so.5.0.99;56822ac0
dbus-daem   448                dbus  DEL       REG              253,1
133005 /usr/lib64/libpcre.so.1.2.0;56822ac0
dbus-daem   448                dbus  DEL       REG              253,1
132825 /usr/lib64/libc-2.17.so;56822cb8
dbus-daem   448                dbus  DEL       REG              253,1
151206 /usr/lib64/librt-2.17.so;56822cb8
dbus-daem   448                dbus  DEL       REG              253,1
132851 /usr/lib64/libpthread-2.17.so;56822cb8
dbus-daem   448                dbus  DEL       REG              253,1
133622 /usr/lib64/libcap-ng.so.0.0.0;56822cb8
dbus-daem   448                dbus  mem       REG              253,1     118792
133084 /usr/lib64/libaudit.so.1.0.0
dbus-daem   448                dbus  mem       REG              253,1     147120
133015 /usr/lib64/libselinux.so.1
dbus-daem   448                dbus  mem       REG              253,1     173288
133153 /usr/lib64/libexpat.so.1.6.0
dbus-daem   448                dbus  DEL       REG              253,1
132818 /usr/lib64/ld-2.17.so;56822cb8
dbus-daem   448                dbus    0r      CHR                1,3        0t0
1028 /dev/null
dbus-daem   448                dbus    1u     unix 0xffff880426d4c740        0t0
14381 socket
dbus-daem   448                dbus    2u     unix 0xffff880426d4c740        0t0
14381 socket
dbus-daem   448                dbus    3u     unix 0xffff880428cd7800        0t0
14082 /var/run/dbus/system_bus_socket
dbus-daem   448                dbus    4u  a_inode                0,9          0
5639 [eventpoll]
dbus-daem   448                dbus    5r  a_inode                0,9          0
5639 inotify
dbus-daem   448                dbus    6u     sock                0,6        0t0
14179 protocol: NETLINK
dbus-daem   448                dbus    7u     unix 0xffff880428cd1e00        0t0
14180 socket
dbus-daem   448                dbus    8u     unix 0xffff880428cd5640        0t0
14181 socket
dbus-daem   448                dbus    9u     unix 0xffff880037101e00        0t0
5347943 /var/run/dbus/system_bus_socket
dbus-daem   448                dbus   10u     unix 0xffff8800292ae900        0t0
626418112 /var/run/dbus/system_bus_socket
dbus-daem   448                dbus   11u     unix 0xffff880426f3cec0        0t0
5345962 socket
dbus-daem   448                dbus   12u     unix 0xffff8801f8149e00        0t0
```

```
626420423 /var/run/dbus/system_bus_socket


[root@netkiller ~]# lsof -u ^www
COMMAND     PID      USER   FD      TYPE             DEVICE  SIZE/OFF       NODE NAME
init          1      root  txt       REG                8,2    150352    2228260
/sbin/init
init          1      root  mem       REG                8,2     65928    5636192
/lib64/libnss_files-2.12.so
```

组监控

```
[root@netkiller neo]# lsof -g 0
COMMAND     PID PGID USER   FD      TYPE DEVICE SIZE/OFF NODE NAME
kthreadd      2    0 root  cwd       DIR  202,1     4096    2 /
kthreadd      2    0 root  rtd       DIR  202,1     4096    2 /
kthreadd      2    0 root  txt   unknown                      /proc/2/exe
ksoftirqd     3    0 root  cwd       DIR  202,1     4096    2 /
ksoftirqd     3    0 root  rtd       DIR  202,1     4096    2 /
ksoftirqd     3    0 root  txt   unknown                      /proc/3/exe
kworker/0     5    0 root  cwd       DIR  202,1     4096    2 /
kworker/0     5    0 root  rtd       DIR  202,1     4096    2 /
kworker/0     5    0 root  txt   unknown                      /proc/5/exe
migration     7    0 root  cwd       DIR  202,1     4096    2 /
migration     7    0 root  rtd       DIR  202,1     4096    2 /
migration     7    0 root  txt   unknown                      /proc/7/exe
```

5.5. 监控进程

列出某个程序进程所打开的文件信息,显示httpd进程现在打开的文件

```
lsof -c httpd
```

显示多个进程命令用法

```
[root@netkiller ~]# lsof -c smbd
COMMAND  PID USER   FD   TYPE             DEVICE SIZE/OFF    NODE NAME
smbd    2506 root  cwd    DIR                8,2     4096       2 /
smbd    2506 root  rtd    DIR                8,2     4096       2 /
smbd    2506 root  txt    REG                8,2 10112200 3935771 /usr/sbin/smbd

[root@netkiller ~]# lsof -c smbd -c httpd
```

-p 进程ID, 显示该进程打开了那些文件

```
pgrep httpd
lsof -p 1782
```

显示进程ID

```
# lsof -t -u apache
4374
4375
4376
4377
4378
4379
4380

列出某个程序号打开的文件

[root@netkiller ~]# lsof -p 2374
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
httpd   2374 root  cwd    DIR    8,2     4096       2 /
httpd   2374 root  rtd    DIR    8,2     4096       2 /
httpd   2374 root  txt    REG    8,2  1772950 4985314 /usr/local/apache/bin/httpd
httpd   2374 root  DEL    REG    0,4            12653 /dev/zero
httpd   2374 root  mem    REG    8,2    90784 5636110 /lib64/libgcc_s-4.4.7-
20120601.so.1
```

监控多个进程ID

```
[root@netkiller neo]# lsof -p 20535,26359,31462 | more
COMMAND   PID    USER   FD   TYPE             DEVICE   SIZE/OFF      NODE NAME
nginx   20535     root cwd    DIR              253,1       4096          2 /
nginx   20535     root rtd    DIR              253,1       4096          2 /
nginx   20535     root txt    REG              253,1    1066704    142069
/usr/sbin/nginx
nginx   20535     root DEL    REG                0,4             686393039 /dev/zero
nginx   20535     root mem    REG              253,1      61928    162109
/usr/lib64/libnss_files-2.17.so
nginx   20535     root mem    REG              253,1     153192    151546
/usr/lib64/liblzma.so.5.0.99
nginx   20535     root mem    REG              253,1     147120    133015
/usr/lib64/libselinux.so.1
nginx   20535     root mem    REG              253,1     110808    162113
/usr/lib64/libresolv-2.17.so
nginx   20535     root mem    REG              253,1      15688    134676
/usr/lib64/libkeyutils.so.1.5
nginx   20535     root mem    REG              253,1      62720    158030
/usr/lib64/libkrb5support.so.0.1
nginx   20535     root mem    REG              253,1     202576    137049
/usr/lib64/libk5crypto.so.3.1
nginx   20535     root mem    REG              253,1      15840    133029
/usr/lib64/libcom_err.so.2.1
nginx   20535     root mem    REG              253,1     950496    137059
/usr/lib64/libkrb5.so.3.3
nginx   20535     root mem    REG              253,1     316528    151679
/usr/lib64/libgssapi_krb5.so.2.2
nginx   20535     root mem    REG              253,1      11376    151527
/usr/lib64/libfreebl3.so
nginx   20535     root mem    REG              253,1    2112384    132823
/usr/lib64/libc-2.17.so
nginx   20535     root mem    REG              253,1      90632    133017
/usr/lib64/libz.so.1.2.7
nginx   20535     root mem    REG              253,1    2016880    132882
/usr/lib64/libcrypto.so.1.0.1e
nginx   20535     root mem    REG              253,1     449904    137215
/usr/lib64/libssl.so.1.0.1e
```

```
nginx   20535    root  mem    REG              253,1     398264    160788
/usr/lib64/libpcre.so.1.2.0
nginx   20535    root  mem    REG              253,1      40816    151198
/usr/lib64/libcrypt-2.17.so
nginx   20535    root  mem    REG              253,1     142304    132849
/usr/lib64/libpthread-2.17.so
nginx   20535    root  mem    REG              253,1      19520    162101
/usr/lib64/libdl-2.17.so
nginx   20535    root  mem    REG              253,1     164440    132816 /usr/lib64/ld-
2.17.so
nginx   20535    root  DEL    REG                0,4            686393042 /dev/zero
nginx   20535    root    0u   CHR                1,3        0t0      1028 /dev/null
nginx   20535    root    1u   CHR                1,3        0t0      1028 /dev/null
```

排除1，4，显示2，3，5

```
[root@netkiller neo]# lsof -p ^1,2,3,^4,5
COMMAND    PID USER   FD      TYPE DEVICE SIZE/OFF NODE NAME
kthreadd     2 root  cwd       DIR  253,1     4096    2 /
kthreadd     2 root  rtd       DIR  253,1     4096    2 /
kthreadd     2 root  txt   unknown                      /proc/2/exe
ksoftirqd    3 root  cwd       DIR  253,1     4096    2 /
ksoftirqd    3 root  rtd       DIR  253,1     4096    2 /
ksoftirqd    3 root  txt   unknown                      /proc/3/exe
kworker/0    5 root  cwd       DIR  253,1     4096    2 /
kworker/0    5 root  rtd       DIR  253,1     4096    2 /
kworker/0    5 root  txt   unknown                      /proc/5/exe
```

5.6. 监控网络

列出所有的网络连接

```
[root@netkiller neo]# lsof -i
COMMAND      PID          USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
php-fpm     2274           www    0u  IPv4  96056019      0t0  TCP localhost:cslistener
(LISTEN)
php-fpm     2274           www    4u  IPv4 688391009      0t0  TCP localhost:43483-
>localhost:27017 (ESTABLISHED)
python3     4384        zabbix    6u  IPv4 688769849      0t0  TCP iZ623qr3xctZ:zabbix-
agent->10.26.6.18:50666 (ESTABLISHED)
python3     4385        zabbix    6u  IPv4 688769848      0t0  TCP iZ623qr3xctZ:zabbix-
agent->10.26.6.18:50668 (ESTABLISHED)
redis-ser   5170         redis    4u  IPv4   5690059      0t0  TCP localhost:6379
(LISTEN)
php-fpm     8277           www    0u  IPv4  96056019      0t0  TCP localhost:cslistener
(LISTEN)
php-fpm     8277           www    4u  IPv4 688149893      0t0  TCP localhost:60933-
>localhost:27017 (ESTABLISHED)
php-fpm     8543           www    0u  IPv4  96056019      0t0  TCP localhost:cslistener
(LISTEN)
beam.smp    9703      rabbitmq    8u  IPv4 626401894      0t0  TCP *:25672 (LISTEN)
beam.smp    9703      rabbitmq    9u  IPv4 626401896      0t0  TCP localhost:42821-
>localhost:epmd (ESTABLISHED)
beam.smp    9703      rabbitmq   17u  IPv6 626403609      0t0  TCP *:amqp (LISTEN)
beam.smp    9703      rabbitmq   18u  IPv4 626402643      0t0  TCP *:15672 (LISTEN)
beam.smp   9703      rabbitmq   20u  IPv6 685257290      0t0  TCP localhost:amqp-
>localhost:57692 (ESTABLISHED)
sshd      11227          root    3u  IPv4 626404210      0t0  TCP *:ssh (LISTEN)
ntpd      11646           ntp   16u  IPv4 626409506      0t0  UDP *:ntp
ntpd      11646           ntp   17u  IPv6 626406239      0t0  UDP *:ntp
ntpd      11646           ntp   18u  IPv4 626406244      0t0  UDP localhost:ntp
ntpd      11646           ntp   19u  IPv4 626406245      0t0  UDP iZ623qr3xctZ:ntp
ntpd      11646           ntp   20u  IPv4 626406246      0t0  UDP iZ623qr3xctZ:ntp


5 列出所有的网络连接/端口


[root@netkiller ~]# lsof -i
COMMAND     PID   USER    FD   TYPE   DEVICE SIZE/OFF NODE NAME
portreser  1698   root    5u  IPv4    10656      0t0  UDP *:ldaps
snmpd      1993   root    7u  IPv4    12071      0t0  UDP *:snmp
snmpd      1993   root    9u  IPv4    12073      0t0  TCP localhost:smux (LISTEN)
sshd       2005   root    3u  IPv4    12109      0t0  TCP *:ssh (LISTEN)
```

什么程序运行在22端口上

```
lsof -i :22
```

谁在联系端口

```
# lsof -i -a -c ssh
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd    2843 root    3r  IPv4  27960      0t0  TCP 192.168.6.9:ssh->192.168.6.30:55363
(ESTABLISHED)
sshd    3003 root    3u  IPv4  28864      0t0  TCP *:ssh (LISTEN)
sshd    3003 root    4u  IPv6  28866      0t0  TCP *:ssh (LISTEN)
```

```
$ lsof -i -a -c nginx
COMMAND    PID USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
nginx   26222  www    8w  IPv4 557827648      0t0  TCP 42.121.14.230:http-
>110.240.206.67:63482 (ESTABLISHED)
nginx   26222  www    9u  IPv4 557817283      0t0  TCP 42.121.14.230:http-
>27.106.154.202:18972 (ESTABLISHED)
nginx   26222  www   10u  IPv4 496452301      0t0  TCP *:http (LISTEN)
nginx   26222  www   17u  IPv4 557826020      0t0  TCP 42.121.14.230:http-
>210.177.78.33:62297 (ESTABLISHED)
nginx   26222  www   18u  IPv4 557827745      0t0  TCP 42.121.14.230:http-
>115.214.39.230:50628 (ESTABLISHED)
nginx   26222  www   19u  IPv4 557826475      0t0  TCP 42.121.14.230:http-
>183.160.124.225:57143 (ESTABLISHED)
nginx   26222  www   20u  IPv4 557827670      0t0  TCP 42.121.14.230:http-
>125.88.77.30:8956 (ESTABLISHED)
nginx   26222  www   21u  IPv4 557826122      0t0  TCP 42.121.14.230:http-
>116.24.229.173:rfid-rp1 (ESTABLISHED)
nginx   26222  www   22u  IPv4 557826127      0t0  TCP 42.121.14.230:http-
>119.137.141.76:21508 (ESTABLISHED)
nginx   26222  www   23u  IPv4 557826476      0t0  TCP 42.121.14.230:http-
>183.160.124.225:57144 (ESTABLISHED)
```

```
nginx   26222  www   24u  IPv4 557821930      0t0  TCP 42.121.14.230:http-
>210.21.127.136:52309 (ESTABLISHED)
nginx   26222  www   25u  IPv4 557826477      0t0  TCP 42.121.14.230:http-
>183.160.124.225:57145 (ESTABLISHED)
nginx   26222  www   26u  IPv4 557827693      0t0  TCP 42.121.14.230:http-
>111.227.215.135:18628 (ESTABLISHED)
```

通过进程ID监控网络连接

```
$ lsof -i -a -p 26222
COMMAND   PID USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
nginx   26222  www    8w  IPv4 557827648      0t0  TCP 42.121.14.230:http-
>110.240.206.67:63482 (ESTABLISHED)
nginx   26222  www    9u  IPv4 557817283      0t0  TCP 42.121.14.230:http-
>27.106.154.202:18972 (ESTABLISHED)
nginx   26222  www   10u  IPv4 496452301      0t0  TCP *:http (LISTEN)
nginx   26222  www   21u  IPv4 557826122      0t0  TCP 42.121.14.230:http-
>116.24.229.173:rfid-rp1 (ESTABLISHED)
nginx   26222  www   26u  IPv4 557827693      0t0  TCP 42.121.14.230:http-
>111.227.215.135:18628 (ESTABLISHED)
nginx   26222  www   31u  IPv4 557798349      0t0  TCP 42.121.14.230:http-
>213.92.156.27.broad.fz.fj.dynamic.163data.com.cn:novation (ESTABLISHED)
nginx   26222  www   33u  IPv4 557807306      0t0  TCP 42.121.14.230:http-
>182.139.49.102:news (ESTABLISHED)
nginx   26222  www   38u  IPv4 557825270      0t0  TCP 42.121.14.230:http-
>122.71.50.188:43694 (ESTABLISHED)
nginx   26222  www   40u  IPv4 557817907      0t0  TCP 42.121.14.230:http-
>120.28.127.54:62009 (ESTABLISHED)
nginx   26222  www   41u  IPv4 557800691      0t0  TCP 42.121.14.230:http-
>27.190.185.75:60475 (ESTABLISHED)
```

UDP 监控

```
# lsof -i udp;
COMMAND    PID     USER   FD   TYPE   DEVICE SIZE/OFF NODE NAME
rpcbind   2431      rpc    6u  IPv4    12483      0t0  UDP *:sunrpc
rpcbind   2431      rpc    7u  IPv4    12487      0t0  UDP *:kink
rpcbind   2431      rpc    9u  IPv6    12490      0t0  UDP *:sunrpc
rpcbind   2431      rpc   10u  IPv6    12492      0t0  UDP *:kink
avahi-dae 2549    avahi   13u  IPv4    12781      0t0  UDP *:mdns
avahi-dae 2549    avahi   14u  IPv4    12782      0t0  UDP *:45747
rpc.statd 2570  rpcuser    5u  IPv4    13011      0t0  UDP *:asia
rpc.statd 2570  rpcuser    8u  IPv4    13015      0t0  UDP *:55218
rpc.statd 2570  rpcuser   10u  IPv6    13023      0t0  UDP *:51236
openvpn   2594   nobody    5u  IPv4    13060      0t0  UDP *:openvpn
cupsd     2661     root    9u  IPv4    13379      0t0  UDP *:ipp
ntpd      2832      ntp   16u  IPv4    14050      0t0  UDP *:ntp
ntpd      2832      ntp   17u  IPv6    14051      0t0  UDP *:ntp
ntpd      2832      ntp   18u  IPv6    14055      0t0  UDP localhost:ntp
ntpd      2832      ntp   19u  IPv6    14056      0t0  UDP
[fe80::225:90ff:fe35:906c]:ntp
ntpd      2832      ntp   20u  IPv4    14057      0t0  UDP localhost:ntp
ntpd      2832      ntp   21u  IPv4    14058      0t0  UDP manager.repo:ntp
ntpd      2832      ntp   22u  IPv4    14059      0t0  UDP 10.8.0.1:ntp
ntpd      2832      ntp   24u  IPv4    15922      0t0  UDP 192.168.122.1:ntp
ntpd      2832      ntp   25u  IPv6    27224      0t0  UDP [fe80::fc54:ff:fe94:b3c2]:ntp
ntpd      2832      ntp   26u  IPv6    27225      0t0  UDP [fe80::fc54:ff:fe54:c9d2]:ntp
ntpd       2832       ntp   27u  IPv6    27948       0t0  UDP [fe80::fc54:ff:fe4e:a846]:ntp
ntpd       2832       ntp   28u  IPv6    28197       0t0  UDP [fe80::fc54:ff:fe19:c00e]:ntp
ntpd       2832       ntp   29u  IPv6 99178415       0t0  UDP [fe80::fc54:ff:fe5a:ace]:ntp
ntpd       2832       ntp   30u  IPv6 99179648       0t0  UDP [fe80::fc54:ff:fe68:54a0]:ntp
ntpd       2832       ntp   31u  IPv6 99180801       0t0  UDP [fe80::fc54:ff:fed6:3593]:ntp
postmaste 3391 postgres    9u  IPv6    15004       0t0  UDP localhost:56631-
>localhost:56631
postmaste 3395 postgres    9u  IPv6    15004       0t0  UDP localhost:56631-
>localhost:56631
postmaste 3396 postgres    9u  IPv6    15004       0t0  UDP localhost:56631-
>localhost:56631
postmaste 3397 postgres    9u  IPv6    15004       0t0  UDP localhost:56631-
>localhost:56631
postmaste 3398 postgres    9u  IPv6    15004       0t0  UDP localhost:56631-
>localhost:56631
postmaste 3399 postgres    9u  IPv6    15004       0t0  UDP localhost:56631-
>localhost:56631
dnsmasq   3647   nobody    5u  IPv4    15671       0t0  UDP *:bootps
dnsmasq   3647   nobody    7u  IPv4    15680       0t0  UDP 192.168.122.1:domain
```

TCP 监控

```
lsof -i tcp;
```

特定的tcp/udp端口, 监控 udp 端口 123

```
[root@netkiller neo]# lsof -i udp:123
COMMAND   PID USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
ntpd    11646  ntp   16u  IPv4 626409506      0t0  UDP *:ntp
ntpd    11646  ntp   17u  IPv6 626406239      0t0  UDP *:ntp
ntpd    11646  ntp   18u  IPv4 626406244      0t0  UDP localhost:ntp
ntpd    11646  ntp   19u  IPv4 626406245      0t0  UDP iZ623qr3xctZ:ntp
ntpd    11646  ntp   20u  IPv4 626406246      0t0  UDP iZ623qr3xctZ:ntp

检测某个端口所占用的进程，如22端口
[root@netkiller ~]# lsof -i :22

[root@netkiller ~]# lsof -i udp:53
```

列出所有tcp/UDP 网络连接信息

```
[root@netkiller ~]# lsof -i tcp/udp
```

列出nginx用户活跃的链接

```
[root@netkiller neo]# lsof  -a -u nginx -i
COMMAND   PID   USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
nginx   20536 nginx   19u  IPv4 686393040      0t0  TCP *:http (LISTEN)
nginx   20536 nginx   20u  IPv4 686393041      0t0  TCP *:https (LISTEN)
nginx   20536 nginx   42u  IPv4 688774445      0t0  TCP iZ623qr3xctZ:http-
>112.224.19.79:32751 (ESTABLISHED)
nginx   20536 nginx   49u  IPv4 688774400      0t0  TCP iZ623qr3xctZ:http-
```

```
>117.156.4.113:58212 (ESTABLISHED)
nginx   20536 nginx   52u  IPv4 688774494      0t0  TCP iZ623qr3xctZ:http-
>112.224.19.79:32753 (ESTABLISHED)
nginx   20536 nginx   53u  IPv4 688774495      0t0  TCP iZ623qr3xctZ:http-
>112.224.19.79:32752 (ESTABLISHED)
nginx   20536 nginx   54u  IPv4 688774555      0t0  TCP iZ623qr3xctZ:http-
>113.128.232.89:37529 (ESTABLISHED)
nginx   20536 nginx   55u  IPv4 688774497      0t0  TCP iZ623qr3xctZ:http-
>112.224.19.79:32754 (ESTABLISHED)
nginx   20536 nginx   56u  IPv4 688774556      0t0  TCP iZ623qr3xctZ:http-
>113.128.232.89:37530 (ESTABLISHED)
nginx   20536 nginx   58u  IPv4 688774500      0t0  TCP iZ623qr3xctZ:http-
>112.224.19.79:32755 (ESTABLISHED)
nginx   20536 nginx   60u  IPv4 688778242      0t0  TCP iZ623qr3xctZ:http-
>113.128.232.89:37532 (ESTABLISHED)
nginx   20536 nginx   61u  IPv4 688774559      0t0  TCP iZ623qr3xctZ:http-
>113.128.232.89:37528 (ESTABLISHED)
nginx   20536 nginx   64u  IPv4 688774562      0t0  TCP iZ623qr3xctZ:http-
>113.128.232.89:37531 (ESTABLISHED)
nginx   20537 nginx   19u  IPv4 686393040      0t0  TCP *:http (LISTEN)
nginx   20537 nginx   20u  IPv4 686393041      0t0  TCP *:https (LISTEN)
nginx   20538 nginx   19u  IPv4 686393040      0t0  TCP *:http (LISTEN)
nginx   20538 nginx   20u  IPv4 686393041      0t0  TCP *:https (LISTEN)
nginx   20539 nginx   18u  IPv4 688777804      0t0  TCP iZ623qr3xctZ:http-
>39.187.213.246:49624 (ESTABLISHED)
nginx   20539 nginx   19u  IPv4 686393040      0t0  TCP *:http (LISTEN)
nginx   20539 nginx   20u  IPv4 686393041      0t0  TCP *:https (LISTEN)
```

5.7. lsof 高级用法

组合参数

```
# lsof -a -c bash -u root
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
bash    1394 root  cwd    DIR    8,2     4096 4849665 /root
bash    1394 root  rtd    DIR    8,2     4096       2 /
bash    1394 root  txt    REG    8,2   938768 3671557 /bin/bash
bash    1394 root  mem    REG    8,2   156872 3014902 /lib64/ld-2.12.so
bash    1394 root  mem    REG    8,2  1922152 3014903 /lib64/libc-2.12.so
bash    1394 root  mem    REG    8,2    22536 3014911 /lib64/libdl-2.12.so
bash    1394 root  mem    REG    8,2   138280 3018719 /lib64/libtinfo.so.5.7
bash    1394 root  mem    REG    8,2    65928 3017998 /lib64/libnss_files-2.12.so
bash    1394 root  mem    REG    8,2    26060 2632051 /usr/lib64/gconv/gconv-
modules.cache
bash    1394 root  mem    REG    8,2 99158576 2648204 /usr/lib/locale/locale-archive
bash    1394 root    0u   CHR  136,7      0t0      10 /dev/pts/7
bash    1394 root    1u   CHR  136,7      0t0      10 /dev/pts/7
bash    1394 root    2u   CHR  136,7      0t0      10 /dev/pts/7
bash    1394 root  255u   CHR  136,7      0t0      10 /dev/pts/7
```

每个5秒刷新一次

```
# lsof -c init -a -r5
```

```
列出www用户的所有活跃的网络端口
```

```
[root@netkiller ~]# lsof -a -u www -i

列出被sshd进程所打开的所有IPV4网络相关文件
[root@netkiller ~]# lsof -i 4 -c sshd  -a

列出被root用户所打开的所有TCP和IPV4网络相关文件
[root@netkiller ~]# lsof -i 4 -i tcp  -u root -a
```

5.8. 根据文件描述列出对应的文件信息

```

lsof -d  fd_type

[root@netkiller ~]# lsof -d 2
COMMAND     PID      USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
init          1      root    2u   CHR    1,3      0t0    3794 /dev/null

根据文件描述范围列出文件信息
[root@netkiller ~]# lsof -d 2-4
COMMAND     PID      USER   FD   TYPE             DEVICE  SIZE/OFF       NODE NAME
init          1      root    2u   CHR                1,3       0t0       3794 /dev/null

列出COMMAND列中包含字符串" httpd"，且文件描符的类型为txt的文件信息
[root@netkiller ~]# lsof -c httpd -a -d txt
COMMAND  PID USER  FD   TYPE DEVICE SIZE/OFF    NODE NAME
httpd   2374 root txt    REG    8,2  1772950 4985314 /usr/local/apache/bin/httpd
```

6. Harddisk IO

6.1. input/output statistics

```
$ sudo apt-get install sysstat
```

iostat

```
$ iostat
Linux 2.6.24-21-generic (netkiller)     Thursday, December 04,
2008

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.57    0.03    0.14    0.41    0.00   98.85

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read
Blk_wrtn
sda               6.45       132.69        68.33     595116
306456
sda1              0.00         0.00         0.00       1606
58
sda2              0.00         0.00         0.00        820
0
sda3              2.20         1.16        17.27    1502618
22448752
```

sudo iostat -x 2

```
# iostat -x 1
avg-cpu: %user %nice %sys %idle
2.04 0.00 97.96 0.00
Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s rkB/s wkB/s avgrq-
sz avgqu-sz await svctm %util
/dev/sda 0.00 633.67 3.06 102.31 24.49 5281.63 12.24 2640.82
288.89 73.67 113.89 27.22 50.00
```

```
从输出我们看到w/s=102,wKB/s=2640.所以2640/102=23KB per I/O.

因此对于连续I/O系统来说我们要关注系统读取大量数据的能力即KB per request.
对于随机I/O系统我们注重IOPS值.
```

5 秒监控一次

```
iostat -d 5
```

6.2. iotop - simple top-like I/O monitor

```
# yum install iotop
```

```
$ sudo apt-get install iotop
```

6.3. ionice - set or get process I/O scheduling class and priority

6.4. smartd - SMART Disk Monitoring Daemon

配置表示smartd以静默状态工作，当SMART中报告PASSED的时候不理睬一旦出现Failure，立刻用邮件通知用户指定的邮箱

```
vi /etc/smartd.conf
/dev/sdb -H -m neo@domain.com
```

修改配置后重启服务:

```
/etc/init.d/smartd start
```

7. Network IO

7.1. netstat

netstat 监控TCP状态

```
#netstat -n | awk ‘/^tcp/ {++S[$NF]} END {for(a in S) print a,
S[a]}’
```

```
状态:                描述
CLOSED:                          无连接是活动的或正在进行
LISTEN:                          服务器在等待进入呼叫
SYN_RECV:           一个连接请求已经到达，等待确认
SYN_SENT:           应用已经开始，打开一个连接
ESTABLISHED: 正常数据传输状态
FIN_WAIT1:          应用说它已经完成
FIN_WAIT2:          另一边已同意释放
ITMED_WAIT: 等待所有分组死掉
CLOSING:            两边同时尝试关闭
TIME_WAIT:          另一边已初始化一个释放
LAST_ACK:           等待所有分组死掉
```

7.2. ss

```
# ss
State      Recv-Q Send-Q      Local Address:Port          Peer
Address:Port
CLOSE-WAIT 1      0           192.168.3.124:19644
130.75.116.209:http
CLOSE-WAIT 1      0           192.168.3.124:31289
170.224.194.69:https
CLOSE-WAIT 1      0           192.168.3.124:64903
198.20.8.241:https
CLOSE-WAIT 1      0           192.168.3.124:64902
198.20.8.241:https
CLOSE-WAIT 1      0           192.168.3.124:27528
```

```
170.224.160.205:https
CLOSE-WAIT 1      0           192.168.3.124:10152
198.20.8.241:https
CLOSE-WAIT 1      0           192.168.3.124:18263
170.224.194.69:http
CLOSE-WAIT 1      0           192.168.3.124:18262
170.224.194.69:http
CLOSE-WAIT 1      0           192.168.3.124:27792
129.89.61.70:http
CLOSE-WAIT 1      0           192.168.3.124:27595
129.89.61.70:http
CLOSE-WAIT 1      0           192.168.3.124:28970
129.89.61.70:http
CLOSE-WAIT 1      0           192.168.3.124:28158
130.75.116.210:http
CLOSE-WAIT 1      0           192.168.3.124:26186
130.75.116.210:http
CLOSE-WAIT 1      0           192.168.3.124:26185
130.75.116.210:http
CLOSE-WAIT 1      0           192.168.3.124:42563
74.125.71.99:http
CLOSE-WAIT 1      0           192.168.3.124:42564
74.125.71.99:http
CLOSE-WAIT 1      0           192.168.3.124:63459
130.75.116.202:http
CLOSE-WAIT 1      0           192.168.3.124:63458
130.75.116.202:http
ESTAB      0      0           192.168.3.124:30829
192.168.3.17:3260
ESTAB      0      0           192.168.3.124:13234
192.168.3.15:3260
ESTAB      0      0        ::ffff:192.168.3.124:ssh
::ffff:192.168.80.5:5
2682
ESTAB      0      1960     ::ffff:192.168.3.124:ssh
::ffff:192.168.80.5:5
2957


$ ss
State      Recv-Q Send-Q                Local Address:Port
Peer Address:Port
ESTAB      0      0                      192.168.80.1:38281
64.4.61.72:1863
ESTAB      0      0                      192.168.80.1:54504
```

```
112.95.240.77:8000
ESTAB      0      0                      192.168.80.1:14698
74.125.71.125:5222
ESTAB      0      0                      192.168.80.1:14697
74.125.71.125:5222
ESTAB      0      0                      192.168.80.1:54123
64.12.28.171:https
ESTAB      0      0                      192.168.80.1:4225
64.4.61.171:1863
ESTAB      0      0                      192.168.80.1:ssh
192.168.80.5:51291
ESTAB      0      0
::ffff:192.168.80.1:microsoft-ds
::ffff:192.168.80.5:51094
ESTAB      0      0                      192.168.80.1:22074
205.188.1.241:https
ESTAB      0      0                      192.168.80.1:59340
64.4.34.213:1863
ESTAB      0      0                      192.168.80.1:9766
91.189.89.114:https
ESTAB      0      0                      192.168.80.1:3300
64.4.44.78:1863
```

查看tcp流量控制相关参数值

```
root@netkiller ~ % ss -itn
State                    Recv-Q                     Send-Q
Local Address:Port
Peer Address:Port
ESTAB                    0                          0
192.168.3.14:22
192.168.3.4:63044
         cubic wscale:6,7 rto:212 rtt:10.681/8.769 ato:40
mss:1448 pmtu:1500 rcvmss:1392 advmss:1448 cwnd:10 ssthresh:16
bytes_acked:33428 bytes_received:9337 segs_out:377 segs_in:522
data_segs_out:360 data_segs_in:160 send 10.8Mbps lastsnd:68
lastrcv:72 lastack:56 pacing_rate 13.0Mbps delivery_rate
20.8Mbps app_limited busy:668ms rcv_rtt:7 rcv_space:28960
rcv_ssthresh:45776 minrtt:1.302
```

7.3. iftop - display bandwidth usage on an interface by host

```
# yum install -y iftop
```

7.4. iptraf - Interactive Colorful IP LAN Monitor

```
[root@development ~]# yum -y install iptraf
```

7.5. nload: Console application which monitors network traffic and bandwidth

CentOS

```
# yum install nload -y
```

Ubuntu

```
# sudo apt-get install nload
```

运行监控命令

```
# nload
```

```
Device eth0 [172.16.3.90] (1/5):
===============================================================
==============
Incoming:
```

```
                                                     Curr:
10.00 kBit/s
                                                     Avg:
103.95 kBit/s
                                                     Min: 0.00
Bit/s
             ||                                      Max: 3.23
MBit/s
             ##                                      Ttl:
1090.93 GByte
Outgoing:

                                                     Curr:
12.84 kBit/s
                                                     Avg: 15.29
kBit/s
                                                     Min: 0.00
Bit/s
                                                     Max:
206.63 kBit/s
                                                     Ttl: 48.57
GByte
```

7.6. bwm - Bandwidth Monitor

```
Bandwidth Monitor 1.1.0

       Iface        RX(KB/sec)   TX(KB/sec)   Total(KB/sec)

          lo             8.366        8.366          16.732
        eth0            24.120      100.005         124.125
        eth1             0.000        0.000           0.000

       Total            32.486      108.371         140.857

Hit CTRL-C to end this madness.
```

7.7. iptstate - A top-like display of IP Tables state table entries

```
# yum install iptstate -y
```

```
                           IPTState - IPTables State Top
Version: 2.2.2        Sort: SrcIP           b: change sorting
h: help
Source                        Destination                  Prt
State       TTL
0.0.0.0                       224.0.0.1                    igmp
0:09:49
192.168.2.1:45981             192.168.2.1:22               tcp
TIME_WAIT     0:01:33
192.168.2.1:46009             192.168.2.1:22               tcp
TIME_WAIT     0:01:57
192.168.2.1:45915             192.168.2.1:22               tcp
TIME_WAIT     0:00:58
192.168.2.1:45975             192.168.2.1:22               tcp
TIME_WAIT     0:01:31
192.168.2.1:54922             202.141.160.110:80           tcp
TIME_WAIT     0:00:57
192.168.2.1:46000             192.168.2.1:22               tcp
TIME_WAIT     0:01:54
192.168.2.1:45973             192.168.2.1:22               tcp
TIME_WAIT     0:01:31
192.168.2.1:45855             192.168.2.1:22               tcp
TIME_WAIT     0:00:26
192.168.2.1:45990             192.168.2.1:22               tcp
TIME_WAIT     0:01:36
192.168.2.1:45822             192.168.2.1:22               tcp
TIME_WAIT     0:00:01
192.168.2.1:45926             192.168.2.1:22               tcp
TIME_WAIT     0:01:01
```

8. Service

8.1. NFS

nfsstat

```
neo@monitor:~$ nfsstat
Client rpc stats:
calls      retrans    authrefrsh
1453045225   19702      744

Client nfs v3:
null         getattr      setattr      lookup       access
readlink
0         0% 114943957  8% 348670069 25% 289174215 20%
133022875  9% 40252     0%
read         write        create       mkdir        symlink
mknod
81907703  5% 99851126  7% 81782798  5% 5528575   0% 3450
0% 427       0%
remove       rmdir        rename       link         readdir
readdirplus
5178074   0% 1021367   0% 79872796  5% 0         0% 7300163
0% 21591431  1%
fsstat       fsinfo       pathconf     commit
30857752  2% 10        0% 5         0% 83581680  6%

Client nfs v4:
null         read         write        commit       open
open_conf
0         0% 3449823   5% 299       0% 248       0% 3494
0% 3066      0%
open_noat    open_dgrd    close        setattr      fsinfo
renew
0         0% 0         0% 3182      0% 1279      0% 385
0% 69        0%
setclntid    confirm      lock         lockt        locku
access
997       0% 997       0% 0         0% 0         0% 0
0% 760098    1%
```

```
getattr      lookup       lookup_root  remove       rename
link
1638029   2% 54272     0% 224       0% 4         0% 251
0% 0         0%
symlink      create       pathconf     statfs       readlink
readdir
6         0% 214       0% 193       0% 62872466 91% 391
0% 3601      0%
server_caps  delegreturn  getacl       setacl
fs_locations
578       0% 35        0% 0         0% 0         0% 0
0%
```

nfswatch

```
yum install -y nfswatch
```

```
J13-85-www                 Mon Sep 19 18:33:54 2011   Elapsed
time:   00:00:30
Interval packets:    125711 (network)      61695 (to host)
0 (dropped)
Total packets:       140549 (network)      68996 (to host)
0 (dropped)
                     Monitoring packets from interface eth0
                     int   pct    total
int   pct    total
NFS3 Read              0    0%        0 TCP Packets
61688  100%    68973
NFS3 Write             0    0%        0 UDP Packets
0    0%        1
NFS Read               0    0%        0 ICMP Packets
0    0%        0
NFS Write              0    0%        0 Routing Control
0    0%        0
NFS Mount              0    0%        0 Addr Resolution
0    0%        3
Port Mapper            0    0%        0 Rev Addr Resol
0    0%        0
RPC Authorization  59257   96%    66197 Ether/FDDI Bdcst
```

```
0    0%         3
Other RPC Packets      1    0%         5 Other Packets
7    0%       19
                                 0 file systems
     File Sys        int   pct    total      File Sys
int   pct    total
```

8.2. apachetop

```
# yum install apachetop -y
```

```
# apachetop
last hit: 00:00:00         atop runtime:  0 days, 00:00:00
09:42:54
All:            0 reqs (   0.0/sec)         0.0B (
0.0B/sec)       0.0B/req
2xx:      0 ( 0.0%) 3xx:       0 ( 0.0%) 4xx:     0 ( 0.0%)
5xx:    0 ( 0.0%)
R (  1s):       0 reqs (   0.0/sec)         0.0B (
0.0B/sec)       0.0B/req
2xx:      0 ( 0.0%) 3xx:       0 ( 0.0%) 4xx:     0 ( 0.0%)
5xx:    0 ( 0.0%)
```

9. 文件监控

https://github.com/facebook/watchman

10. watchdog

11. nmon

http://nmon.sourceforge.net/

例 5.1. nmon

```
$ apt-cache search nmon
libtime-modules-perl - Various Perl modules for time/date
manipulation
nmon - performance monitoring tool for Linux
xfce4-genmon-plugin - Generic Monitor for the Xfce4 panel
xfce4-goodies - enhancements for the Xfce4 Desktop Environment

neo@monitor:~$ sudo apt-get install nmon

neo@monitor:~$ nmon
```

nmon -f -s 360 -c 86400 -m /home/user/nmon

12. Hardware

12.1. temperature/voltage/fan

lm-sensors - utilities to read temperature/voltage/fan sensors

```
$ sudo apt-get install lm-sensors
$ sudo sensors-detect
$ sensors
```

12.2. mcelog - Decode kernel machine check log on x86 machines

```
$ sudo apt-get install mcelog
```

```
Decode machine check ASCII output from kernel logs
Options:
--cpu CPU           Set CPU type CPU to decode (see below for valid
types)
--cpumhz MHZ        Set CPU Mhz to decode time (output unreliable, not
needed on new kernels)
--raw                (with --ascii) Dump in raw ASCII format for machine
processing
--daemon            Run in background waiting for events (needs newer
kernel)
--ignorenodev       Exit silently when the device cannot be opened
--file filename     With --ascii read machine check log from filename
instead of stdin
--syslog            Log decoded machine checks in syslog (default stdout
or syslog for daemon)
--syslog-error       Log decoded machine checks in syslog with error
level
--no-syslog         Never log anything to syslog
--logfile filename  Append log output to logfile instead of stdout
--dmi               Use SMBIOS information to decode DIMMs (needs root)
--no-dmi            Don't use SMBIOS information
--dmi-verbose       Dump SMBIOS information (for debugging)
--filter            Inhibit known bogus events (default on)
--no-filter         Don't inhibit known broken events
--config-file filename Read config information from config file instead
of /etc/mcelog/mcelog.conf
--foreground        Keep in foreground (for debugging)
--num-errors N      Only process N errors (for testing)
```

```
--pidfile file       Write pid of daemon into file
--no-imc-log         Disable extended iMC logging
```

13. sar - System Activity Reporter

sar 是 System Activity Reporter（系统活动情况报告）的缩写。

sar工具将对系统当前的状态进行取样，然后通过计算数据和比例来表达系统的当前运行状态。它的特点是可以连续对系统取样，获得大量的取样数据；取样数据和分析的结果都可以存入文件，所需的负载很小。sar是目前Linux上最为全面的系统性能分析工具之一，可以从14个大方面对系统的活动进行报告，包括文件的读写情况、系统调用的使用情况、串口、CPU效率、内存使用状况、进程活动及IPC有关的活动等，使用也是较为复杂。

sar命令常用格式
sar [options] [-A] [-o file] t [n]

其中:

t为采样间隔，n为采样次数，默认值是1；

-o file表示将命令结果以二进制格式存放在文件中，file 是文件名。

options 为命令行选项，sar命令常用选项如下：

-A：所有报告的总和

-u：输出CPU使用情况的统计信息

-v：输出inode、文件和其他内核表的统计信息

-d：输出每一个块设备的活动信息

-r：输出内存和交换空间的统计信息

-b：显示I/O和传送速率的统计信息

-a：文件读写情况

-c：输出进程统计信息，每秒创建的进程数

-R：输出内存页面的统计信息

-y：终端设备活动情况

```
    -w：输出系统交换活动信息

> Report CPU utilization


[root@netkiller ~]# sar -u 1 3
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月08日
_x86_64_        (2 CPU)

15时05分29秒     CPU     %user     %nice   %system   %iowait
%steal     %idle
15时05分30秒     all      0.00      0.00      0.00      0.00
0.00    100.00
15时05分31秒     all      0.00      0.00      0.50      0.00
0.00     99.50
15时05分32秒     all      0.50      0.00      0.00      0.50
0.00     99.00
平均时间:     all      0.17      0.00      0.17      0.17
0.00     99.50

%user:   显示在用户级别(application)运行使用 CPU 总时间的百分比.
%nice:   显示在用户级别,用于nice操作,所占用CPU总时间的百分比.
%system: 在核心级别(kernel)运行所使用 CPU 总时间的百分比.
%iowait: 显示用于等待I/O操作占用CPU总时间的百分比.
%steal:  管理程序(hypervisor)为另一个虚拟进程提供服务而等待虚拟CPU的百
分比.
%idle:   显示CPU空闲时间占用CPU总时间的百分比.

> Report status of inode, file and other kernel tables


[root@netkiller ~]# sar -v  1 3
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月08日
_x86_64_        (2 CPU)

15时07分57秒 dentunusd   file-nr  inode-nr    pty-nr
15时07分58秒     47524       640     46025         2
15时07分59秒     47524       640     46025         2
15时08分00秒     47524       640     46025         2
平均时间:     47524       640     46025         2

dentunusd: 目录缓存中未使用的缓存条目数
file-nr:   由系统使用的文件数
inode-nr:  由系统使用的inode数
pty-nr:    系统所使用的伪终端数
```

```
> 查看平均负载


sar -q: 查看平均负载

指定-q后，就能查看运行队列中的进程数、系统上的进程大小、平均负载等；与其它
命令相比，它能查看各项指标随时间变化的情况；

[root@netkiller ~]# sar -q 1 3
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月08日
_x86_64_        (2 CPU)

15时24分06秒   runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15
15时24分07秒         0       204      0.00      0.01      0.05
15时24分08秒         1       204      0.00      0.01      0.05
15时24分09秒         0       204      0.00      0.01      0.05
平均时间:          0       204      0.00      0.01      0.05

runq-sz:  运行队列的长度（等待运行的进程数）
plist-sz: 进程列表中进程（processes）和线程（threads）的数量
ldavg-1:  最后1分钟的系统平均负载
ldavg-5:  过去5分钟的系统平均负载
ldavg-15: 过去15分钟的系统平均负载

> Report memory statistics


[root@kvm ~]# sar -R 1 5
Linux 2.6.32-358.11.1.el6.x86_64 (kvm)  11/04/2013
_x86_64_        (24 CPU)

04:12:49 PM   frmpg/s   bufpg/s   campg/s
04:12:50 PM   -174.00      0.00      0.00
04:12:51 PM    -27.08      0.00      0.00
04:12:52 PM    -73.27      0.00      0.00
04:12:53 PM   -498.00      0.00      0.00
04:12:54 PM    322.00      0.00      0.00
Average:       -90.54      0.00      0.00

frmpg/s :每秒钟系统释放的内存页数. 如果是负值，表示每秒钟被系统分配的内存
页数.
bufpg/s :每秒钟系统分配多少内存页作为buffer使用. 如果是负值，表示系统在
回收一定的buffer空间.
campg/s :每秒钟系统分配多少内存页作为bcached使用. 如果是负值，表示系统在
```

```
回收一定的cached空间.

> 查看页面交换发生状况


[root@kvm ~]# sar -W
Linux 2.6.32-358.11.1.el6.x86_64 (kvm)  11/04/2013
_x86_64_        (24 CPU)

12:00:01 AM  pswpin/s pswpout/s
12:10:01 AM      0.00      0.00
12:20:01 AM      0.00      0.00
12:30:01 AM      0.00      0.00
12:40:01 AM      0.00      0.00
12:50:01 AM      0.00      0.00

pswpin/s
    Total number of swap pages the system brought in per
second.

pswpout/s
    Total number of swap pages the system brought out per
second.

> Report task creation and system switching activity


[root@kvm ~]# sar -w 1 5
Linux 2.6.32-358.11.1.el6.x86_64 (kvm)  11/05/2013
_x86_64_        (24 CPU)

03:09:01 PM    proc/s   cswch/s
03:09:02 PM      1.00  21017.00
03:09:03 PM      1.02  18507.14
03:09:04 PM      1.00  20803.00
03:09:05 PM      0.99  17787.13
03:09:06 PM      1.04  22041.67
Average:         1.01  20016.57

proc/s:  每秒创建的任务的总数.
cswch/s: 每秒上下文切换的总数.

> Report I/O and transfer rate statistics.
```

```
[root@kvm ~]# sar -b 1 5
Linux 2.6.32-358.11.1.el6.x86_64 (kvm)  11/05/2013
_x86_64_        (24 CPU)

03:20:15 PM       tps      rtps      wtps   bread/s   bwrtn/s
03:20:16 PM     18.00      0.00     18.00      0.00    383.00
03:20:17 PM      5.05      0.00      5.05      0.00     72.73
03:20:18 PM      0.00      0.00      0.00      0.00      0.00
03:20:19 PM      0.00      0.00      0.00      0.00      0.00
03:20:20 PM      0.00      0.00      0.00      0.00      0.00
Average:         4.60      0.00      4.60      0.00     91.00

tps：     每秒钟向物理设备发出请求(读与写)的总数
rtps:     每秒钟向物理设备发出读请求的总数
wtps:     每秒钟向物理设备发出写请求的总数
bread/s: 每秒从块设备中读取的数据总数
bwrtn/s: 每秒向块设备中写入的数据总数

> Report paging statistics


[root@kvm ~]# sar -B 1 5
Linux 2.6.32-358.11.1.el6.x86_64 (kvm)  11/05/2013
_x86_64_        (24 CPU)

03:36:32 PM  pgpgin/s pgpgout/s   fault/s  majflt/s  pgfree/s
pgscank/s pgscand/s pgsteal/s    %vmeff
03:36:33 PM    192.00    384.00   1125.00      0.00   1709.00
0.00      0.00      0.00      0.00
03:36:34 PM      0.00     16.16    240.40      0.00    935.35
0.00      0.00      0.00      0.00
03:36:35 PM      0.00      1.01    273.74      0.00   1009.09
0.00      0.00      0.00      0.00
03:36:36 PM      0.00    396.04   1052.48      0.00    878.22
0.00      0.00      0.00      0.00
03:36:37 PM      0.00      0.00    228.00      0.00    997.00
0.00      0.00      0.00      0.00
Average:        38.48    160.52    586.17      0.00   1105.81
0.00      0.00      0.00      0.00

pgpgin/s:   每秒从磁盘或SWAP置换到内存的字节数
pgpgout/s: 每秒从内存置换到磁盘或SWAP的字节数
fault/s:    每秒钟系统产生的缺页数,即主缺页与次缺页之和(major + minor)
majflt/s：  每秒钟产生的主缺页数
pgfree/s:   每秒被放入空闲队列中的页个数
```

```
pgscank/s: 每秒被kswapd扫描的页个数
pgscand/s: 每秒直接被扫描的页个数
pgsteal/s: 每秒钟从cache中被回收来满足内存需要的页个数
%vmeff:    每秒回收的页(pgsteal)占总扫描页(pgscank+pgscand)的百分比

缺页异常:
        major（内存中没有需要的数据）
        minor （内存中有这样的数据，单最先不是该进程的）

> Report network statistics


sar命令使用-n选项可以汇报网络相关信息，可用的参数包括：DEV、EDEV、SOCK和
FULL。

1) 如果你使用DEV关键字，那么sar将汇报和网络设备相关的信息，如lo，eth0或
eth1等

[root@netkiller ~]# sar -n DEV 1 1
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月08日
_x86_64_        (2 CPU)

15时30分12秒     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s
rxcmp/s   txcmp/s  rxmcst/s
15时30分13秒       br0      3.03      1.01      0.14      0.16
0.00      0.00      0.00
15时30分13秒      eth0      3.03      1.01      0.18      0.16
0.00      0.00      0.00
15时30分13秒        lo      2.02      2.02      0.09      0.09
0.00      0.00      0.00
15时30分13秒   docker0      0.00      0.00      0.00      0.00
0.00      0.00      0.00

平均时间:     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s
rxcmp/s   txcmp/s  rxmcst/s
平均时间:       br0      3.03      1.01      0.14      0.16
0.00      0.00      0.00
平均时间:      eth0      3.03      1.01      0.18      0.16
0.00      0.00      0.00
平均时间:        lo      2.02      2.02      0.09      0.09
0.00      0.00      0.00
平均时间:   docker0      0.00      0.00      0.00      0.00
0.00      0.00      0.00

IFACE：就是网络设备的名称；
```

```
rxpck/s：每秒钟接收到的包数目

txpck/s：每秒钟发送出去的包数目

rxbyt/s：每秒钟接收到的字节数

txbyt/s：每秒钟发送出去的字节数

rxcmp/s：每秒钟接收到的压缩包数目

txcmp/s：每秒钟发送出去的压缩包数目

txmcst/s：每秒钟接收到的多播包的包数目

2) 如果你使用EDEV关键字，那么会针对网络设备汇报其失败情况，例如：
[root@netkiller ~]# sar -n EDEV 1  1
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月08日
_x86_64_        (2 CPU)

15时31分29秒      IFACE   rxerr/s   txerr/s    coll/s  rxdrop/s
txdrop/s  txcarr/s  rxfram/s  rxfifo/s  txfifo/s
15时31分30秒        br0      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00
15时31分30秒       eth0      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00
15时31分30秒         lo      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00
15时31分30秒    docker0      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00

平均时间:      IFACE   rxerr/s   txerr/s    coll/s  rxdrop/s
txdrop/s  txcarr/s  rxfram/s  rxfifo/s  txfifo/s
平均时间:        br0      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00
平均时间:       eth0      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00
平均时间:         lo      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00
平均时间:    docker0      0.00      0.00      0.00      0.00
0.00      0.00      0.00      0.00      0.00

rxerr/s：每秒钟接收到的损坏的包的数目

txerr/s：当发送包时，每秒钟发生的错误数
```

coll/s: 当发送包时，每秒钟发生的冲撞(collisions)数（这个是在半双工模式下才有）

rxdrop/s：由于缓冲区满，网络设备接收端，每秒钟丢掉的网络包的数目

txdrop/s：由于缓冲区满，网络设备发送端，每秒钟丢掉的网络包的数目

txcarr/s：当发送数据包时，每秒钟载波错误发生的次数

rxfram/s：在接收数据包时，每秒钟发生的帧对齐错误的次数

rxfifo/s：在接收数据包时，每秒钟缓冲区溢出错误发生的次数

txfifo/s：在发送数据包时，每秒钟缓冲区溢出错误发生的次数

3) 如果你使用SOCK关键字，则会针对socket连接进行汇报，例如：

```
[root@netkiller ~]# sar -n SOCK 1  1
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月08日
_x86_64_        (2 CPU)

15时33分29秒    totsck    tcpsck    udpsck    rawsck   ip-frag
tcp-tw
15时33分30秒        86        47         0         0         0
67
平均时间:        86        47         0         0         0
67
```

totsck：被使用的socket的总数目

tcpsck：当前正在被使用于TCP的socket数目

udpsck：当前正在被使用于UDP的socket数目

rawsck：当前正在被使用于RAW的socket数目

ip-frag：当前的IP分片的数目

## iostat

通过iostat方便查看CPU、网卡、tty设备、磁盘、CD-ROM 等等设备的活动情况，负载信息。

```
命令格式
iostat[参数][时间][次数]

-C 显示CPU使用情况
-d 显示磁盘使用情况
-k 以 KB 为单位显示
-m 以 M 为单位显示
-N 显示磁盘阵列(LVM) 信息
-n 显示NFS 使用情况
-p[磁盘] 显示磁盘和分区的情况
-t 显示终端和CPU的信息
-x 显示详细信息
-V 显示版本信息



[root@netkiller ~]# iostat  -k
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月07日
_x86_64_        (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.14    0.00    0.23    0.39    0.00   99.25

Device:            tps    kB_read/s    kB_wrtn/s    kB_read
kB_wrtn
sda               8.65        28.46        95.63   54368942
182705652
dm-0              0.01         0.01         0.19      21684
366024

[root@netkiller ~]# iostat -x
Linux 3.10.5-3.el6.x86_64 (test23)      2017年03月07日
_x86_64_        (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.14    0.00    0.23    0.39    0.00   99.25

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s
wsec/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sda               0.00     3.31    0.49    8.16    56.95
191.26    28.70     0.03    3.95    1.85    4.07   0.95   0.82
dm-0              0.00     0.00    0.00    0.01     0.02
0.38    40.84     0.00   24.60    8.99   25.69   0.77   0.00

>cpu属性值说明
```

```
%user:   CPU处在用户模式下的时间百分比
%nice:   CPU处在带NICE值的用户模式下的时间百分比
%system: CPU处在系统模式下的时间百分比
%iowait: CPU等待输入输出完成时间的百分比
%steal:  管理程序维护另一个虚拟处理器时，虚拟CPU的无意识等待时间百分比
%idle:   CPU空闲时间百分比

> disk属性值说明


rrqm/s:      每秒进行 merge 的读操作数目.即 delta(rmerge)/s
wrqm/s:      每秒进行 merge 的写操作数目.即 delta(wmerge)/s
r/s:         每秒完成的读 I/O 设备次数.即 delta(rio)/s
w/s:         每秒完成的写 I/O 设备次数.即 delta(wio)/s
rsec/s:      每秒读扇区数.即 delta(rsect)/s
wsec/s:      每秒写扇区数.即 delta(wsect)/s
rkB/s:       每秒读K字节数.是 rsect/s 的一半,因为每扇区大小为512字节.
(需要计算)
wkB/s:       每秒写K字节数.是 wsect/s 的一半.(需要计算)
avgrq-sz:    平均每次设备I/O操作的数据大小 (扇
区).delta(rsect+wsect)/delta(rio+wio)
avgqu-sz:    平均I/O队列长度.即 delta(aveq)/s/1000 (因为aveq的单位为
毫秒).
await:       平均每次设备I/O操作的等待时间 (毫秒).即
delta(ruse+wuse)/delta(rio+wio)
svctm:       平均每次设备I/O操作的服务时间 (毫秒).即
delta(use)/delta(rio+wio)
%util:       一秒中有百分之多少的时间用于 I/O 操作,或者说一秒中有多少时间
I/O 队列是非空的.即 delta(use)/s/1000 (因为use的单位为毫秒)

如果 %util 接近 100%,说明产生的I/O请求太多,I/O系统已经满负荷,该磁盘可能
存在瓶颈.

idle小于70% IO压力就较大了,一般读取速度有较多的wait.同时可以结合vmstat
查看查看b参数(等待资源的进程数)和wa参数(IO等待所占用的CPU时间的百分比,高
过30%时IO压力高)

另外 await 的参数也要多和 svctm 来参考.差的过高就一定有 IO 的问题.一般地
系统IO响应时间(await)应该低于5ms，如果大于10ms就比较大了.

avgqu-sz 也是个做 IO 调优时需要注意的地方,这个就是直接每次操作的数据的大
小,如果次数多,但数据拿的小的话,其实 IO 也会很小.如果数据拿的大,才IO 的数
据会高.也可以通过 avgqu-sz × ( r/s or w/s ) = rsec/s or wsec/s.也
```

就是讲,读定速度是这个来决定的.

一个不错的例子.(I/O 系统 vs. 超市排队)

举一个例子,我们在超市排队 checkout 时,怎么决定该去哪个交款台呢?

首当是看排的队人数,5个人总比20人要快吧?

除了数人头,我们也常常看看前面人购买的东西多少,如果前面有个采购了一星期食品的大妈,那么可以考虑换个队排了.

还有就是收银员的速度了,如果碰上了连 钱都点不清楚的新手,那就有的等了.

另外,时机也很重要,可能 5 分钟前还人满为患的收款台,现在已是人去楼空,这时候交款可是很爽啊,当然,前提是那过去的 5 分钟里所做的事情比排队要有意义

I/O 系统也和超市排队有很多类似之处:

r/s+w/s 类似于交款人的总数

平均队列长度(avgqu-sz)类似于单位时间里平均排队人的个数

平均服务时间(svctm)类似于收银员的收款速度

平均等待时间(await)类似于平均每人的等待时间

平均I/O数据(avgrq-sz)类似于平均每人所买的东西多少

I/O 操作率 (%util)类似于收款台前有人排队的时间比例.

我们可以根据这些数据分析出 I/O 请求的模式,以及 I/O 的速度和响应时间.

> 下面是别人写的这个参数输出的分析

```
// # iostat -x 1
avg-cpu: %user %nice %sys %idle
16.24 0.00 4.31 79.44
Device:              rrqm/s wrqm/s r/s   w/s    rsec/s wsec/s
rkB/s  wkB/s  avgrq-sz  avgqu-sz await svctm %util
/dev/cciss/c0d0      0.00   44.90 1.02  27.55    8.16  579.59
4.08   289.80  20.57      22.35  78.21  5.00 14.29
```

上面的 iostat 输出表明秒有 28.57 次设备 I/O 操作: 总IO(io)/s = r/s(读) +w/s(写) = 1.02+27.55 = 28.57 (次/秒) 其中写操作占了主体 (w:r = 27:1).

平均每次设备 I/O 操作只需要 5ms 就可以完成,但每个 I/O 请求却需要等上 78ms,为什么? 因为发出的 I/O 请求太多 (每秒钟约 29 个),假设这些请求是同时发出的,那么平均等待时间可以这样计算:

```
平均等待时间 = 单个 I/O 服务时间 * ( 1 + 2 + … + 请求总数-1) / 请求总
数

应用到上面的例子: 平均等待时间 = 5ms * (1+2+…+28)/29 = 70ms,和
iostat 给出的78ms 的平均等待时间很接近.这反过来表明 I/O 是同时发起的.

每秒发出的 I/O 请求很多 (约 29 个),平均队列却不长 (只有 2 个 左右),这表
明这 29 个请求的到来并不均匀,大部分时间 I/O 是空闲的.

一秒中有 14.29% 的时间 I/O 队列中是有请求的,也就是说,85.71% 的时间里
I/O 系统无事可做,所有 29 个 I/O 请求都在142毫秒之内处理掉了.

delta(ruse+wuse)/delta(io) = await = 78.21 =>
delta(ruse+wuse)/s =78.21 * delta(io)/s = 78.21*28.57 = 2232.8,
表明每秒内的I/O请求总共需要等待2232.8ms.所以平均队列长度应为
2232.8ms/1000ms = 2.23,
而 iostat 给出的平均队列长度 (avgqu-sz) 却为 22.35,为什么?! 因为
iostat 中有 bug,avgqu-sz 值应为 2.23,而不是 22.35.



## vmstat

vmstat是Virtual Meomory Statistics（虚拟内存统计）的缩写，可实时动态
监视操作系统的虚拟内存、进程、CPU活动.

vmstat的语法
    vmstat [-V] [-n] [delay [count]]

r,  可运行队列的线程数，这些线程都是可运行状态，只不过 CPU 暂时不可用.
b,  被 blocked 的进程数，正在等待 IO 请求;
in, 每秒被处理过的中断数
cs, 每秒系统上正在做上下文切换的数目
us, 用户占用 CPU 的百分比
sy, 内核和中断占用 CPU 的百分比
wa, 所有可运行的线程被 blocked 以后都在等待 IO，这时候 CPU 空闲的百分比
id, CPU 完全空闲的百分比

swpd: 使用虚拟内存大小
free: 可用内存大小
buff: 用作缓冲的内存大小
cache: 用作缓存的内存大小

si: 每秒从交换区写到内存的大小
so: 每秒写入交换区的内存大小
```

```
bi: 每秒读取的块数
bo: 每秒写入的块数


### badblocks

badblock 命令用于查找磁盘中损坏的区块.

badblock (options)  (参数)

options:
    -b<区块大小>: 指定磁盘的区块大小, 单位为字节
    -o<输出文件>: 将检查的结果写入指定的输出文件
    -c: 每个区块检查的次数,默认是16次
    -s: 在检查时显示进度
    -v: 执行时显示详细的信息
    -w: 在检查时,执行写入测试

 参数:
    磁盘装置:    指定要检查的磁盘装置
    磁盘区块数: 指定磁盘装置的区块总数
    启始区块:    指定要从哪个区块开始检查

// 检查硬盘是否产生坏道并输出到badblocks.log中
badblocks -s -v -o /root/badblocks.log /dev/sda


// badblocks以4096字节为一个“block”,每一个“block”检查1次, 将结果输出到“hda-badblocks-list.1”文件中, 由第51000 block开始, 到63000 block结束
badblocks -b 4096 -c 1 /dev/hda1 -o hda-badblocks-list.1 63000 51000

> 利用硬盘的重分配特性修复坏道


1) 硬盘上的芯片存有一个GList, 里面存储着盘面上的坏道信息, 当读写到其记录的地址时会自动重映射另一个地址来代替损坏的区域.
而往其中添加内容很简单: 只要往坏道上写数据(读不行), 硬盘会自动重映射.

badblocks -w [-f] /dev/sdXX [-s -b4096] end start

    -w: 写入命令, 通过在坏道地址强制写入来让硬盘自动重映射.
    -f: 强制写入, 在已确定目标不被系统读写而-w仍然拒绝写入时使用.这个参数
```

应该尽量避免!

end,start：强制写入的开始和终止块地址，与-b制定的大小相配和.

2）使用fsck -a /dev/sda1

磁盘坏道分为三种：0磁道坏道，逻辑坏道，硬盘坏道。

其中逻辑坏道可以使用上面的方法修复，0磁道坏道的修复方法是隔离0磁道，使用fdsk划分区的时候从1磁道开始划分区。

如果是硬盘坏道的话，只能隔离不能修复。

硬盘坏道的监测方法：使用上述方法检测修复后，再使用badblocks -s -v -o /root/badblocks.log /dev/sda监测看是否还有坏道存在，如果坏道还是存在的话说明坏道属于硬盘坏道。

硬盘坏道隔离方法，首先记录监测出的硬盘坏道，然后分区的时候把硬盘坏道所在的扇区分在一个分区（大小一般大于坏扇区大小），划分出的坏道分区不使用即可达到隔离的目的。隔离只是暂时方案，建议尽快更换硬盘，因为坏道会扩散，以免以后出现严重的数据问题。

14. SMS

14.1. gnokii

http://www.gnokii.org

安装

Ubuntu

```
neo@monitor:~$ apt-cache search gnokii
opensync-plugin-gnokii - Opensync gnokii plugin
gnokii - Datasuite for mobile phone management
gnokii-cli - Datasuite for mobile phone management (console
interface)
gnokii-common - Datasuite for mobile phone management (base
files)
gnokii-smsd - SMS Daemon for mobile phones
gnokii-smsd-mysql - SMSD plugin for MySQL storage backend
gnokii-smsd-pgsql - SMSD plugin for PostgreSQL storage backend
libgnokii-dev - Gnokii mobile phone interface library
(development files)
libgnokii5 - Gnokii mobile phone interface library
xgnokii - Datasuite for mobile phone management (X interface)


neo@monitor:~$ sudo apt-get install gnokii-cli
```

CentOS

```
# yum search gnokii

gnokii-devel.x86_64 : Gnokii development files
gnokii-smsd.x86_64 : Gnokii SMS daemon
gnokii-smsd-mysql.x86_64 : MySQL support for Gnokii SMS daemon
gnokii-smsd-pgsql.x86_64 : PostgreSQL support for Gnokii SMS
daemon
```

```
gnokii-smsd-sqlite.x86_64 : SQLite support for Gnokii SMS
daemon
gnokii.x86_64 : Linux/Unix tool suite for various mobile phones
xgnokii.x86_64 : Graphical Linux/Unix tool suite for various
mobile phones
```

安装

```
# yum install -y gnokii
```

配置

```
vim /etc/gnokiirc
or
vim ~/.gnokiirc

[global]
port = /dev/ttyS0
model = AT
initlength = default
connection = serial
serial_baudrate = 19200
smsc_timeout = 10
```

发送测试短信

```
$ echo "This is a test message" | gnokii --sendsms +13113668890

$ gnokii --sendsms number <<EOF
hi neo,
This is a test message
EOF
```

接收短信

```
# gnokii --smsreader
GNOKII Version 0.6.31
Entered sms reader mode...

SMS received from number: 8613113668890
Got message 11: hi
```

拨打电话

```
$ gnokii --dialvoice number
```

14.2. AT Commands

发送短信

AT+CSCA=+8613010888500 是设置短信中心号码，只需第一次使用

```
AT
AT+CSCA=+8613010888500
AT+CMGF=1
AT+CMGS="13122993040"
Hello,This is the test of GSM module! Ctrl+z
```

语音通话

```
at+fclass=8
```

```
at#vsps=0
at+vgs=130
at+vsp=1
at+vls=7
ATDT13113668890
```

15. IPMI (Intelligent Platform Management Interface)

```
OpenIPMI： http://openipmi.sourceforge.net/
Ipmitool：  http://ipmitool.sourceforge.net/
ipmiutil：  http://ipmiutil.sourceforge.net/
```

15.1. OpenIPMI

```
# yum install OpenIPMI
```

start

```
/etc/init.d/ipmi start
Starting ipmi drivers:                                     [
OK  ]
```

15.2. freeipmi

```
# yum install freeipmi
```

ipmiping

```
# ipmiping 172.16.5.52
ipmiping 172.16.5.52 (172.16.5.52)
response received from 172.16.5.52: rq_seq=57
response received from 172.16.5.52: rq_seq=58
response received from 172.16.5.52: rq_seq=59
response received from 172.16.5.52: rq_seq=60
response received from 172.16.5.52: rq_seq=61
```

```
^C--- ipmiping 172.16.5.52 statistics ---
5 requests transmitted, 5 responses received in time, 0.0%
packet loss
```

ipmimonitoring

```
# ipmimonitoring -h 172.16.1.23 -u root -pcalvin
Caching SDR repository information: /root/.freeipmi/sdr-
cache/sdr-cache-J10-51-Memcache-0.172.16.5.23
Caching SDR record 125 of 125 (current record ID 125)
Record_ID | Sensor Name | Sensor Group | Monitoring Status|
Sensor Units | Sensor Reading
7 | Ambient Temp | Temperature | Nominal | C | 27.000000
9 | CMOS Battery | Battery | Nominal | N/A | 'OK'
10 | VCORE PG | Voltage | Nominal | N/A | 'State Deasserted'
11 | VCORE PG | Voltage | Nominal | N/A | 'State Deasserted'
13 | 1.5V PG | Voltage | Nominal | N/A | 'State Deasserted'
14 | 1.8V PG | Voltage | Nominal | N/A | 'State Deasserted'
15 | 3.3V PG | Voltage | Nominal | N/A | 'State Deasserted'
16 | 5V PG | Voltage | Nominal | N/A | 'State Deasserted'
17 | 0.75VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
19 | HEATSINK PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
20 | iDRAC6 Ent PRES | Entity Presence | Nominal | N/A |
'Entity Present'
21 | USB CABLE PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
22 | STOR ADAPT PRES | Entity Presence | Nominal | N/A |
'Entity Present'
23 | RISER2 PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
24 | RISER1 PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
25 | 0.75 VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
26 | MEM PG | Voltage | Nominal | N/A | 'State Deasserted'
27 | MEM PG | Voltage | Nominal | N/A | 'State Deasserted'
28 | 0.9V PG | Voltage | Nominal | N/A | 'State Deasserted'
29 | VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
30 | VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
31 | 1.8 PLL PG | Voltage | Nominal | N/A | 'State Deasserted'
32 | 1.8 PLL PG | Voltage | Nominal | N/A | 'State Deasserted'
33 | 8.0V PG | Voltage | Nominal | N/A | 'State Deasserted'
```

```
34 | 1.1V PG | Voltage | Nominal | N/A | 'State Deasserted'
35 | 1.0V LOM PG | Voltage | Nominal | N/A | 'State Deasserted'
36 | 1.0V AUX PG | Voltage | Nominal | N/A | 'State Deasserted'
37 | 1.05V PG | Voltage | Nominal | N/A | 'State Deasserted'
38 | FAN MOD 1A RPM | Fan | Nominal | RPM | 5040.000000
39 | FAN MOD 2A RPM | Fan | Nominal | RPM | 7800.000000
40 | FAN MOD 3A RPM | Fan | Nominal | RPM | 8040.000000
41 | FAN MOD 4A RPM | Fan | Nominal | RPM | 8760.000000
42 | FAN MOD 5A RPM | Fan | Nominal | RPM | 8640.000000
43 | FAN MOD 6A RPM | Fan | Nominal | RPM | 5040.000000
44 | FAN MOD 1B RPM | Fan | Nominal | RPM | 3840.000000
45 | FAN MOD 2B RPM | Fan | Nominal | RPM | 6000.000000
46 | FAN MOD 3B RPM | Fan | Nominal | RPM | 6120.000000
47 | FAN MOD 4B RPM | Fan | Nominal | RPM | 6600.000000
48 | FAN MOD 5B RPM | Fan | Nominal | RPM | 6600.000000
49 | FAN MOD 6B RPM | Fan | Nominal | RPM | 3840.000000
50 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
51 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
52 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
53 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
54 | Presence  | Entity Presence | Nominal | N/A | 'Entity
Present'
55 | Status | Processor | Nominal | N/A | 'Processor Presence
detected'
56 | Status | Processor | Nominal | N/A | 'Processor Presence
detected'
57 | Status | Power Supply | Nominal | N/A | 'Presence
detected'
58 | Status | Power Supply | Critical | N/A | 'Presence
detected' 'Power Supply input lost (AC/DC)'
59 | Riser Config | Cable/Interconnect | Nominal | N/A |
'Cable/Interconnect is connected'
60 | OS Watchdog | Watchdog 2 | Nominal | N/A | 'OK'
62 | Intrusion | Physical Security | Nominal | N/A | 'OK'
64 | Fan Redundancy | Fan | Nominal | N/A | 'Fully Redundant'
66 | Drive | Drive Slot | Nominal | N/A | 'Drive Presence'
67 | Cable SAS A | Cable/Interconnect | Nominal | N/A |
'Cable/Interconnect is connected'
68 | Cable SAS B | Cable/Interconnect | Nominal | N/A |
'Cable/Interconnect is connected'
116 | Current | Current | Nominal | A | 1.400000
```

```
118 | Voltage | Voltage | Nominal | V | 220.000000
120 | System Level | Current | Nominal | W | 329.000000
123 | ROMB Battery | Battery | Nominal | N/A | 'OK'
```

ipmi-sensors

```
# ipmi-sensors -h 172.16.5.23 -u root -pcalvin
1: Temp (Temperature): NA (NA/90.00): [NA]
2: Temp (Temperature): NA (NA/90.00): [NA]
3: Temp (Temperature): NA (NA/NA): [NA]
4: Ambient Temp (Temperature): NA (NA/NA): [NA]
5: Temp (Temperature): NA (NA/NA): [NA]
6: Ambient Temp (Temperature): NA (NA/NA): [NA]
7: Ambient Temp (Temperature): 27.00 C (3.00/47.00): [OK]
8: Planar Temp (Temperature): NA (3.00/97.00): [NA]
9: CMOS Battery (Battery): [OK]
10: VCORE PG (Voltage): [State Deasserted]
11: VCORE PG (Voltage): [State Deasserted]
12: IOH THERMTRIP (Temperature): [NA]
13: 1.5V PG (Voltage): [State Deasserted]
14: 1.8V PG (Voltage): [State Deasserted]
15: 3.3V PG (Voltage): [State Deasserted]
16: 5V PG (Voltage): [State Deasserted]
17: 0.75VTT PG (Voltage): [State Deasserted]
18: PFault Fail Safe (Voltage): [Unknown]
19: HEATSINK PRES (Entity Presence): [Entity Present]
20: iDRAC6 Ent PRES (Entity Presence): [Entity Present]
21: USB CABLE PRES (Entity Presence): [Entity Present]
22: STOR ADAPT PRES (Entity Presence): [Entity Present]
23: RISER2 PRES (Entity Presence): [Entity Present]
24: RISER1 PRES (Entity Presence): [Entity Present]
25: 0.75 VTT PG (Voltage): [State Deasserted]
26: MEM PG (Voltage): [State Deasserted]
27: MEM PG (Voltage): [State Deasserted]
28: 0.9V PG (Voltage): [State Deasserted]
29: VTT PG (Voltage): [State Deasserted]
30: VTT PG (Voltage): [State Deasserted]
31: 1.8 PLL PG (Voltage): [State Deasserted]
32: 1.8 PLL PG (Voltage): [State Deasserted]
33: 8.0V PG (Voltage): [State Deasserted]
34: 1.1V PG (Voltage): [State Deasserted]
35: 1.0V LOM PG (Voltage): [State Deasserted]
```

```
36: 1.0V AUX PG (Voltage): [State Deasserted]
37: 1.05V PG (Voltage): [State Deasserted]
38: FAN MOD 1A RPM (Fan): 5040.00 RPM (1920.00/NA): [OK]
39: FAN MOD 2A RPM (Fan): 8040.00 RPM (1920.00/NA): [OK]
40: FAN MOD 3A RPM (Fan): 7920.00 RPM (1920.00/NA): [OK]
41: FAN MOD 4A RPM (Fan): 9240.00 RPM (1920.00/NA): [OK]
42: FAN MOD 5A RPM (Fan): 9120.00 RPM (1920.00/NA): [OK]
43: FAN MOD 6A RPM (Fan): 5040.00 RPM (1920.00/NA): [OK]
44: FAN MOD 1B RPM (Fan): 3840.00 RPM (1920.00/NA): [OK]
45: FAN MOD 2B RPM (Fan): 6120.00 RPM (1920.00/NA): [OK]
46: FAN MOD 3B RPM (Fan): 6000.00 RPM (1920.00/NA): [OK]
47: FAN MOD 4B RPM (Fan): 6960.00 RPM (1920.00/NA): [OK]
48: FAN MOD 5B RPM (Fan): 6960.00 RPM (1920.00/NA): [OK]
49: FAN MOD 6B RPM (Fan): 3840.00 RPM (1920.00/NA): [OK]
50: Presence (Entity Presence): [Entity Present]
51: Presence (Entity Presence): [Entity Present]
52: Presence (Entity Presence): [Entity Present]
53: Presence (Entity Presence): [Entity Present]
54: Presence  (Entity Presence): [Entity Present]
55: Status (Processor): [Processor Presence detected]
56: Status (Processor): [Processor Presence detected]
57: Status (Power Supply): [Presence detected]
58: Status (Power Supply): [Presence detected][Power Supply
input lost (AC/DC)]
59: Riser Config (Cable/Interconnect): [Cable/Interconnect is
connected]
60: OS Watchdog (Watchdog 2): [OK]
61: SEL (Event Logging Disabled): [Unknown]
62: Intrusion (Physical Security): [OK]
63: PS Redundancy (Power Supply): [NA]
64: Fan Redundancy (Fan): [Fully Redundant]
65: CPU Temp Interf (Temperature): [NA]
66: Drive (Drive Slot): [Drive Presence]
67: Cable SAS A (Cable/Interconnect): [Cable/Interconnect is
connected]
68: Cable SAS B (Cable/Interconnect): [Cable/Interconnect is
connected]
69: DKM Status (OEM Reserved): [OEM State = 0000h]
79: ECC Corr Err (Memory): [Unknown]
80: ECC Uncorr Err (Memory): [Unknown]
81: I/O Channel Chk (Critical Interrupt): [Unknown]
82: PCI Parity Err (Critical Interrupt): [Unknown]
83: PCI System Err (Critical Interrupt): [Unknown]
84: SBE Log Disabled (Event Logging Disabled): [Unknown]
85: Logging Disabled (Event Logging Disabled): [Unknown]
```

```
86: Unknown (System Event): [Unknown]
87: CPU Protocol Err (Processor): [Unknown]
88: CPU Bus PERR (Processor): [Unknown]
89: CPU Init Err (Processor): [Unknown]
90: CPU Machine Chk (Processor): [Unknown]
91: Memory Spared (Memory): [Unknown]
92: Memory Mirrored (Memory): [Unknown]
93: Memory RAID (Memory): [Unknown]
94: Memory Added (Memory): [Unknown]
95: Memory Removed (Memory): [Unknown]
96: Memory Cfg Err (Memory): [Unknown]
97: Mem Redun Gain (Memory): [Unknown]
98: PCIE Fatal Err (Critical Interrupt): [Unknown]
99: Chipset Err (Critical Interrupt): [Unknown]
100: Err Reg Pointer (OEM Reserved): [Unknown]
101: Mem ECC Warning (Memory): [Unknown]
102: Mem CRC Err (Memory): [Unknown]
103: USB Over-current (Memory): [Unknown]
104: POST Err (System Firmware Progress): [Unknown]
105: Hdwr version err (Version Change): [Unknown]
106: Mem Overtemp (Memory): [Unknown]
107: Mem Fatal SB CRC (Memory): [Unknown]
108: Mem Fatal NB CRC (Memory): [Unknown]
109: OS Watchdog Time (Watchdog 1): [Unknown]
110: Non Fatal PCI Er (OEM Reserved): [Unknown]
111: Fatal IO Error (OEM Reserved): [Unknown]
112: MSR Info Log (OEM Reserved): [Unknown]
113: Temp (Temperature): NA (NA/NA): [NA]
114: Temp (Temperature): NA (3.00/47.00): [NA]
115: Temp (Temperature): NA (3.00/47.00): [NA]
116: Current (Current): 1.40 A (NA/NA): [OK]
117: Current (Current): NA (NA/NA): [Unknown]
118: Voltage (Voltage): 220.00 V (NA/NA): [OK]
119: Voltage (Voltage): NA (NA/NA): [Unknown]
120: System Level (Current): 329.00 W (NA/966.00): [OK]
121: Power Optimized (OEM Reserved): [Unrecognized State]
123: ROMB Battery (Battery): [OK]
125: vFlash (Module/Board): [OEM State = 0000h]
```

ipmi-locate

```
# ipmi-locate
```

```
Probing KCS device using DMIDECODE... done
IPMI Version: 2.0
IPMI locate driver: DMIDECODE
IPMI interface: KCS
BMC driver device:
BMC I/O base address: 0xCA8
Register spacing: 4

Probing SMIC device using DMIDECODE... FAILED

Probing BT device using DMIDECODE... FAILED

Probing SSIF device using DMIDECODE... FAILED

Probing KCS device using SMBIOS... done
IPMI Version: 2.0
IPMI locate driver: SMBIOS
IPMI interface: KCS
BMC driver device:
BMC I/O base address: 0xCA8
Register spacing: 4

Probing SMIC device using SMBIOS... FAILED

Probing BT device using SMBIOS... FAILED

Probing SSIF device using SMBIOS... FAILED

Probing KCS device using ACPI... FAILED

Probing SMIC device using ACPI... FAILED

Probing BT device using ACPI... FAILED

Probing SSIF device using ACPI... FAILED

Probing KCS device using PCI... FAILED

Probing SMIC device using PCI... FAILED

Probing BT device using PCI... FAILED

Probing SSIF device using PCI... FAILED

KCS device default values:
```

```
IPMI Version: 1.5
IPMI locate driver: DEFAULT
IPMI interface: KCS
BMC driver device:
BMC I/O base address: 0xCA2
Register spacing: 1

SMIC device default values:
IPMI Version: 1.5
IPMI locate driver: DEFAULT
IPMI interface: SMIC
BMC driver device:
BMC I/O base address: 0xCA9
Register spacing: 1

BT device default values:
SSIF device default values:
IPMI Version: 1.5
IPMI locate driver: DEFAULT
IPMI interface: SSIF
BMC driver device: /dev/i2c-0
BMC SMBUS slave address: 0x42
Register spacing: 1
```

15.3. ipmitool - utility for controlling IPMI-enabled devices

ipmitool

ubuntu

确定硬件是否支持 IPMI

```


neo@monitor:~$ sudo dmidecode |grep -C 5 IPMI
[sudo] password for neo:
Handle 0x2000, DMI type 32, 11 bytes
System Boot Information
        Status: No errors detected

Handle 0x2600, DMI type 38, 18 bytes
IPMI Device Information
```

```
        Interface Type: KCS (Keyboard Control Style)
        Specification Version: 2.0
        I2C Slave Address: 0x10
        NV Storage Device: Not Present
        Base Address: 0x0000000000000CA8 (I/O)
```

```
sudo apt-get install openipmi

sudo apt-get install ipmitool

sudo mkdir -p /var/lock/subsys/ipmi

$ sudo /etc/init.d/openipmi start
 * Starting ipmi drivers                  [ OK ]
```

CentOS

```
# yum search ipmi
========================================== Matched: ipmi
===========================================
OpenIPMI.x86_64 : OpenIPMI (Intelligent Platform Management
Interface) library and tools
OpenIPMI-devel.i386 : The development environment for the
OpenIPMI project.
OpenIPMI-devel.x86_64 : The development environment for the
OpenIPMI project.
OpenIPMI-gui.x86_64 : IPMI graphical user interface tool
OpenIPMI-libs.i386 : The OpenIPMI runtime libraries
OpenIPMI-libs.x86_64 : The OpenIPMI runtime libraries
OpenIPMI-perl.x86_64 : OpenIPMI Perl language bindings
OpenIPMI-python.x86_64 : OpenIPMI Python language bindings
OpenIPMI-tools.x86_64 : OpenIPMI utilities and scripts from
ipmitool
collectd-ipmi.x86_64 : IPMI module for collectd
freeipmi.i386 : FreeIPMI
freeipmi.x86_64 : FreeIPMI
freeipmi-bmc-watchdog.x86_64 : FreeIPMI BMC watchdog
freeipmi-devel.i386 : Development package for FreeIPMI
```

```
freeipmi-devel.x86_64 : Development package for FreeIPMI
freeipmi-ipmidetectd.x86_64 : IPMI node detection monitoring
daemon
openhpi.i386 : openhpi Hardware Platform Interface (HPI)
library and tools
openhpi.x86_64 : openhpi Hardware Platform Interface (HPI)
library and tools
ripmime.x86_64 : Extract attachments out of a MIME encoded
email packages
watchdog.x86_64 : Software and/or Hardware watchdog daemon

# yum install OpenIPMI OpenIPMI-tools -y
```

sensor

```
# ipmitool -I open sensor list
```

ipmitool shell

```
# ipmitool shell
```

mc info

```


ipmitool> mc info
Device ID                 : 32
Device Revision           : 0
Firmware Revision         : 1.54
IPMI Version              : 2.0
Manufacturer ID           : 674
Manufacturer Name         : DELL Inc
Product ID                : 256 (0x0100)
Product Name              : Unknown (0x100)
Device Available          : yes
Provides Device SDRs      : yes
Additional Device Support :
    Sensor Device
    SDR Repository Device
    SEL Device
    FRU Inventory Device
    IPMB Event Receiver
    Bridge
    Chassis Device
Aux Firmware Rev Info     :
    0x00
    0x0f
    0x00
    0x00


ipmitool> lan print 1
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 172.16.1.132
Subnet Mask             : 255.255.255.0
MAC Address             : 84:2b:2b:fd:e2:51
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00
TOS=0x10
Default Gateway IP      : 172.16.1.254
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
```

ipmitool 访问远程主机

```
# ipmitool -H 172.16.1.155 -U root -P 123456 lan print 1
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 172.16.1.15
Subnet Mask             : 255.255.255.0
MAC Address             : 84:2b:2b:fc:fb:cc
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00
TOS=0x10
Default Gateway IP      : 172.16.1.254
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
```

Get chassis status and set power state

```
# ipmitool -I open chassis
Chassis Commands:  status, power, identify, policy,
restart_cause, poh, bootdev, bootparam, selftest

# ipmitool -I open chassis status
System Power         : on
Power Overload       : false
Power Interlock      : inactive
Main Power Fault     : false
Power Control Fault  : false
Power Restore Policy : previous
Last Power Event     :
Chassis Intrusion    : inactive
Front-Panel Lockout  : inactive
Drive Fault          : false
Cooling/Fan Fault    : false
Sleep Button Disable : not allowed
Diag Button Disable  : allowed
Reset Button Disable : not allowed
Power Button Disable : allowed
Sleep Button Disabled: false
Diag Button Disabled : true
Reset Button Disabled: false
Power Button Disabled: false
```

Configure Management Controller

Management Controller status and global enables

```
# ipmitool -I open mc
MC Commands:
  reset <warm|cold>
  guid
  info
  watchdog <get|reset|off>
  selftest
  getenables
  setenables <option=on|off> ...
    recv_msg_intr         Receive Message Queue Interrupt
    event_msg_intr        Event Message Buffer Full Interrupt
    event_msg             Event Message Buffer
    system_event_log      System Event Logging
    oem0                      OEM 0
    oem1                      OEM 1
    oem2                      OEM 2
```

Configure LAN Channels

```
ipmitool -I open lan print 1                                 显示BMC
通道的信息，如果不知道BMC使用的是哪个通道，请使用下面的命令确认：
ipmitool -I open channel info 1
ipmitool -I open lan set 1 ipsrc static                      设置本地
BMC地址为静态，才能设置IP
ipmitool -I open lan set 1 ipaddr 172.16.0.2                 设置本地
BMC的IP地址
ipmitool -I open lan set 1 netmask 255.255.255.0             子网掩
码，别忘了设
ipmitool -I open lan set 1 defgw ipaddr 172.16.0.254         网关，可
设可不设，不过一定要确保监控它的机器位于同一路由
```

Configure Management Controller users

```
ipmitool user list 1                                 查看BMC的用户列表
ipmitool user set name 1 username           对BMC的1号用户设置用户名
username
ipmitool user set password 1 123456  对BMC的1号用户设置密码123456
```

Configure Management Controller channels

```
# ipmitool -I open channel info 1
Channel 0x1 info:
  Channel Medium Type     : 802.3 LAN
  Channel Protocol Type : IPMB-1.0
  Session Support         : multi-session
  Active Session Count    : 0
  Protocol Vendor ID      : 7154
  Volatile(active) Settings
    Alerting            : disabled
    Per-message Auth    : disabled
    User Level Auth     : enabled
    Access Mode         : always available
  Non-Volatile Settings
    Alerting            : disabled
    Per-message Auth    : disabled
    User Level Auth     : enabled
    Access Mode         : always available
```

Example for iDRAC

http://support.dell.com/support/edocs/software/smbmcmu/bmcmu_4_0/cs/ug/bmcugc0d.htm#wp1067804

更改IP地址,子网掩码与网关

查看IP，子网掩码与网关

```
# ipmitool -I open lan print 1
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 172.16.5.23
Subnet Mask             : 255.255.255.0
MAC Address             : 18:03:73:f5:ee:82
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00
TOS=0x10
Default Gateway IP      : 172.16.5.254
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
```

```
802.1q VLAN Priority     : 0
RMCP+ Cipher Suites      : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max    : aaaaaaaaaaaaaaa
                         :     X=Cipher Suite Unused
                         :     c=CALLBACK
                         :     u=USER
                         :     o=OPERATOR
                         :     a=ADMIN
                         :     O=OEM
```

设置IP，子网掩码与网关

```
/usr/bin/ipmitool -I open lan set 1 ipaddr 172.16.8.200
/usr/bin/ipmitool -I open lan set 1 netmask 255.255.255.0
/usr/bin/ipmitool -I open lan set 1 defgw ipaddr 172.16.8.254
/usr/bin/ipmitool -I open lan set 1 access on
```

更改 iDRAC LCD 显示屏

```
# ipmitool delloem lcd set mode userdefined test
# ipmitool delloem lcd info
LCD info
    Setting: User defined
    Text:    test
```

更改 iDRAC 密码

```
# ipmitool user list 2
ID  Name             Callin  Link Auth  IPMI Msg   Channel Priv
Limit
2   root             true    true       true
ADMINISTRATOR
# ipmitool user set password 2 "mypasswd"
```

关机/开机

```
服务器关机
#ipmitool -I lan -U root -P secpass -H 10.10.0.5 power off

服务器开机
#ipmitool -I lan -U root -P secpass -H 10.10.0.5 power on

服务器 reset
#ipmitool -I lan -U root -P secpass -H 10.10.0.5 power reset
```

启动列表

```
ipmitool -I lan -H 10.10.0.5 -U ADMIN -P ADMIN chassis bootdev pxe
```

16. JVM

```
jps:
http://java.sun.com/j2se/1.5.0/docs/tooldocs/share/jps.html
jstat:
http://java.sun.com/j2se/1.5.0/docs/tooldocs/share/jstat.html
jmap:
http://java.sun.com/j2se/1.5.0/docs/tooldocs/share/jmap.html
```

16.1. jconsole

jconsole:
http://java.sun.com/j2se/1.5.0/docs/guide/management/jconsole.html

```
java -jar -Djava.rmi.server.hostname=192.168.0.1 -
Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=911 -
Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false netkiller-1.0-
SNAPSHOT.jar
```

如果是云主机，配置java.rmi.server.hostname=192.168.0.1为内网IP地址，这样只能从内网监控JVM。如果仅仅是开发调试可以不用设置java.rmi.server.hostname

```
java -jar -Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=911 -
Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false netkiller-1.0-
SNAPSHOT.jar
```

启动 jconsole

```
jconsole localhost:911
```

16.2. jps - Java Virtual Machine Process Status Tool

```
# jps
31362 Jps
15888 Bootstrap
```

16.3. jinfo - Configuration Info

观察运行中的java程序的运行环境参数：参数包括Java System属性，各种.properties文件配置参数和JVM命令行参数

```
# jinfo $(pgrep java)
Attaching to process ID 15888, please wait...
Debugger attached successfully.
Server compiler detected.
JVM version is 24.72-b04
Java System Properties:

java.vendor = Oracle Corporation
sun.java.launcher = SUN_STANDARD
catalina.base = /srv/apache-tomcat
sun.management.compiler = HotSpot 64-Bit Tiered Compilers
catalina.useNaming = true
captcha.times = 5
os.name = Linux

...
...

java.vm.name = Java HotSpot(TM) 64-Bit Server VM
cpool.maxIdleTime = 7200
```

```
file.encoding = UTF-8
java.specification.version = 1.7

VM Flags:

-Djava.util.logging.config.file=/srv/apache-
tomcat/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManage
r -Xms512m -Xmx8192m -XX:PermSize=64M -XX:MaxPermSize=512m -
Djava.endorsed.dirs=/srv/apache-tomcat/endorsed -
Dcatalina.base=/srv/apache-tomcat -Dcatalina.home=/srv/apache-
tomcat -Djava.io.tmpdir=/srv/apache-tomcat/temp
```

实例二

```
# jinfo $(jps | grep Bootstrap | cut -d " " -f1)
Attaching to process ID 15888, please wait...
Debugger attached successfully.
Server compiler detected.
JVM version is 24.72-b04
Java System Properties:

java.vendor = Oracle Corporation
sun.java.launcher = SUN_STANDARD
catalina.base = /srv/apache-tomcat
sun.management.compiler = HotSpot 64-Bit Tiered Compilers
catalina.useNaming = true
captcha.times = 5
os.name = Linux
```

16.4. jstat - Java Virtual Machine Statistics Monitoring Tool

```
# jstat -class 15888 1000 10
Loaded Bytes Unloaded Bytes Time
17409 34782.5 231 339.0 13.21
17409 34782.5 231 339.0 13.21
17409 34782.5 231 339.0 13.21
17409 34782.5 231 339.0 13.21
17409 34782.5 231 339.0 13.21
```

```
17409 34782.5 231 339.0 13.21
```

```
# jstat -gc 15888 1000 10
S0C S1C S0U S1U EC EU OC OU PC PU YGC YGCT FGC FGCT GCT
13824.0 13824.0 1204.1 0.0 2766848.0 2327059.3 349696.0
318073.6 229888.0 101912.6 288 4.895 2 1.055 5.949
13824.0 13824.0 1204.1 0.0 2766848.0 2327059.3 349696.0
318073.6 229888.0 101912.6 288 4.895 2 1.055 5.949
13824.0 13824.0 1204.1 0.0 2766848.0 2327059.3 349696.0
318073.6 229888.0 101912.6 288 4.895 2 1.055 5.949
13824.0 13824.0 1204.1 0.0 2766848.0 2327059.3 349696.0
318073.6 229888.0 101912.6 288 4.895 2 1.055 5.949
```

```
# jstat -gcutil 15888
S0 S1 E O P YGC YGCT FGC FGCT GCT
8.71 0.00 84.12 90.96 44.33 288 4.895 2 1.055 5.949
```

```
# jstat -compiler 15888
Compiled Failed Invalid Time FailedType FailedMethod
2987 0 0 59.55 0
```

```
# jstat -gccapacity 15888
NGCMN NGCMX NGC S0C S1C EC OGCMN OGCMX OGC OC PGCMN PGCMX PGC
PC YGC FGC
175104.0 2796544.0 2794496.0 13824.0 13824.0 2766848.0 349696.0
5592064.0 349696.0 349696.0 65536.0 524288.0 229888.0 229888.0
288 2
```

```
# jstat -gcnew 15888
S0C S1C S0U S1U TT MTT DSS EC EU YGC YGCT
13824.0 13824.0 1204.1 0.0 1 15 13824.0 2766848.0 2327429.8 288
4.895
```

```
# jstat -gcnewcapacity 15888
NGCMN NGCMX NGC S0CMX S0C S1CMX S1C ECMX EC YGC FGC
175104.0 2796544.0 2794496.0 931840.0 13824.0 931840.0 13824.0
2795520.0 2766848.0 288 2
```

```
# jstat -gcold 15888
PC PU OC OU YGC FGC FGCT GCT
229888.0 101912.6 349696.0 318073.6 288 2 1.055 5.949
```

```
# jstat -gcoldcapacity 15888
OGCMN OGCMX OGC OC YGC FGC FGCT GCT
349696.0 5592064.0 349696.0 349696.0 288 2 1.055 5.949
```

每 1000 毫秒打印一次，一共打印 5 次，还可以加上 -h3 每三行显示一下标题。

```
# jstat -printcompilation -h3 15888
Compiled Size Type Method
2987 91 1 org/apache/catalina/connector/Request isAlpha
```

16.5. jHiccup

第 6 章 Logs 分析

1. log

1.1. logwatch

logwatch - log analyser with nice output written in Perl

http://www.logwatch.org/

过程 6.1. logwatch 安装步骤:

1. Install

   Ubuntu 7.10

   ```
   netkiller@shenzhen:/etc/webmin$ apt-cache search logwatch
   fwlogwatch - Firewall log analyzer
   logwatch - log analyser with nice output written in Perl
   ```

   apt-get install

   ```
   # apt-get install logwatch
   ```

   the logwatch has been installed, it should create a file in '/etc/cron.daily/00logwatch'.

2. config

   ```
   $ sudo cp /usr/share/logwatch/default.conf/logwatch.conf
   /etc/logwatch/conf/logwatch.conf
   ```

```
$ sudo mkdir /var/cache/logwatch
$ sudo vim /etc/logwatch/conf/logwatch.conf
```

mail to

```
# Default person to mail reports to.  Can be a local
account or a
# complete email address.
MailTo = root, openunix@163.com, other@example.com
```

To change detail level for the report

```
# The default detail level for the report.
# This can either be Low, Med, High or a number.
# Low = 0
# Med = 5
# High = 10
Detail = High
```

Crontab

```
netkiller@shenzhen:~$ cat /etc/cron.daily/00logwatch
#!/bin/bash

#Check if removed-but-not-purged
test -x /usr/share/logwatch/scripts/logwatch.pl || exit 0

#execute
/usr/sbin/logwatch
```

3. The logwatch is command, you can run it.

```
logwatch --print
```

单独查看某个服务，比如 SSH 登录信息

logwatch --service sshd --print

1.2. logcheck : Analyzes log files and sends noticeable events as email

```
# yum search logcheck | grep logcheck
Repodata is over 2 weeks old. Install yum-cron? Or run: yum
makecache fast
============================ N/S matched: logcheck
=============================
logcheck.noarch : Analyzes log files and sends noticeable
events as email
```

安装 logcheck

```
# yum install -y logcheck
```

查看 logchek 包所含文件

```
[root@173 ~]# rpm -ql logcheck
/etc/cron.d/logcheck
/etc/logcheck
/etc/logcheck/cracking.d
/etc/logcheck/cracking.d/kernel
/etc/logcheck/cracking.d/rlogind
/etc/logcheck/cracking.d/rsh
/etc/logcheck/cracking.d/smartd
/etc/logcheck/cracking.d/tftpd
/etc/logcheck/cracking.d/uucico
/etc/logcheck/ignore.d.paranoid
/etc/logcheck/ignore.d.paranoid/bind
/etc/logcheck/ignore.d.paranoid/cron
/etc/logcheck/ignore.d.paranoid/incron
/etc/logcheck/ignore.d.paranoid/logcheck
/etc/logcheck/ignore.d.paranoid/postfix
```

```
/etc/logcheck/ignore.d.paranoid/ppp
/etc/logcheck/ignore.d.paranoid/pureftp
/etc/logcheck/ignore.d.paranoid/qpopper
/etc/logcheck/ignore.d.paranoid/squid
/etc/logcheck/ignore.d.paranoid/ssh
/etc/logcheck/ignore.d.paranoid/stunnel
/etc/logcheck/ignore.d.paranoid/sysklogd
/etc/logcheck/ignore.d.paranoid/telnetd
/etc/logcheck/ignore.d.paranoid/tripwire
/etc/logcheck/ignore.d.paranoid/usb
/etc/logcheck/ignore.d.server
/etc/logcheck/ignore.d.server/NetworkManager
/etc/logcheck/ignore.d.server/acpid
/etc/logcheck/ignore.d.server/amandad
/etc/logcheck/ignore.d.server/amavisd-new
/etc/logcheck/ignore.d.server/anacron
/etc/logcheck/ignore.d.server/anon-proxy
/etc/logcheck/ignore.d.server/apache
/etc/logcheck/ignore.d.server/apcupsd
/etc/logcheck/ignore.d.server/arpwatch
/etc/logcheck/ignore.d.server/asterisk
/etc/logcheck/ignore.d.server/automount
/etc/logcheck/ignore.d.server/bind
/etc/logcheck/ignore.d.server/bluez-utils
/etc/logcheck/ignore.d.server/courier
/etc/logcheck/ignore.d.server/cpqarrayd
/etc/logcheck/ignore.d.server/cpufreqd
/etc/logcheck/ignore.d.server/cron
/etc/logcheck/ignore.d.server/cron-apt
/etc/logcheck/ignore.d.server/cups-lpd
/etc/logcheck/ignore.d.server/cvs-pserver
/etc/logcheck/ignore.d.server/cvsd
/etc/logcheck/ignore.d.server/cyrus
/etc/logcheck/ignore.d.server/dbus
/etc/logcheck/ignore.d.server/dcc
/etc/logcheck/ignore.d.server/ddclient
/etc/logcheck/ignore.d.server/dhclient
/etc/logcheck/ignore.d.server/dhcp
/etc/logcheck/ignore.d.server/dictd
/etc/logcheck/ignore.d.server/dkfilter
/etc/logcheck/ignore.d.server/dkim-filter
/etc/logcheck/ignore.d.server/dnsmasq
/etc/logcheck/ignore.d.server/dovecot
/etc/logcheck/ignore.d.server/dropbear
/etc/logcheck/ignore.d.server/dspam
```

```
/etc/logcheck/ignore.d.server/epmd
/etc/logcheck/ignore.d.server/exim4
/etc/logcheck/ignore.d.server/fcron
/etc/logcheck/ignore.d.server/ftpd
/etc/logcheck/ignore.d.server/git-daemon
/etc/logcheck/ignore.d.server/gnu-imap4d
/etc/logcheck/ignore.d.server/gps
/etc/logcheck/ignore.d.server/grinch
/etc/logcheck/ignore.d.server/horde3
/etc/logcheck/ignore.d.server/hplip
/etc/logcheck/ignore.d.server/hylafax
/etc/logcheck/ignore.d.server/ikiwiki
/etc/logcheck/ignore.d.server/imap
/etc/logcheck/ignore.d.server/imapproxy
/etc/logcheck/ignore.d.server/imp
/etc/logcheck/ignore.d.server/imp4
/etc/logcheck/ignore.d.server/innd
/etc/logcheck/ignore.d.server/ipppd
/etc/logcheck/ignore.d.server/isdnlog
/etc/logcheck/ignore.d.server/isdnutils
/etc/logcheck/ignore.d.server/jabberd
/etc/logcheck/ignore.d.server/kernel
/etc/logcheck/ignore.d.server/klogind
/etc/logcheck/ignore.d.server/krb5-kdc
/etc/logcheck/ignore.d.server/libpam-krb5
/etc/logcheck/ignore.d.server/libpam-mount
/etc/logcheck/ignore.d.server/logcheck
/etc/logcheck/ignore.d.server/login
/etc/logcheck/ignore.d.server/maradns
/etc/logcheck/ignore.d.server/mldonkey-server
/etc/logcheck/ignore.d.server/mon
/etc/logcheck/ignore.d.server/mountd
/etc/logcheck/ignore.d.server/nagios
/etc/logcheck/ignore.d.server/netconsole
/etc/logcheck/ignore.d.server/nfs
/etc/logcheck/ignore.d.server/nntpcache
/etc/logcheck/ignore.d.server/nscd
/etc/logcheck/ignore.d.server/nslcd
/etc/logcheck/ignore.d.server/openvpn
/etc/logcheck/ignore.d.server/otrs
/etc/logcheck/ignore.d.server/passwd
/etc/logcheck/ignore.d.server/pdns
/etc/logcheck/ignore.d.server/perdition
/etc/logcheck/ignore.d.server/policyd
/etc/logcheck/ignore.d.server/popa3d
```

```
/etc/logcheck/ignore.d.server/postfix
/etc/logcheck/ignore.d.server/postfix-policyd
/etc/logcheck/ignore.d.server/ppp
/etc/logcheck/ignore.d.server/pptpd
/etc/logcheck/ignore.d.server/procmail
/etc/logcheck/ignore.d.server/proftpd
/etc/logcheck/ignore.d.server/puppetd
/etc/logcheck/ignore.d.server/pure-ftpd
/etc/logcheck/ignore.d.server/pureftp
/etc/logcheck/ignore.d.server/qpopper
/etc/logcheck/ignore.d.server/rbldnsd
/etc/logcheck/ignore.d.server/rpc_statd
/etc/logcheck/ignore.d.server/rsnapshot
/etc/logcheck/ignore.d.server/rsync
/etc/logcheck/ignore.d.server/sa-exim
/etc/logcheck/ignore.d.server/samba
/etc/logcheck/ignore.d.server/saned
/etc/logcheck/ignore.d.server/sasl2-bin
/etc/logcheck/ignore.d.server/saslauthd
/etc/logcheck/ignore.d.server/schroot
/etc/logcheck/ignore.d.server/scponly
/etc/logcheck/ignore.d.server/slapd
/etc/logcheck/ignore.d.server/smartd
/etc/logcheck/ignore.d.server/smbd_audit
/etc/logcheck/ignore.d.server/smokeping
/etc/logcheck/ignore.d.server/snmpd
/etc/logcheck/ignore.d.server/snort
/etc/logcheck/ignore.d.server/spamc
/etc/logcheck/ignore.d.server/spamd
/etc/logcheck/ignore.d.server/squid
/etc/logcheck/ignore.d.server/ssh
/etc/logcheck/ignore.d.server/stunnel
/etc/logcheck/ignore.d.server/su
/etc/logcheck/ignore.d.server/sudo
/etc/logcheck/ignore.d.server/sympa
/etc/logcheck/ignore.d.server/syslogd
/etc/logcheck/ignore.d.server/systemd
/etc/logcheck/ignore.d.server/teapop
/etc/logcheck/ignore.d.server/telnetd
/etc/logcheck/ignore.d.server/tftpd
/etc/logcheck/ignore.d.server/thy
/etc/logcheck/ignore.d.server/ucd-snmp
/etc/logcheck/ignore.d.server/upsd
/etc/logcheck/ignore.d.server/uptimed
/etc/logcheck/ignore.d.server/userv
```

```
/etc/logcheck/ignore.d.server/vsftpd
/etc/logcheck/ignore.d.server/watchdog
/etc/logcheck/ignore.d.server/wu-ftpd
/etc/logcheck/ignore.d.server/xinetd
/etc/logcheck/ignore.d.workstation
/etc/logcheck/ignore.d.workstation/automount
/etc/logcheck/ignore.d.workstation/bind
/etc/logcheck/ignore.d.workstation/bluetooth-alsa
/etc/logcheck/ignore.d.workstation/bluez-utils
/etc/logcheck/ignore.d.workstation/bonobo
/etc/logcheck/ignore.d.workstation/dhcpcd
/etc/logcheck/ignore.d.workstation/francine
/etc/logcheck/ignore.d.workstation/gconf
/etc/logcheck/ignore.d.workstation/gdm
/etc/logcheck/ignore.d.workstation/hald
/etc/logcheck/ignore.d.workstation/hcid
/etc/logcheck/ignore.d.workstation/ifplugd
/etc/logcheck/ignore.d.workstation/ippl
/etc/logcheck/ignore.d.workstation/kdm
/etc/logcheck/ignore.d.workstation/kernel
/etc/logcheck/ignore.d.workstation/laptop-mode-tools
/etc/logcheck/ignore.d.workstation/libmtp-runtime
/etc/logcheck/ignore.d.workstation/libpam-gnome-keyring
/etc/logcheck/ignore.d.workstation/logcheck
/etc/logcheck/ignore.d.workstation/login
/etc/logcheck/ignore.d.workstation/net-acct
/etc/logcheck/ignore.d.workstation/nntpcache
/etc/logcheck/ignore.d.workstation/polypaudio
/etc/logcheck/ignore.d.workstation/postfix
/etc/logcheck/ignore.d.workstation/ppp
/etc/logcheck/ignore.d.workstation/proftpd
/etc/logcheck/ignore.d.workstation/pump
/etc/logcheck/ignore.d.workstation/sendfile
/etc/logcheck/ignore.d.workstation/slim
/etc/logcheck/ignore.d.workstation/squid
/etc/logcheck/ignore.d.workstation/udev
/etc/logcheck/ignore.d.workstation/wdm
/etc/logcheck/ignore.d.workstation/winbind
/etc/logcheck/ignore.d.workstation/wpasupplicant
/etc/logcheck/ignore.d.workstation/xdm
/etc/logcheck/ignore.d.workstation/xlockmore
/etc/logcheck/logcheck.conf
/etc/logcheck/logcheck.logfiles
/etc/logcheck/violations.d
/etc/logcheck/violations.d/kernel
```

```
/etc/logcheck/violations.d/smartd
/etc/logcheck/violations.d/su
/etc/logcheck/violations.d/sudo
/etc/logcheck/violations.ignore.d
/etc/logcheck/violations.ignore.d/logcheck-su
/etc/logcheck/violations.ignore.d/logcheck-sudo
/etc/tmpfiles.d/logcheck.conf
/usr/bin/logcheck-test
/usr/sbin/logcheck
/usr/sbin/logtail
/usr/sbin/logtail2
/usr/share/doc/logcheck-1.3.15
/usr/share/doc/logcheck-1.3.15/LICENSE
/usr/share/doc/logcheck-1.3.15/README-psionic
/usr/share/doc/logcheck-1.3.15/README.Maintainer
/usr/share/doc/logcheck-1.3.15/README.how.to.interpret
/usr/share/doc/logcheck-1.3.15/README.keywords
/usr/share/doc/logcheck-1.3.15/README.logcheck
/usr/share/doc/logcheck-1.3.15/README.logcheck-database
/usr/share/doc/logcheck-1.3.15/README.logtail
/usr/share/doc/logcheck-1.3.15/logcheck-test.1
/usr/share/doc/logcheck-1.3.15/logcheck.sgml
/usr/share/doc/logcheck-1.3.15/logtail.8
/usr/share/doc/logcheck-1.3.15/logtail2.8
/usr/share/doc/logcheck-1.3.15/tools
/usr/share/doc/logcheck-1.3.15/tools/log-summary-ssh
/usr/share/logtail
/usr/share/logtail/detectrotate
/usr/share/logtail/detectrotate/10-savelog.dtr
/usr/share/logtail/detectrotate/20-logrotate.dtr
/usr/share/logtail/detectrotate/30-logrotate-dateext.dtr
/usr/share/man/man1/logcheck-test.1.gz
/usr/share/man/man8/logcheck.8.gz
/usr/share/man/man8/logtail.8.gz
/usr/share/man/man8/logtail2.8.gz
/var/lib/logcheck
/var/lock/logcheck
```

1.3. nulog

例 6.1. config.php

2. Web

2.1. Apache Log

```
1、查看当天有多少个IP访问：
awk '{print $1}' log_file|sort|uniq|wc -l

2、查看某一个页面被访问的次数：
grep "/index.php" log_file | wc -l

3、查看每一个IP访问了多少个页面：
awk '{++S[$1]} END {for (a in S) print a,S[a]}' log_file

4、将每个IP访问的页面数进行从小到大排序：
awk '{++S[$1]} END {for (a in S) print S[a],a}' log_file | sort
-n

5、查看某一个IP访问了哪些页面：
grep ^111.111.111.111 log_file| awk '{print $1,$7}'

6、去掉搜索引擎统计当天的页面：
awk '{print $12,$1}' log_file | grep ^\"Mozilla | awk '{print
$2}' |sort | uniq | wc -l

7、查看2009年6月21日14时这一个小时内有多少IP访问：
awk '{print $4,$1}' log_file | grep 21/Jun/2009:14 | awk
'{print $2}'| sort | uniq | wc -l
```

删除日志

删除一个月前的日志

```
rm -f /www/logs/access.log.$(date -d '-1 month' +'%Y-%m')*
```

统计爬虫

```
grep -E 'Googlebot|Baiduspider'
/www/logs/www.example.com/access.2011-02-23.log | awk '{ print
$1 }' | sort | uniq
```

统计浏览器

```
cat /www/logs/example.com/access.2010-09-20.log | grep -v -E
'MSIE|Firefox|Chrome|Opera|Safari|Gecko|Maxthon' | sort | uniq
-c | sort -r -n | head -n 100
```

IP 统计

```
# grep '22/May/2012' /tmp/myid.access.log | awk '{print $1}' |
awk -F'.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -
r -n | head -n 10
   2206 219.136.134.13
   1497 182.34.15.248
   1431 211.140.143.100
   1431 119.145.149.106
   1427 61.183.15.179
   1427 218.6.8.189
   1422 124.232.150.171
   1421 106.187.47.224
   1420 61.160.220.252
   1418 114.80.201.18
```

统计网段

```
# cat /www/logs/www/access.2010-09-20.log | awk '{print $1}' |
awk -F'.' '{print $1"."$2"."$3".0"}' | sort | uniq -c | sort -r
-n | head -n 200
```

压缩文件处理

```
zcat www.example.com.access.log-20130627.gz | grep
'/xml/data.json' | awk '{print $1}' | awk -F'.' '{print
$1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n | head -n 20
```

统计域名

```
# cat  /www/logs/access.2011-07-27.log |awk '{print
$2}'|sort|uniq -c|sort -rn|more
```

HTTP Status

```
# cat  /www/logs/access.2011-07-27.log |awk '{print
$9}'|sort|uniq -c|sort -rn|more
5056585 304
1125579 200
   7602 400
      5 301
```

URL 统计

```
cat  /www/logs/access.2011-07-27.log |awk '{print
$7}'|sort|uniq -c|sort -rn|more
```

文件流量统计

```
cat /www/logs/access.2011-08-03.log |awk
'{sum[$7]+=$10}END{for(i in sum){print sum[i],i}}'|sort -
rn|more

grep ' 200 ' /www/logs/access.2011-08-03.log |awk
'{sum[$7]+=$10}END{for(i in sum){print sum[i],i}}'|sort -
```

```
rn|more
```

URL访问量统计

```
# cat www.access.log | awk '{print $7}' | egrep '\?|&' | sort |
uniq -c | sort -rn | more
```

脚本运行速度

查出运行速度最慢的脚本

```
grep -v 0$ access.2010-11-05.log | awk -F '\" ' '{print $4" "
$1}' web.log | awk '{print $1" "$8}' | sort -n -k 1 -r | uniq >
/tmp/slow_url.txt
```

IP, URL 抽取

```
# tail -f /www/logs/www.365wine.com/access.2012-01-04.log |
grep '/test.html' | awk '{print $1" "$7}'
```

2.2. awstats

http://sourceforge.net/projects/awstats/

1. install

```
sudo apt-get install awstats
```

2. configure

   sudo vim /etc/awstats/awstats.conf or awstats.conf.local

```
$ sudo vim /etc/awstats/awstats.conf.local

LogFile="/home/netkiller/logs/access_log"
SiteDomain="netkiller.8800.org"
```

   or

```
# cd /usr/share/doc/awstats/examples/
#/usr/share/doc/awstats/examples$ perl awstats_configure.pl
```

3. apache

```
sudo cp /usr/share/doc/awstats/examples/apache.conf
/etc/apache2/conf.d/awstats.conf
```

4. how do I test awstats.

   http://netkiller.8800.org/awstats/awstats.pl

5. Generating the First Stats

```
sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl
-update -config=netkiller.8800.org
```

6. Automatising the stats generation using Cron

   If we check the file installed by awstats and search for the word cron using the following command line:

```
$ dpkg -L awstats | grep cron
/etc/cron.d
/etc/cron.d/awstats
```

sudo vim /etc/cron.d/awstats

```
0,10,20,30,40,50 * * * * www-data [ -x /usr/lib/cgi-
bin/awstats.pl -a -f /etc/awstats/awstats.conf -a -r
/home/netkiller/logs/access.log ] && /usr/lib/cgi-
bin/awstats.pl -config=netkiller.8800.org -update
>/dev/null
```

7. web 测试

   http://netkiller.8800.org/awstats/awstats.pl

   http://netkiller.8800.org/awstats/awstats.pl?config=other.8800.org

语言

```
awstats.pl -update -config=sitename -lang=cn
```

输出HTML文档

```
perl awstats.pl -config=www.example.com -output -staticlinks -
lang=cn > awstats.example.html
```

多站点配置

```
$ sudo gunzip
/usr/share/doc/awstats/examples/awstats.model.conf.gz

$ sudo cp /usr/share/doc/awstats/examples/awstats.model.conf
/etc/awstats/awstats.www.example.com.conf
$ sudo cp /usr/share/doc/awstats/examples/awstats.model.conf
/etc/awstats/awstats.www.other.com.conf
```

```
neo@monitor:/etc/awstats$ vim awstats.www.example.com.conf
LogFile = /opt/logs/21/access.log
SiteDomain="www.example.com"

neo@monitor:/etc/awstats$ vim awstats.www.other.com.conf
LogFile = /opt/logs/22/access.log
SiteDomain="www.other.com"
```

```
$ sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl -
update -config=www.example.com
$ sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl -
update -config=www.other.com
```

```
http://localhost/cgi-bin/awstats.pl?config=www.example.com
http://localhost/cgi-bin/awstats.pl?config=www.other.com
```

批量生成

```
awstats_updateall.pl now -awstatsprog=/usr/lib/cgi-
bin/awstats.pl -configdir=/etc/awstats/
```

合并日志

/usr/share/doc/awstats/examples/logresolvemerge.pl

```
$ vim awstats.www.example.com.conf
LogFile="/usr/share/doc/awstats/examples/logresolvemerge.pl
/var/log/*/access_log.* |"
LogFile="/usr/share/doc/awstats/examples/logresolvemerge.pl
/mnt/*/logs/www/access.%YYYY-24-%MM-24-%DD-24.log |"
```

```
sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl -
update -config=www.examples.com
```

http://localhost/cgi-bin/awstats.pl?config=www.example.com

```
$ grep -v "^#" awstats.www.example.com.conf | sed /^$/d
LogFile="/usr/share/doc/awstats/examples/logresolvemerge.pl
/mnt/*/logs/www/access.%YYYY-24-%MM-24-%DD-24.log |"
LogType=W
LogFormat=1
LogSeparator=" "
SiteDomain="www.example.com"
HostAliases="localhost 127.0.0.1 REGEX[myserver\.com$]"
DNSLookup=2
DirData="."
DirCgi="/cgi-bin"
DirIcons="/icon"
AllowToUpdateStatsFromBrowser=0
AllowFullYearView=2
EnableLockForUpdate=0
DNSStaticCacheFile="dnscache.txt"
DNSLastUpdateCacheFile="dnscachelastupdate.txt"
SkipDNSLookupFor=""
AllowAccessFromWebToAuthenticatedUsersOnly=0
AllowAccessFromWebToFollowingAuthenticatedUsers=""
AllowAccessFromWebToFollowingIPAddresses=""
CreateDirDataIfNotExists=0
BuildHistoryFormat=text
BuildReportFormat=html
SaveDatabaseFilesWithPermissionsForEveryone=0
PurgeLogFile=0
```

```
ArchiveLogRecords=0
KeepBackupOfHistoricFiles=0
DefaultFile="index.html"
SkipHosts=""
SkipUserAgents=""
SkipFiles=""
SkipReferrersBlackList=""
OnlyHosts=""
OnlyUserAgents=""
OnlyUsers=""
OnlyFiles=""
NotPageList="css js class gif jpg jpeg png bmp ico rss xml swf"
ValidHTTPCodes="200 304"
ValidSMTPCodes="1 250"
AuthenticatedUsersNotCaseSensitive=0
URLNotCaseSensitive=0
URLWithAnchor=0
URLQuerySeparators="?;"
URLWithQuery=0
URLWithQueryWithOnlyFollowingParameters=""
URLWithQueryWithoutFollowingParameters=""
URLReferrerWithQuery=0
WarningMessages=1
ErrorMessages=""
DebugMessages=0
NbOfLinesForCorruptedLog=50
WrapperScript=""
DecodeUA=0
MiscTrackerUrl="/js/awstats_misc_tracker.js"
LevelForBrowsersDetection=2         # 0 disables Browsers
detection.
                                    # 2 reduces AWStats speed
by 2%
                                    # allphones reduces AWStats
speed by 5%
LevelForOSDetection=2               # 0 disables OS detection.
                                    # 2 reduces AWStats speed
by 3%
LevelForRefererAnalyze=2            # 0 disables Origin
detection.
                                    # 2 reduces AWStats speed
by 14%
LevelForRobotsDetection=2           # 0 disables Robots
detection.
                                    # 2 reduces AWStats speed
```

```
by 2.5%
LevelForSearchEnginesDetection=2    # 0 disables Search engines
detection.
                                    # 2 reduces AWStats speed
by 9%
LevelForKeywordsDetection=2         # 0 disables
Keyphrases/Keywords detection.
                                    # 2 reduces AWStats speed
by 1%
LevelForFileTypesDetection=2        # 0 disables File types
detection.
                                    # 2 reduces AWStats speed
by 1%
LevelForWormsDetection=0            # 0 disables Worms
detection.
                                    # 2 reduces AWStats speed
by 15%
UseFramesWhenCGI=1
DetailedReportsOnNewWindows=1
Expires=0
MaxRowsInHTMLOutput=1000
Lang="auto"
DirLang="./lang"
ShowMenu=1
ShowSummary=UVPHB
ShowMonthStats=UVPHB
ShowDaysOfMonthStats=VPHB
ShowDaysOfWeekStats=PHB
ShowHoursStats=PHB
ShowDomainsStats=PHB
ShowHostsStats=PHBL
ShowAuthenticatedUsers=0
ShowRobotsStats=HBL
ShowWormsStats=0
ShowEMailSenders=0
ShowEMailReceivers=0
ShowSessionsStats=1
ShowPagesStats=PBEX
ShowFileTypesStats=HB
ShowFileSizesStats=0
ShowOSStats=1
ShowBrowsersStats=1
ShowScreenSizeStats=0
ShowOriginStats=PH
ShowKeyphrasesStats=1
```

```
ShowKeywordsStats=1
ShowMiscStats=a
ShowHTTPErrorsStats=1
ShowSMTPErrorsStats=0
ShowClusterStats=0
AddDataArrayMonthStats=1
AddDataArrayShowDaysOfMonthStats=1
AddDataArrayShowDaysOfWeekStats=1
AddDataArrayShowHoursStats=1
IncludeInternalLinksInOriginSection=0
MaxNbOfDomain = 10
MinHitDomain  = 1
MaxNbOfHostsShown = 10
MinHitHost    = 1
MaxNbOfLoginShown = 10
MinHitLogin   = 1
MaxNbOfRobotShown = 10
MinHitRobot   = 1
MaxNbOfPageShown = 10
MinHitFile    = 1
MaxNbOfOsShown = 10
MinHitOs      = 1
MaxNbOfBrowsersShown = 10
MinHitBrowser = 1
MaxNbOfScreenSizesShown = 5
MinHitScreenSize = 1
MaxNbOfWindowSizesShown = 5
MinHitWindowSize = 1
MaxNbOfRefererShown = 10
MinHitRefer   = 1
MaxNbOfKeyphrasesShown = 10
MinHitKeyphrase = 1
MaxNbOfKeywordsShown = 10
MinHitKeyword = 1
MaxNbOfEMailsShown = 20
MinHitEMail   = 1
FirstDayOfWeek=1
ShowFlagLinks=""
ShowLinksOnUrl=1
UseHTTPSLinkForUrl=""
MaxLengthOfShownURL=64
HTMLHeadSection=""
HTMLEndSection=""
Logo="awstats_logo6.png"
LogoLink="http://awstats.sourceforge.net"
```

```
BarWidth   = 260
BarHeight  = 90
StyleSheet=""
color_Background="FFFFFF"               # Background color for
main page (Default = "FFFFFF")
color_TableBGTitle="CCCCDD"             # Background color for
table title (Default = "CCCCDD")
color_TableTitle="000000"               # Table title font
color (Default = "000000")
color_TableBG="CCCCDD"                  # Background color for
table (Default = "CCCCDD")
color_TableRowTitle="FFFFFF"    # Table row title font color
(Default = "FFFFFF")
color_TableBGRowTitle="ECECEC"  # Background color for row
title (Default = "ECECEC")
color_TableBorder="ECECEC"              # Table border color
(Default = "ECECEC")
color_text="000000"                             # Color of text
(Default = "000000")
color_textpercent="606060"              # Color of text for
percent values (Default = "606060")
color_titletext="000000"                # Color of text title
within colored Title Rows (Default = "000000")
color_weekend="EAEAEA"                  # Color for week-end
days (Default = "EAEAEA")
color_link="0011BB"                             # Color of HTML
links (Default = "0011BB")
color_hover="605040"                    # Color of HTML on-
mouseover links (Default = "605040")
color_u="FFAA66"                                # Background
color for number of unique visitors (Default = "FFAA66")
color_v="F4F090"                                # Background
color for number of visites (Default = "F4F090")
color_p="4477DD"                                # Background
color for number of pages (Default = "4477DD")
color_h="66DDEE"                                # Background
color for number of hits (Default = "66DDEE")
color_k="2EA495"                                # Background
color for number of bytes (Default = "2EA495")
color_s="8888DD"                                # Background
color for number of search (Default = "8888DD")
color_e="CEC2E8"                                # Background
color for number of entry pages (Default = "CEC2E8")
color_x="C1B2E2"                                # Background
color for number of exit pages (Default = "C1B2E2")
```

```
ExtraTrackedRowsLimit=500
```

Flush history file on disk (unique url reach flush limit of 5000) 优化

```
$LIMITFLUSH=50000
```

JAWStats

http://www.jawstats.com/

2.3. webalizer

What is Webalizer?

The Webalizer is a fast, free web server log file analysis program. It produces highly detailed, easily configurable usage reports in HTML format, for viewing with a standard web browser

1. install webalizer

   ```
   sudo apt-get install webalizer
   ```

2. config

   ```
   vim /etc/webalizer/webalizer.conf

   LogFile /home/netkiller/logs/access.log
   OutputDir /home/netkiller/public_html/webalizer
   ```

   rotate log

```
Incremental yes
```

3. crontab

/etc/cron.daily/webalizer

```
netkiller@shenzhen:~$ cat /etc/cron.daily/webalizer
#!/bin/sh
# /etc/cron.daily/webalizer: Webalizer daily maintenance
script
# This script was originally written by
# Remco van de Meent <remco@debian.org>
# and now, all rewrited by Jose Carlos Medeiros
<jose@psabs.com.br>

# This script just run webalizer agains all .conf files in
/etc/webalizer directory

WEBALIZER=/usr/bin/webalizer
WEBALIZER_CONFDIR=/etc/webalizer

[ -x ${WEBALIZER} ] || exit 0;
[ -d ${WEBALIZER_CONFDIR} ] || exit 0;

for i in ${WEBALIZER_CONFDIR}/*.conf; do
  # run agains a rotated or normal logfile
  LOGFILE=`awk '$1 ~ /^LogFile$/ {print $2}' $i`;

  # empty ?
  [ -s "${LOGFILE}" ] || continue;
  # readable ?
  [ -r "${LOGFILE}" ] || continue;

  # there was a output ?
  OUTDIR=`awk '$1 ~ /^OutputDir$/ {print $2}' $i`;
  #  exists something ?
  [ "${OUTDIR}" != "" ] || continue;
  # its a directory ?
  [ -d ${OUTDIR} ] || continue;
  # its writable ?
  [ -w ${OUTDIR} ] || continue;

  # Run Really quietly, exit with status code if !0
  ${WEBALIZER} -c ${i} -Q || continue;
  RET=$?;

  # Non rotated log file
  NLOGFILE=`awk '$1 ~ /^LogFile$/ {gsub(/\.[0-9]+
(\.gz)?/,""); print $2}' $i`;

  # check current log, if last log is a rotated logfile
  if [ "${LOGFILE}" != "${NLOGFILE}" ]; then
    # empty ?
    [ -s "${NLOGFILE}" ] || continue;
    # readable ?
    [ -r "${NLOGFILE}" ] || continue;

    ${WEBALIZER} -c ${i} -Q ${NLOGFILE};
    RET=$?;
  fi;
done;

# exit with webalizer's exit code
exit $RET;
```

4. initialization

```
sudo /usr/bin/webalizer
```

5. http://netkiller.8800.org/webalizer/

```
最后附上Webalizer的参数表:
可以执行webalizer –h得到所有命令行参数:
Usage: webalizer [options] [log file]
-h = 打印帮助信息
-v -V = 打印版本信息
-d = 打印附加调试信息
-F type = 日志格式类型. type= (clf | ftp | squid)
-i = 忽略历史文件
```

```
-p = 保留状态 (递增模式)
-q = 忽略消息信息
-Q = 忽略所有信息
-Y = 忽略国家图形
-G = 忽略小时统计图形
-H = 忽略小时统计信息
-L = 忽略彩色图例
-l num = 在图形中使用数字背景线
-m num = 访问超时 (seconds)
-T = 打印时间信息
-c file = 指定配置文件
-n name = 使用的主机名
-o dir = 结果输出目录
-t name = 指定报告题目上的主机名
-a name = 隐藏用户代理名称
-r name = 隐藏访问链接
-s name = 隐藏客户
-u name = 隐藏URL
-x name = 使用文件扩展名
-P name = 页面类型扩展名
-I name = index别名
-A num = 显示前几名客户类型
-C num = 显示前几名国家
-R num = 显示前几名链接
-S num = 显示前几名客户
-U num = 显示前几名URLs
-e num = 显示前几名访问页面
-E num = 显示前几名不存在的页面
-X = 隐藏个别用户
-D name = 使用dns缓存文件
-N num = DNS 进程数 (0=禁用dns)
```

手工生成

```
$ sudo webalizer -c /etc/webalizer/webalizer.conf -o
/var/www/webalizer/web2 /opt/logs/web2/www/access_log
```

分析多个文件

```
# find ./ -exec sudo webalizer -p -c
/etc/webalizer/webalizer.conf -o /var/www/webalizer/my
/mnt/logs/www/{} \;
```

批量处理历史数据

下面脚本可以批量处理历史日志,等这个脚本运行完后在crontab中加入另一个脚本。

```
for f in /mnt/logs/cdn/*.gz ; do webalizer -c
/etc/webalizer/webalizer.conf -o /var/www/webalizer/cdn/ $f ;
done
```

crontab

```
webalizer -c /etc/webalizer/webalizer.conf -o
/var/www/webalizer/cdn/ /mnt/logs/cdn/$(date -d '-1 day' +'%Y-
%m-%d').log.gz
```

多域名批量处理

```
for d in /mnt/cdn/* ; do
    htmldir=/var/www/webalizer/$(basename $d)
    mkdir -p $htmldir
    for f in $d/*.log.gz ; do webalizer -c
/etc/webalizer/webalizer.conf -o $htmldir $f ; done
done
```

crontab

```
#!/bin/bash
for d in /mnt/cdn/*;
do
```

```
    htmldir=/var/www/webalizer/$(basename $d)
    mkdir -p $htmldir
    webalizer -c /etc/webalizer/webalizer.conf -o $htmldir
$d/$(date -d '-1 day' +'%Y_%m_%d').log.gz
done
```

crontab

```
sudo webalizer  -F clf -p -t www.example.com -Q -c
/etc/webalizer/webalizer.conf -o /var/www/webalizer/example
/mnt/logs/www/access.$(date -d '-1 day' +'%Y-%m-%d').log
```

2.4. Sarg - Squid Analysis Report Generator

http://sarg.sourceforge.net/

```

```

2.5. goaccess - Fast web log analyzer and interactive viewer.

http://goaccess.prosoftcorp.com/

CentOS

```
yum install goaccess
```

Ubuntu

```
$ sudo apt-get install goaccess
```

使用方法

```
# goaccess -f access.log
```

3. Tomcat

Tomcat 日志监控主要是分析 catalina.out 文件

3.1. 截取 0-3 点区间的日志

```
egrep '^2011-08-02 0[0-3].*' sale-debug.log
```

3.2. 监控Redis

```
redis.clients.jedis.exceptions.JedisConnectionException:
java.net.SocketTimeoutException: Read timed out
```

4. Mail

4.1. pflogsumm.pl - Produce Postfix MTA logfile summary

```
# yum install -y postfix-perl-scripts
```

```
pflogsumm `ls -rt /var/log/maillog*`
pflogsumm -d today /var/log/maillog
pflogsumm -d yesterday /var/log/maillog
```

发送统计报表到邮箱

```
0 5 * * * pflogsumm -d yesterday /var/log/maillog 2>&1 | mail -s "Mail Report" postmaster@netkiller.cn
```

5. OpenSSH 日志 /var/log/secure

查询出恶意穷举密码的IP地址

```
# cat /var/log/rinetd.log | awk '{print $2}' | awk -F'.'
'{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n |
head -n 50
```

查看曾经登陆成功的IP地址

```
grep Accepted /var/log/secure | grep -oE "\b([0-9]{1,3}\.){3}
[0-9]{1,3}\b" | sort | uniq
```

5.1. 查看登陆用户

密码登陆用户

```
# grep "Accepted password" /var/log/secure

Feb 15 15:29:31 iZ623qr3xctZ sshd[25181]: Accepted password for
root from 157.90.182.21 port 29836 ssh2
Feb 15 16:24:18 iZ623qr3xctZ sshd[22150]: Accepted password for
root from 211.90.123.18 port 27553 ssh2
```

证书登陆用户

```
# grep "Accepted publickey" /var/log/secure

Feb 15 15:51:25 iZ623qr3xctZ sshd[17334]: Accepted publickey
for root from 147.90.40.39 port 42252 ssh2: RSA
ea:a9:94:d8:03:a7:39:22:05:bb:cc:f5:d8:b2:92:18
Feb 15 16:21:41 iZ623qr3xctZ sshd[19469]: Accepted publickey
for root from 147.90.40.39 port 42296 ssh2: RSA
```

```
ea:a9:94:d8:03:a7:39:22:05:bb:cc:f5:d8:b2:92:18
```

6. rinetd.log

top 50 IP Address

```
# cat /var/log/rinetd.log | awk '{print $2}' | awk -F'.'
'{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n |
head -n 50
```

7. php-syslog-ng

8. Log Analyzer

http://loganalyzer.adiscon.com/

9. Splunk

10. Octopussy

http://www.8pussy.org/

11. eventlog-to-syslog

https://code.google.com/p/eventlog-to-syslog/

12. Apache Flume

http://flume.apache.org/

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application.

12.1. 安装 Apache flume

```
cd /usr/local/src
wget
http://mirrors.tuna.tsinghua.edu.cn/apache/flume/1.7.0/apache-
flume-1.7.0-bin.tar.gz
tar zvf apache-flume-1.7.0-bin.tar.gz
mv apache-flume-1.7.0-bin /srv/apache-flume-1.7.0
ln -s /srv/apache-flume-1.7.0 /srv/apache-flume
cp /srv/apache-flume/conf/flume-env.sh.template /srv/apache-
flume/conf/flume-env.sh
cp /srv/apache-flume/conf/flume-conf.properties.template
/srv/apache-flume/conf/flume-conf.properties
```

12.2. 基本配置

```
# Define a memory channel called ch1 on agent1
agent1.channels.ch1.type = memory

# Define an Avro source called avro-source1 on agent1 and tell
it
# to bind to 0.0.0.0:41414. Connect it to channel ch1.
```

```
agent1.sources.avro-source1.channels = ch1
agent1.sources.avro-source1.type = avro
agent1.sources.avro-source1.bind = 0.0.0.0
agent1.sources.avro-source1.port = 41414

# Define a logger sink that simply logs all events it receives
# and connect it to the other end of the same channel.
agent1.sinks.log-sink1.channel = ch1
agent1.sinks.log-sink1.type = logger

# Finally, now that we've defined all of our components, tell
# agent1 which ones we want to activate.
agent1.channels = ch1
agent1.sources = avro-source1
agent1.sinks = log-sink1
```

在agent的机器上执行以下命令启动flume server

```
$ bin/flume-ng agent --conf ./conf/ -f conf/flume.conf -
Dflume.root.logger=DEBUG,console -n agent1
```

在client的机器上执行以下命令接收日志

```
$ bin/flume-ng avro-client --conf conf -H localhost -p 41414 -F
/etc/passwd -Dflume.root.logger=DEBUG,console
```

12.3. 配置 MySQL 存储日志

```
cp flume-mysql-sink-1.x.x.jar /srv/apache-flume/lib
cp /usr/share/java/mysql-connector-java.jar /srv/apache-
flume/lib
```

```
DROP TABLE IF EXISTS flume;
CREATE TABLE flume (
ROW_KEY BIGINT,
timeid BIGINT,
systemid INT,
functionid INT,
bussinessid TEXT,
bussinessType INT,
nodeid INT,
userid INT,
logtype INT,
timeout INT,
detail TEXT,
PRIMARY KEY (ROW_KEY)
) ENGINE=INNODB DEFAULT CHARSET=utf8;
```

```
a1.sources = source1
a1.sinks = sink1
a1.channels = channel1

# Describe/configure source1
a1.sources.source1.type = avro
a1.sources.source1.bind = 0.0.0.0
a1.sources.source1.port = 44444

# Use a channel which buffers events in memory
a1.channels.channel1.type = memory
a1.channels.channel1.capacity = 1000
a1.channels.channel1.transactionCapactiy = 100

# Bind the source and sink to the channel
a1.sources.source1.channels = channel1
a1.sinks.sink1.channel = channel1
a1.sinks.sink1.type=org.flume.mysql.sink.RegexMysqlSink
a1.sinks.sink1.hostname=192.168.10.94
a1.sinks.sink1.databaseName=logging
a1.sinks.sink1.port=3306
a1.sinks.sink1.user=flume
a1.sinks.sink1.password=flume
a1.sinks.sink1.regex=^([^,]+),([^,]+),([^,]+),([^,]+),([^,]+),
([^,]+),([^,]+),([^,]+),([^,]+),([^,]+),([^,]+)$
a1.sinks.sink1.tableName=flume
a1.sinks.sink1.colNames=ROW_KEY,timeid,systemid,functionid,buss
inessid,bussinessType,nodeid,userid,logtype,timeout,detail
a1.sinks.sink1.colDataTypes=LONG,LONG,INT,INT,TEXT,INT,INT,INT,
INT,INT,TEXT
```

```
a1.sinks.sink1.batchSize=100
```

启动

```
[root@netkiller]/srv/apache-flume# bin/flume-ng agent --conf
conf --conf-file conf/flume-conf.properties --name a1 -
Dflume.root.logger=INFO,console
```

12.4. 配置 HDFS 存储日志

```
配置conf/flume.conf

# Define a memory channel called ch1 on agent1
agent1.channels.ch1.type = memory

# Define an Avro source called avro-source1 on agent1 and tell
it
# to bind to 0.0.0.0:41414. Connect it to channel ch1.
agent1.sources.spooldir-source1.channels = ch1
agent1.sources.spooldir-source1.type = spooldir
agent1.sources.spooldir-
source1.spoolDir=/opt/hadoop/flume/tmpData
agent1.sources.spooldir-source1.bind = 0.0.0.0
agent1.sources.spooldir-source1.port = 41414

# Define a logger sink that simply logs all events it receives
# and connect it to the other end of the same channel.
agent1.sinks.hdfs-sink1.channel = ch1
agent1.sinks.hdfs-sink1.type = hdfs
agent1.sinks.hdfs-sink1.hdfs.path = hdfs://master:9000/flume
agent1.sinks.hdfs-sink1.hdfs.filePrefix = events-
agent1.sinks.hdfs-sink1.hdfs.useLocalTimeStamp = true
agent1.sinks.hdfs-sink1.hdfs.round = true
agent1.sinks.hdfs-sink1.hdfs.roundValue = 10

# Finally, now that we've defined all of our components, tell
# agent1 which ones we want to activate.
agent1.channels = ch1
agent1.sources = spooldir-source1
```

```
agent1.sinks = hdfs-sink1
```

启动agent

```
bin/flume-ng agent --conf ./conf/ -f ./conf/flume.conf --name
agent1 -Dflume.root.logger=DEBUG,console
```

查看结果

到Hadoop提供的WEB GUI界面可以看到刚刚上传的文件是否成功。GUI界面地址为：http://master:50070/explorer.html#/test 其中，master为Hadoop的Namenode所在的机器名。

13. graylog - Enterprise Log Management for All

https://www.graylog.org

第 7 章 上一代监控系统

流行于2015年之前

1. Varnish Dashboard

https://github.com/brandonwamboldt/varnish-dashboard

2. Cacti

Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with hundreds of devices.

homepage: http://www.cacti.net/

2.1. Install Cacti for Ubuntu

过程 7.1. Step by step Install Cacti

- Install Cacti for

  Ubuntu

  ```
  netkiller@shenzhen:~$ sudo apt-get install cacti
  ```

  ```
     ┌────────────────┤ Configuring libphp-adodb
  ├────────────────────┐
     │
  │
     │ WARNING: include path for php has changed!
  │
     │
  │
     │ libphp-adodb is no longer installed in /usr/share/adodb. New
  installation path is now      │
     │ /usr/share/php/adodb.
  │
     │
  │
     │ Please update your php.ini file. Maybe you must also change
  your web-server configuraton.  │
     │
  │
     │                                           <Ok>
  │
  ```

```
   │
│
   └──────────────────────────────────────────────────────┘
```

```
 ┌──────────────────────┤ Configuring cacti ├──────────────────────┐
 │
│
 │ cacti must have a database installed and configured before it can
be used.  If you like,       │
 │ this can be handled with dbconfig-common.
│
 │
│
 │ If you are an advanced database administrator and know that you
want to perform this          │
 │ configuration manually, or if your database has already been
installed and configured, you    │
 │ should refuse this option.  Details on what needs to be done
should most likely be provided   │
 │ in /usr/share/doc/cacti.
│
 │
│
 │ Otherwise, you should probably choose this option.
│
 │
│
 │ Configure database for cacti with dbconfig-common?
│
 │
│
 │                           <Yes>                              <No>
│
 │
│
 └────────────────────────────────────────────────────────┘
```

```
  ┌────────────────────┤ Configuring cacti ├───────────────────────┐
  │ What is the password for the administrative account with which
this package should create    │
  │ its MySQL database and user?
│
  │
```

```
│
  │ Password of your database's administrative user:
│
  │
│
  │
──────────────────────────────────────────────────────────────────────
──────────────────────── │
  │
│
  │                          <Ok>
<Cancel>                          │
  │
│
  └────────────────────────────────────────────────────────┘
```

reset password of admin

```
mysql> use cacti;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

mysql> select * from user_auth;
+----+----------+----------------------------------+-------+------------
---+----------------------+-----------+-----------+--------------+------
----------+------------+--------------+--------------+--------------+--
----------------------+---------+
| id | username | password                         | realm | full_name
| must_change_password | show_tree | show_list | show_preview |
graph_settings | login_opts | policy_graphs | policy_trees |
policy_hosts | policy_graph_templates | enabled |
+----+----------+----------------------------------+-------+------------
---+----------------------+-----------+-----------+--------------+------
----------+------------+--------------+--------------+--------------+--
----------------------+---------+
|  1 | admin    | 21232f297a57a5a743894a0e4a801fc3 |     0 |
Administrator | on                   | on        | on        | on
| on             |          1 |             1 |            1 |
1 |                      1 | on      |
|  3 | guest    | 43e9a4ab75570f5b                 |     0 | Guest
Account | on                   | on        | on        | on           |
on             |          3 |             1 |            1 |
1 |                      1 |         |
+----+----------+----------------------------------+-------+------------
---+----------------------+-----------+-----------+--------------+------
```

```
----------+------------+---------------+--------------+--------------+--
----------------------+---------+
2 rows in set (0.00 sec)


mysql> update user_auth set password=md5("chen") where id='1' and
username='admin';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

2.2. Yum 安装

```
yum install cacti
```

创建数据库

```
# mysql -u root -p
mysql> create database cacti;
mysql> GRANT ALL ON cacti.* TO cacti@localhost IDENTIFIED BY 'cacti';
mysql> FLUSH privileges;
mysql> quit;

mysql -ucacti -pcacti cacti < /usr/share/doc/cacti-0.8.8b/cacti.sql
```

数据配置

```
# cat /etc/cacti/db.php
<?php
/*
 +-------------------------------------------------------------------------
---+
 | Copyright (C) 2004-2013 The Cacti Group
|
 |
|
 | This program is free software; you can redistribute it and/or
|
 | modify it under the terms of the GNU General Public License
```

```
|
 | as published by the Free Software Foundation; either version 2
|
 | of the License, or (at your option) any later version.
|
 |
|
 | This program is distributed in the hope that it will be useful,
|
 | but WITHOUT ANY WARRANTY; without even the implied warranty of
|
 | MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
|
 | GNU General Public License for more details.
|
 +----------------------------------------------------------------------
---+
 | Cacti: The Complete RRDTool-based Graphing Solution
|
 +----------------------------------------------------------------------
---+
 | This code is designed, written, and maintained by the Cacti Group.
See  |
 | about.php and/or the AUTHORS file for specific developer information.
|
 +----------------------------------------------------------------------
---+
 | http://www.cacti.net/
|
 +----------------------------------------------------------------------
---+
*/

/* make sure these values refect your actual database/host/user/password
*/
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "cacti";
$database_port = "3306";
$database_ssl = false;

/*
   Edit this to point to the default URL of your Cacti install
   ex: if your cacti install as at http://serverip/cacti/ this
   would be set to /cacti/
*/
//$url_path = "/cacti/";

/* Default session name - Session name must contain alpha characters */
```

```
//$cacti_session_name = "Cacti";

?>
```

配置httpd

```
# cat /etc/httpd/conf.d/cacti.conf
#
# Cacti: An rrd based graphing tool
#

# For security reasons, the Cacti web interface is accessible only to
# localhost in the default configuration. If you want to allow other
clients
# to access your Cacti installation, change the httpd ACLs below.
# For example:
# On httpd 2.4, change "Require host localhost" to "Require all
granted".
# On httpd 2.2, change "Allow from localhost" to "Allow from all".

Alias /cacti    /usr/share/cacti

<Directory /usr/share/cacti/>
        <IfModule mod_authz_core.c>
                # httpd 2.4
                #Require host any
                Require all granted
        </IfModule>
</Directory>

<Directory /usr/share/cacti/install>
        # mod_security overrides.
        # Uncomment these if you use mod_security.
        # allow POST of application/x-www-form-urlencoded during install
        #SecRuleRemoveById 960010
        # permit the specification of the rrdtool paths during install
        #SecRuleRemoveById 900011
</Directory>


# These sections marked "Require all denied" (or "Deny from all")
# should not be modified.
# These are in place in order to harden Cacti.
<Directory /usr/share/cacti/log>
        <IfModule mod_authz_core.c>
                Require all denied
        </IfModule>
```

```
</Directory>
<Directory /usr/share/cacti/rra>
        <IfModule mod_authz_core.c>
                Require all denied
        </IfModule>
</Directory>
```

2.3. Source Install

Cacti requires MySQL, PHP, RRDTool, net-snmp, and a webserver that supports PHP such as Apache.

```
sudo apt-get install rrdtool
sudo apt-get install snmp snmpd
sudo apt-get install php5-snmp
```

At first, install snmp for linux

1. wget http://www.cacti.net/downloads/cacti-0.8.7b.tar.gz
2. tar zxvf cacti-0.8.7b.tar.gz
3. mv cacti-0.8.7b /home/netkiller/public_html/cacti
4. mysqladmin --user=root create cacti
5. mysql -uroot -p cacti < cacti.sql
6. echo "GRANT ALL ON cacti.* TO cactiuser@localhost IDENTIFIED BY 'somepassword';" | mysql -uroot -p
7. echo "flush privileges;" | mysql -uroot -p
8. vi include/config.php

例 7.1. cacti config.php

```


$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
```

```
$database_username = "cactiuser";
$database_password = "somepassword";
$database_port = "3306";
```

9. crontab -e

   */5 * * * * php /var/www/neo.6600.org/html/cacti/poller.php > /dev/null 2>&1

   or

   /etc/crontab

   */5 * * * * nobody php /home/netkiller/public_html/cacti/poller.php > /dev/null 2>&1

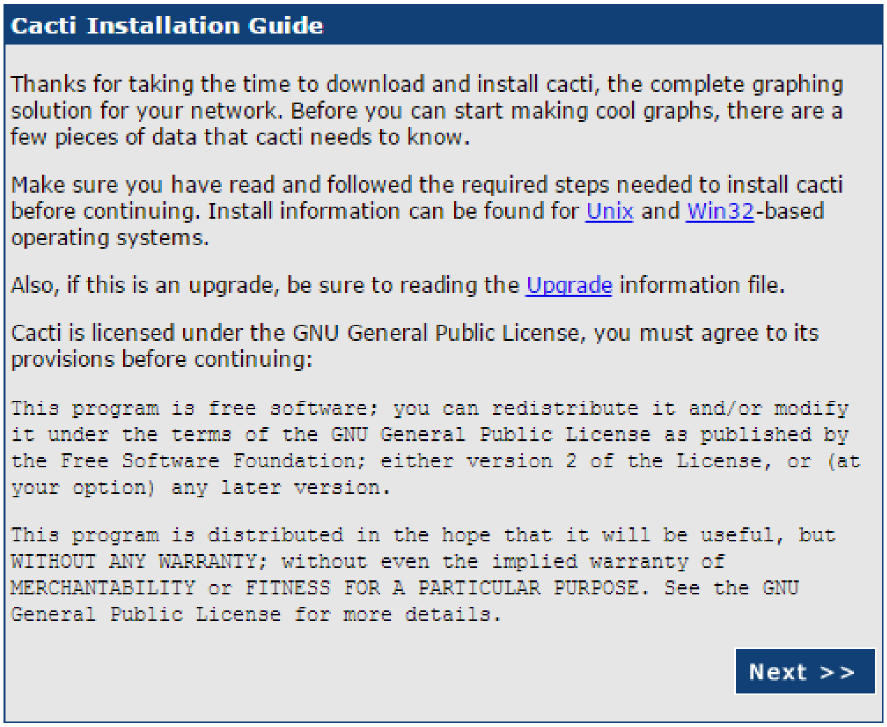
10. mkdir -p /var/log/cacti/

configure cacti

http://your-server/cacti/

2.4. Web 安装

登陆WEB界面http://your-server/cacti/



Cacti Installation Guide

Thanks for taking the time to download and install cacti, the complete graphing solution for your network. Before you can start making cool graphs, there are a few pieces of data that cacti needs to know.

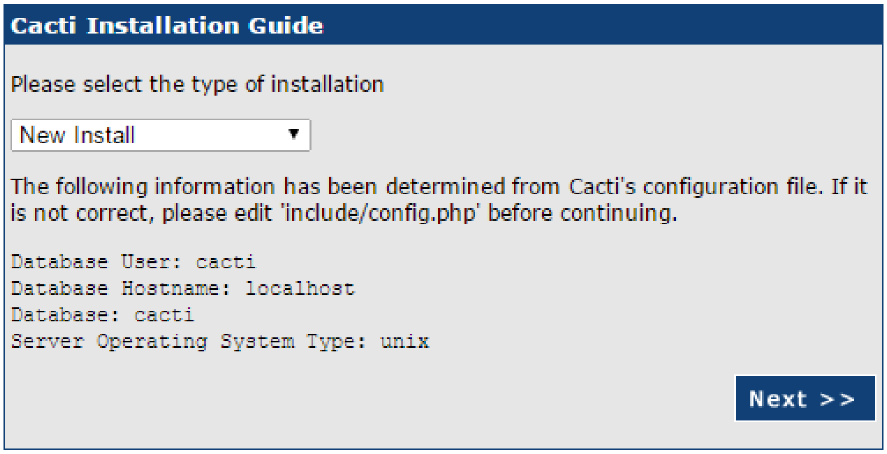
Make sure you have read and followed the required steps needed to install cacti before continuing. Install information can be found for Unix and Win32-based operating systems.

Also, if this is an upgrade, be sure to reading the Upgrade information file.

Cacti is licensed under the GNU General Public License, you must agree to its provisions before continuing:

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or (at
your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.
```

下一步



Cacti Installation Guide

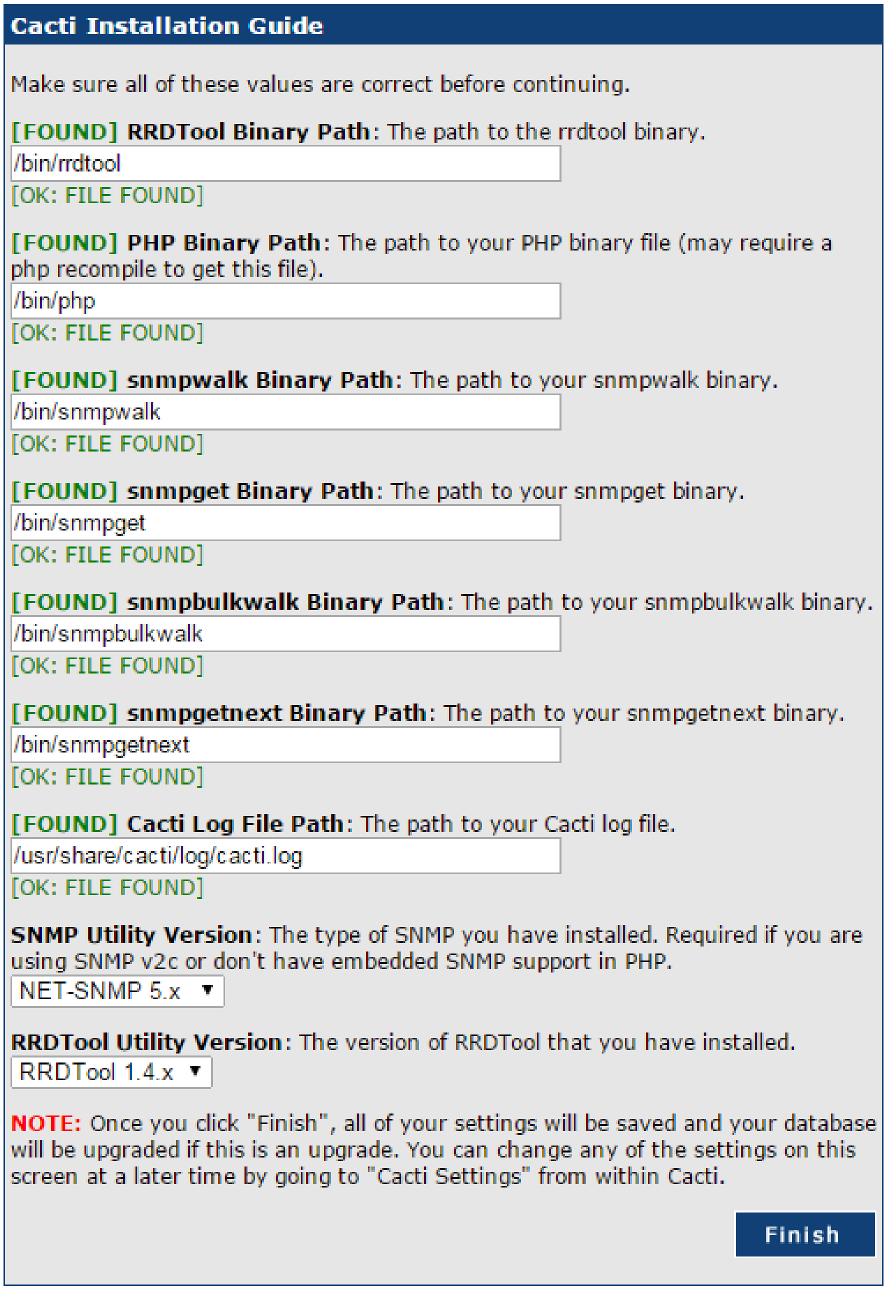
Please select the type of installation

New Install

The following information has been determined from Cacti's configuration file. If it is not correct, please edit 'include/config.php' before continuing.

```
Database User: cacti
Database Hostname: localhost
Database: cacti
Server Operating System Type: unix
```

下一步



Cacti Installation Guide

Make sure all of these values are correct before continuing.

[FOUND] RRDTool Binary Path: The path to the rrdtool binary.
/bin/rrdtool
[OK: FILE FOUND]

[FOUND] PHP Binary Path: The path to your PHP binary file (may require a php recompile to get this file).
/bin/php
[OK: FILE FOUND]

[FOUND] snmpwalk Binary Path: The path to your snmpwalk binary.
/bin/snmpwalk
[OK: FILE FOUND]

[FOUND] snmpget Binary Path: The path to your snmpget binary.
/bin/snmpget
[OK: FILE FOUND]

[FOUND] snmpbulkwalk Binary Path: The path to your snmpbulkwalk binary.
/bin/snmpbulkwalk
[OK: FILE FOUND]

[FOUND] snmpgetnext Binary Path: The path to your snmpgetnext binary.
/bin/snmpgetnext
[OK: FILE FOUND]

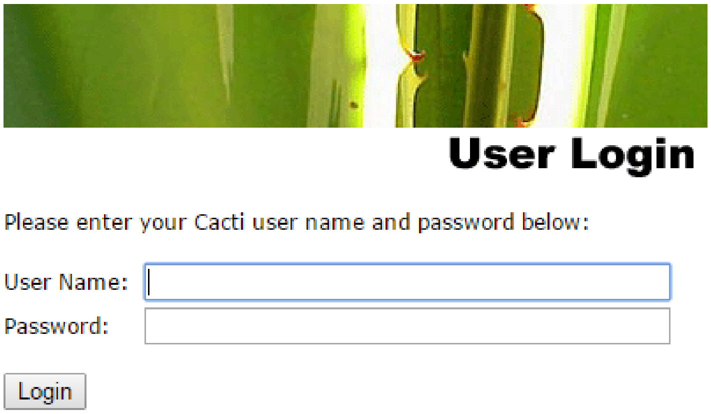
[FOUND] Cacti Log File Path: The path to your Cacti log file.
/usr/share/cacti/log/cacti.log
[OK: FILE FOUND]

SNMP Utility Version: The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.
NET-SNMP 5.x ▼

RRDTool Utility Version: The version of RRDTool that you have installed.
RRDTool 1.4.x ▼

NOTE: Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.

完成



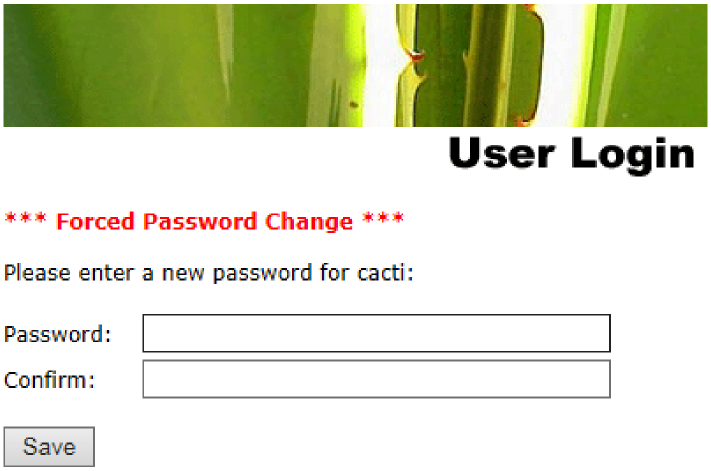
User Login

Please enter your Cacti user name and password below:

User Name:

Password:

登陆Cacti，首次登陆默认用户admin,密码是admin



User Login

*** Forced Password Change ***

Please enter a new password for cacti:

Password:

Confirm:

登陆后会提示你修改密码

2.5. Cacti plugins

http://docs.cacti.net/plugins

下载插件解压到下面目录

```
cd /usr/share/cacti/plugins
```

进入Console -> Plugin Management配置插件

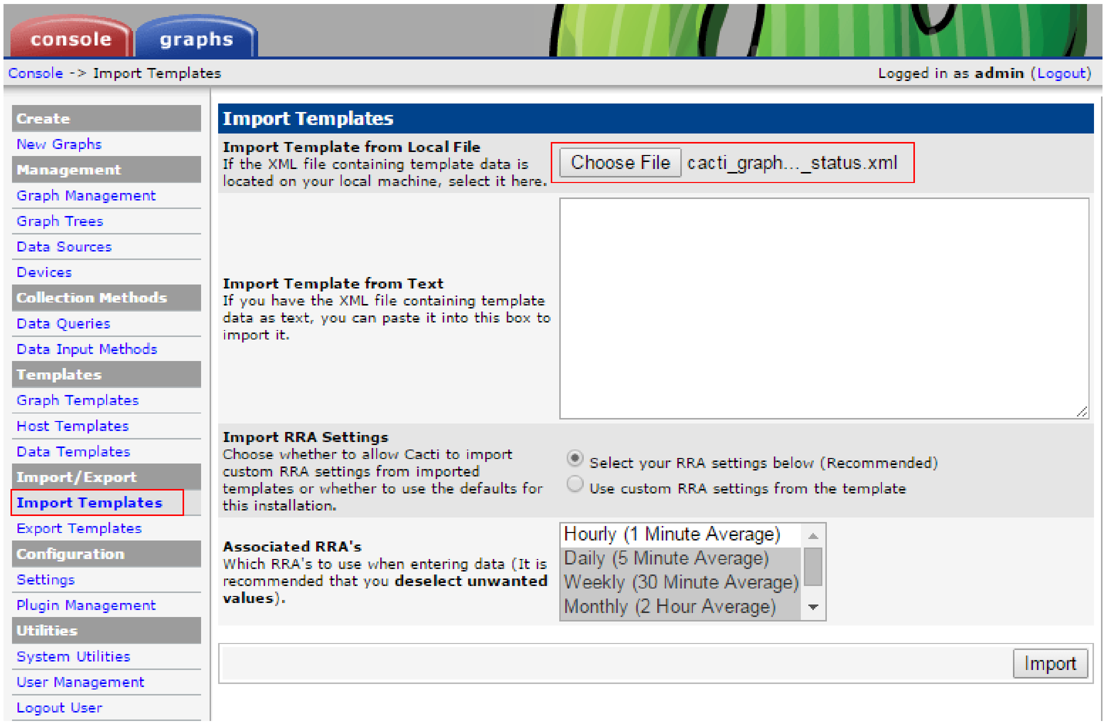
Percona monitoring plugins

http://www.percona.com/software/percona-monitoring-plugins

```
yum localinstall http://www.percona.com/downloads/percona-monitoring-
plugins/1.1.4/percona-cacti-templates-1.1.4-1.noarch.rpm
```

2.6. Template

模板的导入步骤是首先点击"Choose File"按钮选择文件



Console -> Import Templates

Logged in as **admin** (Logout)

Import Templates

Import Template from Local File
If the XML file containing template data is located on your local machine, select it here.
Choose File cacti_graph..._status.xml

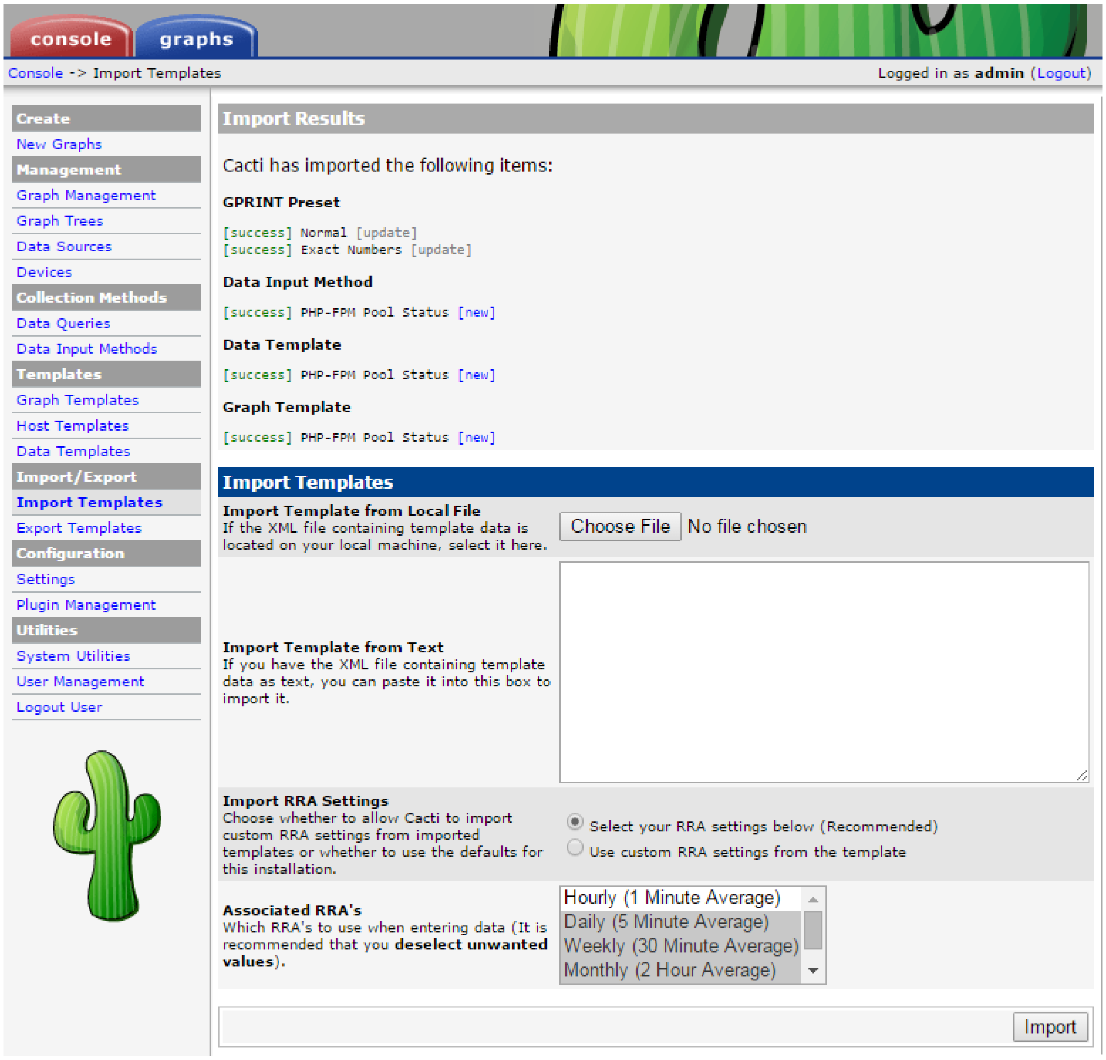
Import Template from Text
If you have the XML file containing template data as text, you can paste it into this box to import it.

Import RRA Settings
Choose whether to allow Cacti to import custom RRA settings from imported templates or whether to use the defaults for this installation.
● Select your RRA settings below (Recommended)
○ Use custom RRA settings from the template

Associated RRA's
Which RRA's to use when entering data (It is recommended that you **deselect unwanted values**).
Hourly (1 Minute Average)
Daily (5 Minute Average)
Weekly (30 Minute Average)
Monthly (2 Hour Average)

Import

然后点击Import按钮



Import Results

Cacti has imported the following items:

GPRINT Preset

```
[success] Normal [update]
[success] Exact Numbers [update]
```

Data Input Method

```
[success] PHP-FPM Pool Status [new]
```

Data Template

```
[success] PHP-FPM Pool Status [new]
```

Graph Template

```
[success] PHP-FPM Pool Status [new]
```

Import Templates

Import Template from Local File
If the XML file containing template data is located on your local machine, select it here.

No file chosen

Import Template from Text
If you have the XML file containing template data as text, you can paste it into this box to import it.

Import RRA Settings
Choose whether to allow Cacti to import custom RRA settings from imported templates or whether to use the defaults for this installation.

● Select your RRA settings below (Recommended)
○ Use custom RRA settings from the template

Associated RRA's
Which RRA's to use when entering data (It is recommended that you **deselect unwanted values**).

Hourly (1 Minute Average)
Daily (5 Minute Average)
Weekly (30 Minute Average)
Monthly (2 Hour Average)

确认导入事项，最后点击Import按钮。

完成倒入后，配置数据采集脚本，请继续阅读下面章节。

Nginx

```
wget http://forums.cacti.net/download/file.php?id=12676
```

http://forums.cacti.net/about26458.html

nginx 配置

```
    location /nginx_status {
        stub_status on;
        access_log  off;
        allow 22.82.21.12;
        deny all;
    }
```

php-fpm

```
yum -y install perl-FCGI perl-FCGI-Client perl-LWP-Protocol-http10

git clone https://github.com/oscm/Cacti.git
cd Cacti
cp Templates/php-fpm/get_php_fpm_status.pl /usr/share/cacti/scripts/
chmod +x /usr/share/cacti/scripts/get_php_fpm_status.pl
```

配置连接协议

```
# vim +/mode /usr/share/cacti/scripts/get_php_fpm_status.pl

#my $mode = MODE_FCGI; 注释此行
my $mode = MODE_HTTP; 添加此行
```

配置 php-fpm.conf 文件

```
; Default Value: not set
pm.status_path = /status
```

配置nginx

```
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 22.82.21.12;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_param SCRIPT_FILENAME $fastcgi_script_name;
        include fastcgi_params;
    }
```

MySQL

Template: http://code.google.com/p/mysql-cacti-templates/

```
$ cd /usr/local/src/
$ wget http://mysql-cacti-templates.googlecode.com/files/better-cacti-
templates-1.1.8.tar.gz
$ tar zxvf better-cacti-templates-1.1.8.tar.gz
$ cd better-cacti-templates-1.1.8/
$ cp scripts/ss_get_mysql_stats.php /usr/share/cacti/scripts/
```

default password

```
vim /usr/share/cacti/site/scripts/ss_get_mysql_stats.php.cnf
<?php
$mysql_user = "root";
$mysql_pass = "s3cret";
?>
```

Import Templates

倒入下面模板 templates/cacti_host_template_x_mysql_server_ht_0.8.6i-sver1.1.8.xml

```
"Import/Export" -> "Import Templates" -> "Import Template from Local
File" -> Import
```

设置模版

```
Templates ->

X MyISAM Indexes DT
X MyISAM Key Cache DT
X MySQL Binary/Relay Logs DT
X MySQL Command Counters DT
X MySQL Connections DT
X MySQL Files and Tables DT
X MySQL Handlers DT
X MySQL Network Traffic DT
```

```
X MySQL Processlist DT
X MySQL Query Cache DT
X MySQL Query Cache Memory DT
X MySQL Replication DT
X MySQL Select Types DT
X MySQL Sorts DT
X MySQL Table Locks DT
X MySQL Temporary Objects DT
X MySQL Threads DT
X MySQL Transaction Handler DT

->

Custom Data
Hostname
Username        #单击复选框，并输入默认用户名
Password        #单击复选框，并输入默认密码
Port

-> Save
```

Redis

```
easy_install redis
```

https://github.com/oscm/Cacti.git

```
cp redis-stats.py /usr/share/cacti/scripts/
```

测试采集脚本

```
# python redis-stats.py 172.18.52.163
total_connections_received:578761 connected_clients:14
used_memory:870032 expires:47 keys:47 total_commands_processed:1814080
```

Percona JMX Monitoring Template for Cacti

http://www.percona.com/doc/percona-monitoring-plugins/1.0/cacti/jmx-templates.html

3. Nagios

homepage: http://www.nagios.org/

3.1. Install

Nagios core

Nagios 是一种开放源代码监视软件，它可以扫描主机、服务、网络方面存在的问题。Nagios 与其他类似的包之间的主要区别在于，Nagios 将所有的信息简化为“工作（working）”、“可疑的（questionable）”和“故障（failure）”状态，并且 Nagios 支持由插件组成的非常丰富的“生态系统”。这些特性使得用户能够进行有效安装，在此过程中无需过多地关心细节内容，只提供他们所需的信息即可。

install

```
$ sudo apt-get install nagios3 nagios-nrpe-plugin
```

add user nagiosadmin for nagios

```
$ sudo htpasswd -c /etc/nagios2/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Create a new nagcmd group for allowing external commands to be submitted through the web interface. Add both the nagios user and the apache user to the group.

```
$ groupadd nagcmd
$ sudo usermod -a -G nagcmd nagios
$ sudo usermod -a -G nagcmd www-data
$ cat /etc/group
nagcmd:x:1003:nagios,www-data
```

reload apache

```
$ sudo /etc/init.d/apache2 reload
 * Reloading web server config apache2                    [ OK ]
```

Monitor Client nrpe

```
nagios-nrpe-server --------> nagios core (nagios-nrpe-plugin)
```

nagios-nrpe-server 的功能是向服务器发送监控数据,而服务器端通过nagios-nrpe-plugin接收监控数据。

```
sudo apt-get install nagios-nrpe-server nagios-plugins
```

/etc/nagios/nrpe.cfg

/etc/nagios/nrpe_local.cfg

```
$ sudo vim /etc/nagios/nrpe_local.cfg
allowed_hosts=172.16.1.2

command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c
10
command[check_load]=/usr/lib/nagios/plugins/check_load -w
15,10,5 -c 30,25,20
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs
-w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -
w 150 -c 200
```

```
command[check_procs]=/usr/lib/nagios/plugins/check_procs -w 150
-c 200
command[check_swap]=/usr/lib/nagios/plugins/check_swap -w 20% -c
10%
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w
20% -c 10% -e
command[check_disk_root]=/usr/lib/nagios/plugins/check_disk -w
20% -c 10% -p /
command[check_disk_home]=/usr/lib/nagios/plugins/check_disk -w
20% -c 10% -p /home
command[check_sda_iostat]=/usr/lib/nagios/plugins/check_iostat -
d sda -w 100 -c 200
command[check_sdb_iostat]=/usr/lib/nagios/plugins/check_iostat -
d sdb -w 100 -c 200
# command[check_uri_user]=/usr/lib/nagios/plugins/check_http -I
127.0.0.1 -p 80 -u http://example.com/test/ok.php
# command[check_mysql]=/usr/lib/nagios/plugins/check_mysql -H
localhost -u root -ppassword test -P 3306
```

重启后生效

```
/etc/init.d/nagios-nrpe-server restart
```

Monitoring Windows Machines

Nagios 可以监控windows服务器，需要安装下面软件。

NSClient++

http://sourceforge.net/projects/nscplus

PNP4Nagios 图表插件

http://www.pnp4nagios.org/

3.2. nagios

Install Nagios & Plugins

```
[root@database ~]# yum -y install nagios nagios-plugins-all
nagios-plugins-nrpe
```

Create the default Nagios web access user & set a password

```
# htpasswd -c /etc/nagios/passwd nagiosadmin
```

Verify default config files

```
nagios -v /etc/nagios/nagios.cfg
```

Start Nagios

```
Start Nagios
```

Configure it to start on boot

```
chkconfig --levels 345 nagios on
```

http://localhost/nagios/

3.3. nrpe node

```
# yum install nrpe nagios-plugins-all

allowed_hosts=172.16.1.2

command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5
-c 10
command[check_load]=/usr/lib64/nagios/plugins/check_load -w
15,10,5 -c 30,25,20
```

```
command[check_hda1]=/usr/lib64/nagios/plugins/check_disk -w 20%
-c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/lib64/nagios/plugins/check_proc
s -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib64/nagios/plugins/check_procs
-w 150 -c 200
command[check_http]=/usr/lib64/nagios/plugins/check_http -I
127.0.0.1 -p 80 -u http://www.example.com/index.html
command[check_swap]=/usr/lib64/nagios/plugins/check_swap -w 20%
-c 10%
command[check_all_disks]=/usr/lib64/nagios/plugins/check_disk -w
20% -c 10% -e

# chkconfig nrpe on
# service nrpe start
```

其实没有必要安装所有的监控插件

```
yum install nrpe -y
yum install nagios-plugins-disk  nagios-plugins-load nagios-
plugins-ping nagios-plugins-procs nagios-plugins-swap nagios-
plugins-users -y
```

3.4. 配置 Nagios

```
$ sudo vim /etc/nagios3/nagios.cfg

cfg_dir=/etc/nagios3/hosts
cfg_dir=/etc/nagios3/servers
cfg_dir=/etc/nagios3/switches
cfg_dir=/etc/nagios3/routers

admin_email=nagios, neo.chen@example.com
```

authorized

add user neo for nagios

```
$ sudo htpasswd /etc/nagios3/htpasswd.users neo
New password:
Re-type new password:
Adding password for user neo
```

```
# grep default_user_name cgi.cfg
#default_user_name=guest

# grep authorized cgi.cfg
authorized_for_system_information=nagiosadmin
authorized_for_configuration_information=nagiosadmin
authorized_for_system_commands=nagiosadmin
authorized_for_all_services=nagiosadmin
authorized_for_all_hosts=nagiosadmin
authorized_for_all_service_commands=nagiosadmin
authorized_for_all_host_commands=nagiosadmin
#authorized_for_read_only=user1,user2
```

```
$ sudo vim /etc/nagios3/cgi.cfg

authorized_for_all_services=nagiosadmin,neo
authorized_for_all_hosts=nagiosadmin,neo
```

contacts

```
$ sudo vim /etc/nagios3/conf.d/contacts_nagios2.cfg

################################################################
###############
# contacts.cfg
################################################################
###############

define contact{
        contact_name                    neo
        alias                           Neo
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,r
        service_notification_commands   notify-service-by-email
        host_notification_commands      notify-host-by-email
        email                           neo.chen@example.com
        }

################################################################
###############
################################################################
###############
#
# CONTACT GROUPS
#
################################################################
###############
################################################################
###############

# We only have one contact in this simple configuration file, so
there is
# no need to create more than one contact group.

define contactgroup{
        contactgroup_name       admins
        alias                   Nagios Administrators
        members                 root, neo
        }

```

当服务出现w—报警(warning),u—未知(unkown),c—严重(critical),r—从异常恢复到正常，在这四种情况下通知联系人

当主机出现d- 当机(down),u—返回不可达(unreachable),r—从异常情况恢复正常,在这3种情况下通知联系人

确认 contact_groups 已经设置

```

neo@monitor:/etc/nagios3$ grep admins conf.d/generic-
```

```
host_nagios2.cfg
                contact_groups                  admins
neo@monitor:/etc/nagios3$ grep admins conf.d/generic-
service_nagios2.cfg
                contact_groups                  admins
```

hostgroups

```
$ sudo vim /etc/nagios3/conf.d/hostgroups_nagios2.cfg

define hostgroup {
        hostgroup_name  mysql-servers
                alias           MySQL Servers
                members         *
        }
```

generic-service

```
$ cat /etc/nagios3/conf.d/generic-service_nagios2.cfg
# generic service template definition
define service{
        name                            generic-service ; The
'name' of this service template
        active_checks_enabled           1       ; Active service
checks are enabled
        passive_checks_enabled          1       ; Passive
service checks are enabled/accepted
        parallelize_check               1       ; Active service
checks should be parallelized (disabling this can lead to major
performance problems)
        obsess_over_service             1       ; We should
obsess over this service (if necessary)
        check_freshness                 0       ; Default is to
NOT check service 'freshness'
        notifications_enabled           1       ; Service
notifications are enabled
        event_handler_enabled           1       ; Service event
handler is enabled
```

```
        flap_detection_enabled          1       ; Flap detection
is enabled
        failure_prediction_enabled      1       ; Failure
prediction is enabled
        process_perf_data               1       ; Process
performance data
        retain_status_information       1       ; Retain status
information across program restarts
        retain_nonstatus_information    1       ; Retain non-
status information across program restarts
                notification_interval           0
; Only send notifications on status change by default.
                is_volatile                     0
                check_period                    24x7
                normal_check_interval           5
                retry_check_interval            1
                max_check_attempts              4
                notification_period             24x7
                notification_options            w,u,c,r
                contact_groups                  admins
        register                        0       ; DONT REGISTER
THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!
        }
```

- notification_interval 报警发送间隔，单位分钟
- normal_check_interval 间隔时间
- retry_check_interval 重试间隔时间
- max_check_attempts 检查次数，4次失败后报警

SOUND OPTIONS

发出警报声

```


$ sudo vim /etc/nagios3/cgi.cfg

# SOUND OPTIONS
# These options allow you to specify an optional audio file
# that should be played in your browser window when there are
# problems on the network.  The audio files are used only in
# the status CGI.  Only the sound for the most critical problem
# will be played.  Order of importance (higher to lower) is as
# follows: unreachable hosts, down hosts, critical services,
# warning services, and unknown services. If there are no
# visible problems, the sound file optionally specified by
# 'normal_sound' variable will be played.
#
#
# <varname>=<sound_file>
#
# Note: All audio files must be placed in the /media
subdirectory
# under the HTML path (i.e. /usr/local/nagios/share/media/).

host_unreachable_sound=hostdown.wav
host_down_sound=hostdown.wav
service_critical_sound=critical.wav
service_warning_sound=warning.wav
service_unknown_sound=warning.wav
normal_sound=noproblem.wav
```

SMS 短信

```
vim /etc/nagios3/commands.cfg

# 'notify-host-by-sms' command definition
define command{
        command_name    notify-host-by-sms
        command_line    /srv/sms/sms $CONTACTPAGER$ "Host:
$HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo:
$HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n"
        }

# 'notify-service-by-sms' command definition
define command{
        command_name    notify-service-by-sms
        command_line    /srv/sms/sms $CONTACTPAGER$ "Service:
$SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$"
        }
```

```
sudo vim /etc/nagios3/conf.d/contacts_nagios2.cfg
define contact{
        contact_name                    neo
        alias                           Neo
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,r
        service_notification_commands   notify-service-by-email,
notify-service-by-sms
        host_notification_commands      notify-host-by-email,
notify-host-by-sms
        email                           neo.chen@example.com
        pager
13113668899
        }
```

nrpe plugins

```
neo@monitor:/etc/nagios3/hosts$ sudo cat www.example.com.cfg

define host{
        use             generic-host            ; Inherit
default values from a template
        host_name       www.example.com             ; The name
we're giving to this host
        alias           Some Remote Host        ; A longer name
associated with the host
        address         172.16.1.10              ; IP address of
the host
        hostgroups      http-servers                    ; Host
groups this host is associated with
        }

# NRPE disk check.
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-disk
        check_command
check_nrpe_1arg!check_all_disks!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-users
        check_command
check_nrpe_1arg!check_users!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-swap
        check_command
check_nrpe_1arg!check_swap!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-procs
        check_command
check_nrpe_1arg!check_total_procs!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-load
        check_command
check_nrpe_1arg!check_load!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-zombie_procs
        check_command
check_nrpe_1arg!check_zombie_procs!172.16.1.10
}
```

3.5. 配置监控设备

routers

```
vim /etc/nagios3/routers/firewall.cfg

define host{

        use             generic-host; Inherit default values
from a template

        host_name       firewall         ; The name we're giving
to this switch

        alias           Cisco PIX 515E Firewall ; A longer name
associated with the switch

        address         172.16.1.254            ; IP address of
the switch

        hostgroups      all,networks            ; Host groups
this switch is associated with

        }

define service{

        use                     generic-service ; Inherit values
from a template

        host_name                       firewall ; The name of
the host the service is associated with

        service_description     PING            ; The service
description

        check_command           check_ping!200.0,20%!600.0,60%
; The command used to monitor the service

        normal_check_interval   5       ; Check the service
every 5 minutes under normal conditions

        retry_check_interval    1       ; Re-check the service
every minute until its final/hard state is determined

        }
```

```
define service{

        use                     generic-service ; Inherit values
from a template

        host_name                       firewall

        service_description     Uptime

        check_command           check_snmp!-C public -o
sysUpTime.0

        }

```

host

```
define service{
    use                                 local-service
    host_name                           www.example.com
    service_description                 Host Alive
    check_command                       check-host-alive
    }
```

service

http

hosts

```
$ cat /etc/nagios3/hosts/www.example.com.cfg
define host{

        use             generic-host            ; Inherit
default values from a template

        host_name       www.example.com             ; The name
we're giving to this host
```

```
        alias           Some Remote Host        ; A longer name
associated with the host

        address         120.132.14.6           ; IP address of
the host

        hostgroups      all,http-servers        ; Host groups
this host is associated with

        }

define service{

        use             generic-service         ; Inherit
default values from a template

        host_name               www.example.com

        service_description     HTTP

        check_command   check_http

        }
```

HTTP状态

```
neo@monitor:~$ /usr/lib/nagios/plugins/check_http -H
www.example.com -I 172.16.0.8 -s "HTTs"
HTTP CRITICAL: HTTP/1.1 404 Not Found - string not found - 336
bytes in 0.001 second response time |time=0.000733s;;;0.000000
size=336B;;;0

neo@monitor:~$ /usr/lib/nagios/plugins/check_http -H
www.example.com -I 172.16.0.8 -e '404'
HTTP OK: Status line output matched "404" - 336 bytes in 0.001
second response time |time=0.000715s;;;0.000000 size=336B;;;0
```

mysql hosts

```
$ sudo vim /etc/nagios3/hosts/mysql.cfg


define host{

        use             generic-host            ; Inherit
default values from a template

        host_name       mysql-master.example.com            ;
The name we're giving to this host

        alias           Some Remote Host        ; A longer name
associated with the host

        address         172.16.1.6             ; IP address of
the host

        hostgroups      all,mysql-servers       ; Host groups
this host is associated with

        }

define service{

        use             generic-service         ; Inherit
default values from a template

        host_name               mysql-master.example.com

        service_description     MySQL

        check_command
check_mysql_database!user!passwd!database

        }
```

check_tcp

```
define service{
    use                         generic-service
    host_name                   db.example.com
    service_description         MySQL Master1 Port
    check_command               check_tcp!3306
    }
```

3.6. Nagios Plugins

检查命令配置文件 /etc/nagios-plugins/config/

check_ping

nagios check_ping命令使用方法

```
具体如下:
-H     主机地址
-w      WARNING 状态:    响应时间(毫秒), 丢包率 (%)    阀值
-c      CRITICAL状态:     响应时间(毫秒), 丢包率 (%)    阀值
-p      发送的包数                默认5个包
-t      超时时间                  默认10秒
-4|-6                              使用ipv4|ipv6 地址      默认ipv4
```

实例:

```
/usr/lib64/nagios/plugins/check_ping -H 74.125.71.106 -w
100.0,20% -c 200.0,50%
```

check_procs

```
# /usr/lib64/nagios/plugins/check_procs
PROCS OK: 75 processes

# /usr/lib64/nagios/plugins/check_procs -a mingetty
PROCS OK: 6 processes with args 'mingetty'

# /usr/lib64/nagios/plugins/check_procs -C crond
```

```
PROCS OK: 1 process with command name 'crond'
```

check_users

监控如果有用户登陆就发出警告

```
# /usr/lib64/nagios/plugins/check_users -w 0 -c 5
USERS WARNING - 1 users currently logged in |users=1;0;5;0
```

监控用户上线5

```
# /usr/lib64/nagios/plugins/check_users -w 5 -c 50
USERS OK - 1 users currently logged in |users=1;5;50;0
```

check_http

命令定义

```
define command{
        command_name    check_http_404
        command_line    /usr/lib/nagios/plugins/check_http -H
'$HOSTADDRESS$' -I '$HOSTADDRESS$' -e '404'
        }

define command{
        command_name    check_http_status
        command_line    /usr/lib/nagios/plugins/check_http -H
'$HOSTADDRESS$' -I '$HOSTADDRESS$' -e '$ARG1$'
        }

define command{
        command_name    check_http_url
        command_line    /usr/lib/nagios/plugins/check_http -H
'$HOSTADDRESS$' -I '$HOSTADDRESS$'  -u '$ARG1$'
        }
```

默认HTTP健康检查超时时间是10秒，如果你的网站需要更长的时间才能打开可以使用-t参数修改默认Timeout时间

```
# 'check_http' command definition
define command{
        command_name    check_http
        command_line    /usr/lib/nagios/plugins/check_http -t 30
-H '$HOSTADDRESS$' -I '$HOSTADDRESS$'
        }
```

```
# /srv/nagios/libexec/check_http -H www.163.com
HTTP OK: HTTP/1.0 200 OK - 657627 bytes in 1.772 second response
time |time=1.771681s;;;0.000000 size=657627B;;;0

$ /usr/lib/nagios/plugins/check_http -H www.example.com -I
172.16.0.8 -s "HTTs"
HTTP CRITICAL: HTTP/1.1 404 Not Found - string not found - 336
bytes in 0.001 second response time |time=0.000733s;;;0.000000
size=336B;;;0

$ /usr/lib/nagios/plugins/check_http -H www.example.com -I
172.16.0.8 -e '404'
HTTP OK: Status line output matched "404" - 336 bytes in 0.001
second response time |time=0.000715s;;;0.000000 size=336B;;;0
```

check_mysql

命令参数

```
check_mysql [-d database] [-H host] [-P port] [-s socket]
       [-u user] [-p password] [-S]


/usr/lib64/nagios/plugins/check_mysql -d dbname -H
202.176.120.10 -P 3306 -u test -p password
Uptime: 254264  Threads: 16  Questions: 535110791  Slow queries:
21  Opens: 110  Flush tables: 1  Open tables: 81  Queries per
```

```
second avg: 2104.547
```

check_mysql

```
$ /usr/lib64/nagios/plugins/check_mysql --hostname=172.16.1.5 --
port=3306 --username=monitor --password=monitor
Uptime: 27001  Threads: 8  Questions: 25280156  Slow queries:
14941  Opens: 1389932  Flush tables: 3  Open tables: 128
Queries per second avg: 936.267
```

mysql.cfg check_mysql_replication

```
cat >> /usr/lib64/nagios/plugins/check_mysql_replication <<EOF
#!/bin/bash

declare -a slave_is

slave_is=($(mysql -h$1 -umonitor -pxmNhj -e "show slave
status\G"|grep Running |awk '{print $2}'))

if [ "${slave_is[0]}" = "Yes" -a "${slave_is[1]}" = "Yes" ]
     then
     echo "OK - Slave is running"
     exit 0
else
     echo "Critical - Slave is error"
     exit 2
fi
EOF
```

```
sudo chmod +x /usr/lib64/nagios/plugins/check_mysql_replication
/usr/lib64/nagios/plugins/check_mysql_replication 172.16.1.4
Critical - slave is error
```

```
vim /etc/nagios-plugins/config/mysql.cfg

# 'check_mysql_replication' command definition
define command{
        command_name    check_mysql_replication
        command_line
/usr/lib/nagios/plugins/check_mysql_replication $HOSTADDRESS$
}
define command{
        command_name    check_mysql_replication_host
        command_line
/usr/lib/nagios/plugins/check_mysql_replication '$ARG1$'
}
```

nrpe.cfg check_mysql_replication

nrpe.cfg

```
cat >> /usr/lib64/nagios/plugins/check_mysql_replication <<EOF
#!/bin/bash

declare -a slave_is

slave_is=($(mysql -umonitor -pxmNhj -e "show slave
status\G"|grep Running |awk '{print $2}'))

if [ "${slave_is[0]}" = "Yes" -a "${slave_is[1]}" = "Yes" ]
     then
     echo "OK - slave is running"
     exit 0
else
     echo "Critical - slave is error"
     exit 2
fi
EOF

command[check_mysql_slave]=/usr/lib64/nagios/plugins/check_mysql
```

```
_replication

/usr/local/nagios/libexec/check_nrpe -H 192.168.1.1
/usr/local/nagios/libexec/check_nrpe -H 192.168.1.1 -c
check_mysql_replication


define service {
        host_name 192.168.10.232
        service_description check_mysql_replication
        check_period 24x7
        max_check_attempts 5
        normal_check_interval 3
        retry_check_interval 2
        contact_groups mygroup
        notification_interval 5
        notification_period 24x7
        notification_options w,u,c,r
        check_command check_nrpe!check_mysql_replication
}
```

Disk

disk.cfg

```
$ cat /etc/nagios-plugins/config/disk.cfg
# 'check_disk' command definition
define command{
        command_name    check_disk
        command_line    /usr/lib/nagios/plugins/check_disk -w
'$ARG1$' -c '$ARG2$' -e -p '$ARG3$'
        }

# 'check_all_disks' command definition
define command{
        command_name    check_all_disks
        command_line    /usr/lib/nagios/plugins/check_disk -w
'$ARG1$' -c '$ARG2$' -e
        }

# 'ssh_disk' command definition
```

```
define command{
        command_name    ssh_disk
        command_line    /usr/lib/nagios/plugins/check_by_ssh -H
'$HOSTADDRESS$' -C '/usr/lib/nagios/plugins/check_disk -w
'\''$ARG1$' -c '\''$ARG2$'\'' -e -p '\''$ARG3$'\'
        }

####
# use these checks, if you want to test IPv4 connectivity on
IPv6 enabled systems
####

# 'ssh_disk_4' command definition
define command{
        command_name    ssh_disk_4
        command_line    /usr/lib/nagios/plugins/check_by_ssh -H
'$HOSTADDRESS$' -C '/usr/lib/nagios/plugins/check_disk -w
'\''$ARG1$'\'' -c '\''$ARG2$'\'' -e -p '\''$ARG3$'\' -4
        }
```

check_disk

WARNING/CRITICAL 报警阀值

```
-w 10% -c 5%
-w 100M -c 50M
```

-p, --path=PATH, --partition=PARTITION 参数监控路径，可以一次写多个参数

```
$ /usr/lib/nagios/plugins/check_disk -w 10% -c 5% -p / -p /opt -
p /boot
DISK OK - free space: / 23872 MB (66% inode=92%); /opt 99242 MB
(47% inode=93%); /boot 276 MB (63% inode=99%);|
/=11767MB;33792;35669;0;37547
/opt=110882MB;199232;210300;0;221369 /boot=160MB;414;437;0;460

$ /usr/lib/nagios/plugins/check_disk -w 100M -c 50M -p / -p /opt
-p /boot
DISK OK - free space: / 23872 MB (66% inode=92%); /opt 99242 MB
```

```
(47% inode=93%); /boot 276 MB (63% inode=99%);|
/=11768MB;37447;37497;0;37547
/opt=110882MB;221269;221319;0;221369 /boot=160MB;360;410;0;460
```

-x, --exclude_device=PATH 排除监控路径

```
/usr/lib64/nagios/plugins/check_disk -w 10% -c 5% -e -x /bak -x
/u01
```

disk-smb.cfg

```
$ cat disk-smb.cfg
# 'check_disk_smb' command definition
define command{
        command_name    check_disk_smb
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
H '$ARG1$' -s '$ARG2$'
        }


# 'check_disk_smb_workgroup' command definition
define command{
        command_name    check_disk_smb_workgroup
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
H '$ARG1$' -s '$ARG2$' -W '$ARG3$'
        }


# 'check_disk_smb_host' command definition
define command{
        command_name    check_disk_smb_host
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$'
        }


# 'check_disk_smb_workgroup_host' command definition
define command{
        command_name    check_disk_smb_workgroup_host
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$' -W '$ARG3$'
        }


# 'check_disk_smb_user' command definition
define command{
        command_name    check_disk_smb_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
H '$ARG1$' -s '$ARG2$' -u '$ARG3$' -p '$ARG4$' -w '$ARG5$' -c
'$ARG6$'
        }


# 'check_disk_smb_workgroup_user' command definition
define command{
        command_name    check_disk_smb_workgroup_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
H '$ARG1$' -s '$ARG2$' -W '$ARG3$' -u '$ARG4$' -p '$ARG5$'
        }


# 'check_disk_smb_host_user' command definition
define command{
        command_name    check_disk_smb_host_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$' -u '$ARG3$' -p
'$ARG4$'
        }


# 'check_disk_smb_workgroup_host_user' command definition
define command{
        command_name    check_disk_smb_workgroup_host_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb -
a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$' -W '$ARG3$' -u
'$ARG4$' -p '$ARG5$'
        }
```

check_tcp

端口检查

```
$ /usr/lib/nagios/plugins/check_tcp -H 172.16.1.2 -p 80
TCP OK - 0.000 second response time on port
80|time=0.000369s;;;0.000000;10.000000
```

Memcache

```
$ /usr/lib64/nagios/plugins/check_tcp -H localhost -p 11211 -t 5
-E -s 'stats\r\nquit\r\n' -e 'uptime' -M crit
TCP OK - 0.001 second response time on port 11211 [STAT pid
29253
STAT uptime 36088
STAT time 1311100189
STAT version 1.4.5
STAT pointer_size 64
STAT rusage_user 3.207512
STAT rusage_system 50.596308
STAT curr_connections 10
STAT total_connections 97372
STAT connection_structures 84
STAT cmd_get 84673
STAT cmd_set 273
STAT cmd_flush 0
STAT get_hits 84336
STAT get_misses 337
STAT delete_misses 0
STAT delete_hits 0
STAT incr_misses 0
STAT incr_hits 0
STAT decr_misses 0
STAT decr_hits 0
STAT cas_misses 0
STAT cas_hits 0
STAT cas_badval 0
STAT auth_cmds 0
STAT auth_errors 0
STAT bytes_read 49280152
STAT bytes_written 46326517326
STAT limit_maxbytes 4294967296
STAT accepting_conns 1
STAT listen_disabled_num 0
STAT threads 4
STAT conn_yields 0
STAT bytes 1345
STAT curr_items 14
STAT total_items 241
STAT evictions 0
STAT reclaimed 135
END]|time=0.000658s;;;0.000000;5.000000
```

Redis

```
# /usr/lib64/nagios/plugins/check_tcp -H 192.168.2.1 -p 6379 -t
5 -E -s 'info\r\n' -q 'quit\r\n' -e 'uptime_in_days' -M crit
TCP OK - 0.001 second response time on port 6379 [$1043
redis_version:2.4.10
redis_git_sha1:00000000
redis_git_dirty:0
arch_bits:64
multiplexing_api:epoll
gcc_version:4.4.6
process_id:21331
uptime_in_seconds:18152153
uptime_in_days:210
lru_clock:1801614
used_cpu_sys:1579.41
used_cpu_user:2279.26
used_cpu_sys_children:54.32
used_cpu_user_children:54.11
connected_clients:2
connected_slaves:1
client_longest_output_list:0
client_biggest_input_buf:0
blocked_clients:0
used_memory:1158016
used_memory_human:1.10M
used_memory_rss:1560576
used_memory_peak:1289920
used_memory_peak_human:1.23M
mem_fragmentation_ratio:1.35
mem_allocator:jemalloc-2.2.5
loading:0
aof_enabled:0
changes_since_last_save:2
bgsave_in_progress:0
last_save_time:1423107828
bgrewriteaof_in_progress:0
total_connections_received:594376
total_commands_processed:1350747
expired_keys:12199
evicted_keys:0
keyspace_hits:511525
keyspace_misses:124116
pubsub_channels:0
pubsub_patterns:0
latest_fork_usec:361
vm_enabled:0
role:master
slave0:192.168.6.1,58091,online
db0:keys=1913,expires=7]|time=0.000815s;;;0.000000;5.000000
```

check_log

官方的 check_log 有很多缺陷，不能监控大文件。它的监控原理是 cat log to oldlog 然后通过diff比较

check_traffic

http://exchange.nagios.org/directory/Plugins/Network-Connections,-Stats-and-Bandwidth/check_traffic-2Esh/details

https://github.com/cloved/check_traffic

网卡流量监测

Nagios nrpe plugins

nrpe 插件接收来自nagios-nrpe-server数据报告

```
cat /etc/nagios3/hosts/host.example.org.cfg

define host{

        use                generic-host            ; Inherit
default values from a template

        host_name       host.example.org        ; The name we're
giving to this host

        alias           Some Remote Host        ; A longer name
associated with the host

        address         172.16.1.3              ; IP address of
the host

        hostgroups      all                     ; Host groups
this host is associated with

        }

# NRPE disk check.
define service {
        use                             generic-service
        host_name                       backup
        service_description             nrpe-disk
        check_command
check_nrpe_1arg!check_all_disks!172.16.1.3
}
define service {
        use                             generic-service
        host_name                       backup
        service_description             nrpe-users
        check_command
check_nrpe_1arg!check_users!172.16.1.3
}
define service {
        use                             generic-service
        host_name                       backup
        service_description             nrpe-swap
        check_command
check_nrpe_1arg!check_swap!172.16.1.3
}
define service {
        use                             generic-service
        host_name                       backup
        service_description             nrpe-procs
        check_command
check_nrpe_1arg!check_procs!172.16.1.3
}
```

check_nt

Define windows services that should be monitored.

```
# Define a host for the Windows machine we'll be monitoring
# Change the host_name, alias, and address to fit your situation

define host{
use             windows-server              ; Inherit default
values from a template
host_name   remote-windows-host      ; The name we're giving to
this host
alias            Remote Windows Host     ; A longer name
associated with the host
address       192.168.1.4                   ; IP address of the
remote windows host
}

define service{
use                     generic-service
host_name               remote-windows-host
service_description     NSClient++ Version
check_command           check_nt!CLIENTVERSION
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     Uptime
check_command           check_nt!UPTIME
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     CPU Load
check_command           check_nt!CPULOAD!-l 5,80,90
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     Memory Usage
check_command           check_nt!MEMUSE!-w 80 -c 90
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     C:\ Drive Space
check_command           check_nt!USEDDISKSPACE!-l c -w 80 -c 90
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     W3SVC
check_command           check_nt!SERVICESTATE!-d SHOWALL -l
W3SVC
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     Explorer
check_command           check_nt!PROCSTATE!-d SHOWALL -l
Explorer.exe
}
```

Enable Password Protection

```
define command{
command_name    check_nt
command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s
My2Secure$Password -v $ARG1$ $ARG2$
}
```

nsca - Nagios Service Check Acceptor

```
# yum install nsca
```

jmx

nagios plugin to check jmx

https://code.google.com/p/jmxquery/

```
wget https://jmxquery.googlecode.com/files/jmxquery-1.3-bin.zip
unzip jmxquery-1.3-bin.zip
chmod +x check_jmx
```

```
                <![CDATA[
# ./check_jmx -help
Usage: check_jmx [-option...] -U url -O object -A attribute
       (to query an attribute)
   or  check_jmx [-option...] -U url -O object -M method
       (to invoke a zero-argument method)
   or  check_jmx -help
       (to display this help page)

Mandatory parameters are:
 -U     JMX URL, for example:
"service:jmx:rmi:///jndi/rmi://localhost:1616/jmxrmi"
 -O     Object name to be checked, for example,
"java.lang:type=Memory"
 -A     Attribute of the object to be checked, for example,
"NonHeapMemoryUsage" (not compatible with -M switch)
 -M     Zero-argument method to be invoked (not compatible with
-A switch)

Options are:
 -K <key>
        Key for compound data, for example, "used"
 -I <info attribute>
        Attribute of the object containing information for text
output
 -J <info attribute key>
        Attribute key for -I attribute compound data, for
example, "used"
 -v[v[v[v]]]
            Verbatim level controlled as a number of v
 -w <limit>
            Warning long value
 -c <limit>
            Critical long value
 -default <value>
        Use default value if requested object/attribute/method
```

```
does not exist
 -username <user name> -password <password>
            Credentials for JMX

Note that if warning level > critical, system checks object
attribute value to be LESS THAN OR EQUAL warning, critical
If warning level < critical, system checks object attribute
value to be MORE THAN OR EQUAL warning, critical
```

例 7.2.

```
# ./check_jmx -U
service:jmx:rmi:///jndi/rmi://localhost:9012/jmxrmi -O
java.lang:type=Memory -A HeapMemoryUsage -K used -I
HeapMemoryUsage -J used -vvvv -w 731847066 -c 1045495808
JMX OK - HeapMemoryUsage.used=98617544 |
HeapMemoryUsage.used=98617544,committed=514850816;init=536870912
;max=7635730432;used=98617544
```

```
# ./check_jmx -U
service:jmx:rmi:///jndi/rmi://localhost:9012/jmxrmi -O
org:type=Spring,name=BackgroundService -A QueueSize -w 10 -c 20
JMX CRITICAL - org:type=Spring,name=BackgroundService
```

3.7. FAQ

Macro Name

http://nagios.sourceforge.net/docs/3_0/macrolist.html

插件开发手册

https://nagios-plugins.org/doc/guidelines.html#THRESHOLDFORMAT

4. Munin

http://munin-monitoring.org/

4.1. Ubuntu

http://munin-monitoring.org/

Installation Monitor Server

```
$ sudo apt-get install munin

neo@monitor:~$ sudo vim /etc/munin/munin.conf
neo@monitor:~$ sudo service munin-node restart


[example.com]
        address 127.0.0.1
        use_node_name yes

[web2]
    address 172.16.1.2
    use_node_name yes

[web3]
    address 172.16.1.3
    use_node_name yes

[database]
    address 172.16.1.10
    use_node_name yes
```

Installation Node

```
sudo apt-get install munin-node

vim /etc/munin/munin-node.conf

allow ^172\.16\.1\.2$
```

Additional Plugins

```
sudo apt-get install munin-plugins-extra
```

plugins

mysql

```
ln -s /usr/share/munin/plugins/mysql_* /etc/munin/plugins/
```

/etc/munin/plugin-conf.d/munin-node

```
$ sudo vim /etc/munin/plugin-conf.d/munin-node

[mysql*]
user root
env.mysqlopts --defaults-file=/etc/mysql/debian.cnf
env.mysqluser debian-sys-maint
env.mysqlconnection
DBI:mysql:mysql;mysql_read_default_file=/etc/mysql/debian.cnf

[mysql*]
env.mysqlopts -h 192.168.3.40 -uneo -pchen
```

apache

```
$ sudo vim /etc/munin/plugin-conf.d/munin-node

[apache_*]
env.url   http://127.0.0.1/server-status?auto
env.ports 80
```

4.2. CentOS

```
# rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-
release-5-4.noarch.rpm
# yum install munin -y
# yum install munin-node -y
# yum install munin-java-plugins -y
# yum install unbound-munin -y
# service munin-node start
# chkconfig munin-node on
```

test

```
# telnet localhost 4949
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
# munin node at datacenter.example.com
list
cpu df df_inode entropy forks fw_packets http_loadtime if_err_eth0
if_eth0 interrupts iostat iostat_ios irqstats load memory munin_stats
netstat open_files open_inodes proc_pri processes sendmail_mailqueue
sendmail_mailstats sendmail_mailtraffic swap threads uptime users vmstat
yum
```

http://localhost/munin/

4.3. 用户认证

```
$ sudo vim /etc/apache2/conf.d/munin.conf

        AuthUserFile /etc/munin/munin-htpasswd
        AuthName "Munin"
        AuthType Basic
        require valid-user
```

4.4. munin-node and plugins

config: /etc/munin/munin-node.conf

plugins: /usr/share/munin/plugins/

munin-node.conf

```
allow ^127\.0\.0\.1$
allow ^192\.168\.3\.5$
```

mysql plugin

mysql

```
# ln -s /usr/share/munin/plugins/mysql_* /etc/munin/plugins
```

```
# vim /etc/munin/plugin-conf.d/munin-node
env.mysqlopts -uneo -pchen

# or

env.mysqlopts -h 172.16.1.17 -u monitor -ppassword

# service munin-node start
```

验证安装，telnet localhost 4949 之后，执行 fetch mysql_queries

apache plugin

apache

```
# ln -s /usr/share/munin/plugins/apache_* /etc/munin/plugins
```

```
# vim /etc/httpd/conf/httpd.conf
ExtendedStatus On
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from .example.com
        Allow from localhost
</Location>
```

```
# /etc/init.d/httpd restart
# service munin-node restart
```

验证安装,telnet localhost 4949 之后，执行 fetch apache_processes

memcached plugin

memcached plugin要求符号链接名字的格式是: memcached_connections_[IP Address]_[Port], IP与Port是在符号链接名字中配置的

```
ln -s /usr/share/munin/plugins/memcached_bytes_
/etc/munin/plugins/memcached_bytes_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_connections_
/etc/munin/plugins/memcached_connections_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_hits_
/etc/munin/plugins/memcached_hits_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_items_
/etc/munin/plugins/memcached_items_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_requests_
/etc/munin/plugins/memcached_requests_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_traffic_
/etc/munin/plugins/memcached_traffic_127_0_0_1_11211
```

验证安装，telnet localhost 4949 之后，执行 fetch memcached_requests_127_0_0_1_11211

4.5. munin.conf

```
# vim /etc/munin/munin.conf
# a simple host tree
[localhost]
    address 127.0.0.1
    use_node_name yes
[database]
    address 192.168.3.40
    use_node_name yes
```

4.6. munin-node

```
# yum install munin-node -y
# chkconfig munin-node on
# service munin-node start
```

munin-node.conf

vim /etc/munin/munin-node.conf allow ^127\.16\.1\.2$

5. Observium

http://www.observium.org

5.1. Installation

```
aptitude install libapache2-mod-php5 php5-cli php5-mysql php5-
gd php5-snmp \
php-pear snmp graphviz subversion mysql-server mysql-client
rrdtool \
fping imagemagick whois mtr-tiny nmap ipmitool
```

安装 Net_IPv6

```
Install the IPv4 and IPv6 pear libraries:
$ sudo pear install Net_IPv6
$ sudo pear install Net_IPv4
```

安装observium软件

http://www.observium.org/observium-latest.tar.gz

```
$ wget http://www.observium.org/observium-latest.tar.gz
$ tar zxvf observium-latest.tar.gz
$ sudo mv observium /opt
$ cd /opt/observium/
$ cp config.php.default config.php
$ sudo mkdir graphs rrd
$ chown www-data.www-data graphs rrd
$ mkdir /opt/observium/logs
```

创建数据库SQL脚本

```
CREATE DATABASE observium;
GRANT ALL PRIVILEGES ON observium.* TO 'observium'@'localhost'
IDENTIFIED BY '<observium db password>';
```

创建数据库

```
$ mysql -uroot -p
Enter password: <mysql root password>
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 238145
Server version: 5.1.41-3ubuntu12.10 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> CREATE DATABASE observium;
Query OK, 1 row affected (0.10 sec)

mysql> GRANT ALL PRIVILEGES ON observium.* TO
'observium'@'localhost' IDENTIFIED BY 'observium';
Query OK, 0 rows affected (0.06 sec)
```

修改配置文件

```
$ vim config.php

### Database config
$config['db_host'] = "localhost";
$config['db_user'] = "observium";
$config['db_pass'] = "observium";
$config['db_name'] = "observium";

### List of networks to allow scanning-based discovery
$config['nets'][] = "172.16.1.0/24";
$config['nets'][] = "172.16.3.0/24";
```

```
or
$config['nets'][] = "172.16.0.0/16";
```

创建数据库表

```
$ mysql -uobservium -pobservium observium < database-schema.sql
```

配置WEB服务器

```
$ sudo vim /etc/apache2/sites-available/observium

<VirtualHost *:80>
       ServerAdmin webmaster@localhost
       ServerName  observium.domain.com
       DocumentRoot /opt/observium/html
       <Directory />
               Options FollowSymLinks
               AllowOverride None
       </Directory>
       <Directory /opt/observium/html/>
               Options Indexes FollowSymLinks MultiViews
               AllowOverride All
               Order allow,deny
               allow from all
       </Directory>
       ErrorLog /var/log/apache2/error.log
       LogLevel warn
       CustomLog /var/log/apache2/access.log combined
       ServerSignature On
</VirtualHost>
```

启用Rewrite

```
$ sudo a2enmod rewrite
Enabling module rewrite.
Run '/etc/init.d/apache2 restart' to activate new
configuration!

$ sudo a2ensite observium
Enabling site observium.
Run '/etc/init.d/apache2 reload' to activate new configuration!

$ sudo apache2ctl restart
```

添加用户

```
$ ./adduser.php
Add User Tool
Usage: ./adduser.php <username> <password> <level 1-10> [email]

$ ./adduser.php neo chen 1 neo.chen@example.com

$ ./adduser.php netkiller 3655927 10 neo.chen@example.com
User netkiller added successfully




$ ./addhost.php

Observium v0.11.9.2439 Add Host Tool

Usage: ./addhost.php <hostname> [community] [v1|v2c] [port]
[udp|udp6|tcp|tcp6]

$ ./addhost.php localhost public v2c
Trying community public
Added device localhost (1)
```

```
./discovery.php -h all
```

```
./poller.php -h all
```

设置定时任务

```
$ crontab -e

33 */6  * * *  cd /opt/observium/ && ./discovery.php -h all >>
/dev/null 2>&1
*/5 *   * * *  cd /opt/observium/ && ./discovery.php -h new >>
/dev/null 2>&1
*/5 *   * * *  cd /opt/observium/ && ./poller.php -h all >>
/dev/null 2>&1

$ sudo /etc/init.d/cron reload
```

6. Ganglia

Ganglia是一个集群监控软件

Ganglia 是一个开源项目，它为高性能计算系统（例如集群和网格）提供了一个免费的可扩展分布式监视系统。

6.1. Server

```
sudo apt-get install ganglia-monitor ganglia-webfrontend

Restart apache2? 选择 Yes

sudo ln -s /usr/share/ganglia-webfrontend/ /var/www/ganglia
```

/etc/ganglia/gmond.conf

```
name = "my servers"   (只改了这个地方，改成"my cluster”)
```

在浏览器输入”http://localhost/ganglia”就可以看到Web UI

6.2. Client

```
# apt-get install ganglia-monitor
$ sudo vim /etc/ganglia/gmond.conf
sudo cp /etc/ganglia/gmond.conf  /etc/ganglia/gmond.conf.old

sudo cp /etc/ganglia/gmetad.conf /etc/ganglia/gmetad.conf.old
sudo vim /etc/ganglia/gmetad.conf

$ sudo  /etc/init.d/gmetad restart

$ sudo  /etc/init.d/ganglia-monitor restart
```

ip route add 239.2.11.71 dev eth1

6.3. Plugin

6.4. Installing Ganglia on Centos

http://www.jansipke.nl/installing-ganglia-on-centos

启动

```
# service gmond start
Starting GANGLIA gmond:                                    [
OK  ]
# chkconfig --list gmond
gmond           0:off   1:off   2:off   3:off   4:off   5:off
6:off
# chkconfig gmond on
# chkconfig --list gmond
gmond           0:off   1:off   2:on    3:on    4:on    5:on
6:off
```

7. icinga

https://www.icinga.org/

8. Graphite

http://groups.csail.mit.edu/carbon

8.1. Graphite - Scalable Realtime Graphing

http://graphite.wikidot.com/

9. Apache SkyWalking

10. BIG BROTHER

waiting ...

11. Big Sister

12. OpenNMS

http://www.opennms.org/

13. Performance Co-Pilot

http://oss.sgi.com/projects/pcp/

Performance Co-Pilot (PCP) provides a framework and services to support system-level performance monitoring and management. It presents a unifying abstraction for all of the performance data in a system, and many tools for interrogating, retrieving and processing that data.

14. Clumon Performance Monitor

http://clumon.ncsa.illinois.edu/

15. Zenoss

http://www.linuxjournal.com/article/10070

16. 商业软件

首选上ITM ，OpenView

其次 Solarwinds

国产 BTNM ，siteview

17. Hyperic HQ

http://www.hyperic.com/

18. OSSIM,Spiceworks,FireGen,LANSweeper,OSSEC,HIDS

19. HawtIO

http://hawt.io/

hawtio has lots of plugins such as: a git-based Dashboard and Wiki, logs, health, JMX, OSGi, Apache ActiveMQ, Apache Camel, Apache OpenEJB, Apache Tomcat, Jetty, JBoss and Fuse Fabric

20. moloch

https://github.com/aol/moloch

第 8 章 网络监控

1. NET SNMP (Simple Network Management Protocol)

1.1. 安装SNMP

Ubuntu

search package

```
netkiller@neo:~$ apt-cache search snmp
libsnmp-base - NET SNMP (Simple Network Management Protocol)
MIBs and Docs
libsnmp-perl - NET SNMP (Simple Network Management Protocol)
Perl5 Support
libsnmp-session-perl - Perl support for accessing SNMP-aware
devices
libsnmp9 - NET SNMP (Simple Network Management Protocol)
Library
libsnmp9-dev - NET SNMP (Simple Network Management Protocol)
Development Files
snmp - NET SNMP (Simple Network Management Protocol) Apps
snmpd - NET SNMP (Simple Network Management Protocol) Agents
php5-snmp - SNMP module for php5
tcpdump - A powerful tool for network monitoring and data
acquisition
```

安装

```
netkiller@neo:~$ sudo apt-get install snmp snmpd
```

snmpd.conf

配置 /etc/snmp/snmpd.conf

配置agentAddress

```
agentAddress  udp:172.16.1.3:161
```

```
#       sec.name  source          community
com2sec paranoid  default          chen

#           incl/excl subtree                         mask
view all    included  .1                              80
view system included  .iso.org.dod.internet.mgmt.mib-2.system
view system included  .iso.org.dod.internet.mgmt.mib-2.host
view system included  .iso.org.dod.internet.mgmt.mib-
2.interfaces
```

.iso.org.dod.internet.mgmt.mib-2.host 可以使用命令 snmptranslate -Onf -IR hrStorageDescr得到

参考:http://www.mkssoftware.com/docs/man1/snmptranslate.1.asp

SNMP v3

```
neo@debian:~$ sudo /etc/init.d/snmpd stop
Stopping network management services: snmpd snmptrapd.

neo@debian:~$ sudo net-snmp-config --create-snmpv3-user -ro -a
"netadminpassword" netadmin
adding the following line to /var/lib/snmp/snmpd.conf:
   createUser netadmin MD5 "netadminpassword" DES
adding the following line to /usr/share/snmp/snmpd.conf:
   rouser netadmin

neo@debian:~$ sudo /etc/init.d/snmpd start
Starting network management services: snmpd.
```

test

```
neo@debian:~$ snmpget -v 3 -u netadmin -l authNoPriv -a MD5 -A
<passwd> 127.0.0.1 sysUpTime.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (6342)
0:01:03.42
```

With a different password this fails:

```
neo@debian:~$ snmpget -v 3 -u netadmin -l authNoPriv -a MD5 -A
nopasswd 127.0.0.1 sysUpTime.0
snmpget: Authentication failure (incorrect password, community
or key) (Sub-id not found: (top) -> sysUpTime)
```

Note that this can be stuck in a snmp.conf file in ~/.snmp:

```
neo@debian:~$ mkdir ~/.snmp
neo@debian:~$ vim ~/.snmp/snmp.conf
defSecurityName netadmin
defContext ""
defAuthType MD5
defSecurityLevel authNoPriv
defAuthPassphrase <netadminpassword>
defVersion 3
```

test

```
neo@debian:~$ snmpget 127.0.0.1 sysUpTime.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (39471)
0:06:34.71
```

CentOS

```
yum install net-snmp -y

cp /etc/snmp/snmpd.conf{,.original}

vim /etc/snmp/snmpd.conf <<VIM > /dev/null 2>&1
:62,62s/systemview/all/
:85,85s/^#//
:162,162s/syslocation Unknown/syslocation Neo/
:163,163s/syscontact Root <root@localhost>/syscontact Neo
<netkiller@msn.com>/
:wq
VIM

service snmpd start
chkconfig snmpd on
```

Configure SNMPv3 on CentOS or RHEL

```
# yum install net-snmp-utils net-snmp-devel
# service snmpd stop
# net-snmp-create-v3-user -ro -A snmpv3pass -a MD5 -x DES
snmpv3user
# service snmpd start
```

Test SNMPv3

```
# snmpwalk -u snmpv3user -A snmpv3pass -a MD5 -l authnoPriv
192.168.1.2 -v3
```

1.2. 配置SNMP

community 配置

默认为 public, 版本支持v1与v2c, 只读权限

```
#       sec.name  source          community
com2sec notConfigUser  default       public

#       groupName      securityModel securityName
group   notConfigGroup v1           notConfigUser
group   notConfigGroup v2c           notConfigUser

#       group          context sec.model sec.level prefix read
write  notif
access  notConfigGroup ""      any       noauth    exact
systemview none none
```

现在我们新增一个 community

```
```

定义可操作的范围

下面我们定义一个最大可操作范围用于Cacti监控

```
#access  notConfigGroup ""      any       noauth    exact
systemview none none
access  notConfigGroup ""      any       noauth    exact  all
none none

#       name           incl/excl     subtree
mask(optional)
view all    included  .1                               80
```

A variable list

name

默认是 systemview 这里使用all

incl/excl

是包含于排除

subtree

视图中涉及的MIB子树

mask(optional)

掩码

1.3. SNMP 命令

snmpwalk

```
$ snmpwalk -c public -v2c 172.16.1.10 hrSWRunPerfMem | awk
'BEGIN {total_mem=0} { if ($NF == "KBytes")
{total_mem=total_mem+$(NF-1)}}  END {print total_mem}'
655784
```

$ snmpwalk -c public -v 1 127.0.0.1 1.3.6.1.2.1.1

```
netkiller@neo:/etc/snmp$ snmpwalk -c public -v 1 127.0.0.1
1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Linux neo.example.org 2.6.17-
10-server #2 SMP Tue Dec 5 22:29:32 UTC 2006 i686
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-
MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (120146)
0:20:01.46
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost>
(configure /etc/snmp/snmpd.local.conf)
SNMPv2-MIB::sysName.0 = STRING: neo.example.org
SNMPv2-MIB::sysLocation.0 = STRING: Unknown (configure
/etc/snmp/snmpd.local.conf)
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (18) 0:00:00.18
```

```
SNMPv2-MIB::sysORID.1 = OID: IF-MIB::ifMIB
SNMPv2-MIB::sysORID.2 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.3 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.4 = OID: IP-MIB::ip
SNMPv2-MIB::sysORID.5 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.6 = OID: SNMP-VIEW-BASED-ACM-
MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.7 = OID: SNMP-FRAMEWORK-
MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.8 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.9 = OID: SNMP-USER-BASED-SM-
MIB::usmMIBCompliance
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module to describe
generic objects for network interface sub-layers
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for SNMPv2
entities
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for managing
TCP implementations
SNMPv2-MIB::sysORDescr.4 = STRING: The MIB module for managing
IP and ICMP implementations
SNMPv2-MIB::sysORDescr.5 = STRING: The MIB module for managing
UDP implementations
SNMPv2-MIB::sysORDescr.6 = STRING: View-based Access Control
Model for SNMP.
SNMPv2-MIB::sysORDescr.7 = STRING: The SNMP Management
Architecture MIB.
SNMPv2-MIB::sysORDescr.8 = STRING: The MIB for Message
Processing and Dispatching.
SNMPv2-MIB::sysORDescr.9 = STRING: The management information
definitions for the SNMP User-based Security Model.
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.2 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.3 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.4 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.5 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.6 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.7 = Timeticks: (18) 0:00:00.18
SNMPv2-MIB::sysORUpTime.8 = Timeticks: (18) 0:00:00.18
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (18) 0:00:00.18
End of MIB
netkiller@neo:/etc/snmp$ snmpget -v 1 -c public localhost
sysDescr.0
SNMPv2-MIB::sysDescr.0 = STRING: Linux neo.example.org 2.6.17-
10-server #2 SMP Tue Dec 5 22:29:32 UTC 2006 i686
netkiller@neo:/etc/snmp$
```

snmpget

snmpget -v 1 -c public localhost sysDescr.0

```
snmpwalk -v 1 -c OFcx6CvN 127.0.0.1 extEntry
```

snmptest

```
# snmptest -v2c -c public localhost
Variable: system.sysDescr.0
Variable: system.sysContact.0
Variable:
Received Get Response from UDP: [127.0.0.1]:161->
[0.0.0.0]:48968
requestid 0x611A34EA errstat 0x0 errindex 0x0
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain
3.10.0-123.20.1.el7.x86_64 #1 SMP Thu Jan 29 18:05:33 UTC 2015
x86_64
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost>
(configure /etc/snmp/snmp.local.conf)
```

1.4. Cisco MBI

Cisco 3750

```
snmpwalk -c public -v2c 172.16.1.1
```

system.sysDescr

```
$ snmpget -v2c -c public 172.16.1.1 system.sysDescr.0
SNMPv2-MIB::sysDescr.0 = STRING: Cisco IOS Software, C3750
Software (C3750-IPBASE-M), Version 12.2(35)SE5, RELEASE
SOFTWARE (fc1)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 19-Jul-07 19:15 by nachen

$ snmpget -v2c -c public 172.16.1.1 sysName.0
SNMPv2-MIB::sysName.0 = STRING: Switch-3750-LAN

$ snmpwalk -v2c -c public 172.16.1.1
interfaces.ifTable.ifEntry.ifDescr
IF-MIB::ifDescr.1 = STRING: Vlan1
IF-MIB::ifDescr.2 = STRING: Vlan2
IF-MIB::ifDescr.3 = STRING: Vlan3
IF-MIB::ifDescr.4 = STRING: Vlan4
IF-MIB::ifDescr.5 = STRING: Vlan5
IF-MIB::ifDescr.5179 = STRING: StackPort1
IF-MIB::ifDescr.5180 = STRING: StackSub-St1-1
IF-MIB::ifDescr.5181 = STRING: StackSub-St1-2
IF-MIB::ifDescr.10101 = STRING: GigabitEthernet1/0/1
IF-MIB::ifDescr.10102 = STRING: GigabitEthernet1/0/2
IF-MIB::ifDescr.10103 = STRING: GigabitEthernet1/0/3
IF-MIB::ifDescr.10104 = STRING: GigabitEthernet1/0/4
IF-MIB::ifDescr.10105 = STRING: GigabitEthernet1/0/5
IF-MIB::ifDescr.10106 = STRING: GigabitEthernet1/0/6
IF-MIB::ifDescr.10107 = STRING: GigabitEthernet1/0/7
IF-MIB::ifDescr.10108 = STRING: GigabitEthernet1/0/8
IF-MIB::ifDescr.10109 = STRING: GigabitEthernet1/0/9
IF-MIB::ifDescr.10110 = STRING: GigabitEthernet1/0/10
IF-MIB::ifDescr.10111 = STRING: GigabitEthernet1/0/11
IF-MIB::ifDescr.10112 = STRING: GigabitEthernet1/0/12
IF-MIB::ifDescr.10113 = STRING: GigabitEthernet1/0/13
IF-MIB::ifDescr.10114 = STRING: GigabitEthernet1/0/14
IF-MIB::ifDescr.10115 = STRING: GigabitEthernet1/0/15
IF-MIB::ifDescr.10116 = STRING: GigabitEthernet1/0/16
IF-MIB::ifDescr.10117 = STRING: GigabitEthernet1/0/17
IF-MIB::ifDescr.10118 = STRING: GigabitEthernet1/0/18
IF-MIB::ifDescr.10119 = STRING: GigabitEthernet1/0/19
IF-MIB::ifDescr.10120 = STRING: GigabitEthernet1/0/20
IF-MIB::ifDescr.10121 = STRING: GigabitEthernet1/0/21
IF-MIB::ifDescr.10122 = STRING: GigabitEthernet1/0/22
IF-MIB::ifDescr.10123 = STRING: GigabitEthernet1/0/23
IF-MIB::ifDescr.10124 = STRING: GigabitEthernet1/0/24
```

```
IF-MIB::ifDescr.10125 = STRING: GigabitEthernet1/0/25
IF-MIB::ifDescr.10126 = STRING: GigabitEthernet1/0/26
IF-MIB::ifDescr.10127 = STRING: GigabitEthernet1/0/27
IF-MIB::ifDescr.10128 = STRING: GigabitEthernet1/0/28
IF-MIB::ifDescr.14501 = STRING: Null0


$ snmpget -v2c -c public 172.16.1.1 interfaces.ifNumber.0
IF-MIB::ifNumber.0 = INTEGER: 37
```

Cisco ASA 5550

```
snmpget -v2c -c public 172.16.1.254 IF-MIB::ifInOctets.3 IF-
MIB::ifInOctets.9 IF-MIB::ifOutOctets.3 IF-MIB::ifOutOctets.9
snmpget -v2c -c public 172.16.1.254 IF-MIB::ifOperStatus.3 IF-
MIB::ifOperStatus.9
```

```
#!/bin/bash
echo -n `date +%H:%M:%S` " "
snmpget -v2c -c public 172.16.1.254 IF-MIB::ifInOctets.3 IF-
MIB::ifInOctets.9 IF-MIB::ifOutOctets.3 IF-MIB::ifOutOctets.9 |
awk -F ': ' '{print $2}' | tr "\n" " "
echo
```

```
$ crontab -l
# m h  dom mon dow   command
*/5 * * * * /home/mgmt/test/test.sh >> /home/mgmt/test/test.log
```

2. Bandwidth

http://bandwidthd.sourceforge.net/

2.1. apt-get install

```
$ apt-cache search bandwidthd
bandwidthd - Tracks usage of TCP/IP and builds html files with
graphs
bandwidthd-pgsql - Tracks usage of TCP/IP and builds html files
with graphs

$ sudo apt-get install bandwidthd

  ┌─────────────────────────────────────────┤ BandwidthD
├─────────────────────────────────────────┐
  │ Bandwidthd needs to know which interface it should listen
for traffic on. Only a single      │
  │ interface can be specified. If you want to listen on all
interfaces you should specify the   │
  │ metainterface "any". Running "bandwidthd -l" will list
available interfaces.               │
  │
│
  │ Interface to listen on:
│
  │
│
  │                                              any
│
  │                                              lo
│
  │                                              eth0
│
  │                                              eth1
│
  │                                              tun0
│
  │
│
```

```
  │
│
  │                                            <Ok>
│
  │
│

└────────────────────────────────────────────────────────────────
─────────────────────────────────┘


 ┌──────────────────────────────────────────┤ BandwidthD
├──────────────────────────────────────────┐
 │ Bandwidthd can create graphs for one or several ip-subnets.
Subnets are specified either in   │
 │ dotted-quad format (192.168.0.0 255.255.0.0) or in CIDR
format (192.168.0.0/16) and           │
 │ separated by a comma. Example: 192.168.0.0/16, 10.0.0.0
255.0.0.0, 172.16.1.0/24. If you      │
 │ don't know what to specify then you can use 0.0.0.0/0 but it
is strongly discouraged.          │
 │
│
 │ Subnets to log details about:
│
 │
│
 │ 10.8.0.2/32, 172.16.2.0/24, 10.8.0.0/24,
172.16.1.0/24_______________________________________ │
 │
│
 │                                            <Ok>
│
 │
│

└────────────────────────────────────────────────────────────────
──────────────────────────────────┘

 $ sudo mkdir /www/bandwidth
 $ sudo vim /etc/bandwidthd/bandwidthd.conf
 htdocs_dir "/www/bandwidthd"

 $ sudo /etc/init.d/bandwidthd restart
 * Stopping BandwidthD bandwidthd              [ OK ]
 * Starting BandwidthD bandwidthd              [ OK ]
```

http://localhost/bandwidthd/index.html

2.2. CentOS rpm/yum

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/i386/epel-
release-5-4.noarch.rpm

# yum search bandwidthd
bandwidthd.i386 : Tracks network usage and builds html and
graphs

# yum install bandwidthd

# rpm -ql bandwidthd
/etc/bandwidthd.conf
/etc/httpd/conf.d/bandwidthd.conf
/etc/rc.d/init.d/bandwidthd
/usr/sbin/bandwidthd
/usr/share/doc/bandwidthd-2.0.1
/usr/share/doc/bandwidthd-2.0.1/CHANGELOG
/usr/share/doc/bandwidthd-2.0.1/README
/usr/share/doc/bandwidthd-2.0.1/TODO
/usr/share/doc/bandwidthd-2.0.1/phphtdocs
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/bd_pgsql_purge.sh
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/config.conf
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/details.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/footer.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/graph.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/include.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/index.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/legend.gif
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/logo.gif
/var/www/bandwidthd
/var/www/bandwidthd/htdocs
/var/www/bandwidthd/htdocs/legend.gif
/var/www/bandwidthd/htdocs/logo.gif
                        </screen>
                        <screen>
```

```
# cat /etc/bandwidthd.conf

####################################################
# Bandwidthd.conf
#
# Commented out options are here to provide
# documentation and represent defaults

# Subnets to collect statistics on.  Traffic that
# matches none of these subnets will be ignored.
# Syntax is either IP Subnet Mask or CIDR
subnet 10.0.0.0 255.0.0.0
subnet 192.168.0.0/16
subnet 172.16.0.0/12

# Device to listen on
# Bandwidthd listens on the first device it detects
# by default.  Run "bandwidthd -l" for a list of
# devices.
#dev "eth0"

###################################################
# Options that don't usually get changed

# An interval is 2.5 minutes, this is how many
# intervals to skip before doing a graphing run
#skip_intervals 0

# Graph cutoff is how many k must be transfered by an
# ip before we bother to graph it
#graph_cutoff 1024

#Put interface in promiscuous mode to score to traffic
#that may not be routing through the host machine.
#promiscuous true

#Log data to cdf file htdocs/log.cdf
#output_cdf false

#Read back the cdf file on startup
#recover_cdf false

#Libpcap format filter string used to control what bandwidthd
see's
#Please always include "ip" in the string to avoid strange
```

```
problems
#filter "ip"

#Draw Graphs - This default to true to graph the traffic
bandwidthd is recording
#Usually set this to false if you only want cdf output or
#you are using the database output option.  Bandwidthd will use
very little
#ram and cpu if this is set to false.
#graph true

#Set META REFRESH seconds (default 150, use 0 to disable).
#meta_refresh 150
```

```
cd /etc/nginx/conf

htpasswd -c -d htpasswd user_name

server {
        listen 80;
        server_name monitor.example.com;
        root /var/www/bandwidthd/htdocs;
        index index.html;

        location / {
                try_files $uri $uri/ /index.html;
                auth_basic            "Login";
        auth_basic_user_file  htpasswd;
        }
}
```

http://monitor.example.com

CentOS rpmforge-release 安装注意事项

```
wget http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el5.rf.i386.rpm
```

```
rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm

yum install bandwidth
```

rpmforge-release 中有一个bandwidth 是一个内从测试软件 不是 bandwidthd

```
# yum search bandwidth
bandwidth.i386 : Artificial benchmark for measuring memory
bandwidth
```

2.3. source code

```
tar zxvf bandwidthd-2.0.1.tgz
cd bandwidthd-2.0.1
./configure --prefix=/srv/bandwidthd-2.0.1
make
make install
```

2.4. /etc/bandwidthd.conf

```
# 监控所有地址
subnet 0.0.0.0 0.0.0.0
# 监控某一段IP地址
subnet 10.0.0.0 255.0.0.0
subnet 192.168.0.0/16
subnet 172.16.0.0/12
```

3. NetFlow

查看设备是否发送Netflow包

```
$ sudo tcpdump -n udp port 2055
```

3.1. flow-tools - collects and processes NetFlow data

```
$ sudo apt-get install flow-tools
```

flow-capture

```
mkdir /opt/netflow
flow-capture -z 6 -n 143 -e 8928 -V 5 -w /opt/netflow 0/0/2055
```

NetFlow into MySQL with flow-tools

NetFlow into MySQL with flow-tools

创建netflow数据库，创建flows表

```
CREATE TABLE `flows` (
  `FLOW_ID` int(32) NOT NULL AUTO_INCREMENT,
  `UNIX_SECS` int(32) unsigned NOT NULL default '0',
  `UNIX_NSECS` int(32) unsigned NOT NULL default '0',
  `SYSUPTIME` int(20) NOT NULL,
  `EXADDR` varchar(16) NOT NULL,
  `DPKTS` int(32) unsigned NOT NULL default '0',
  `DOCTETS` int(32) unsigned NOT NULL default '0',
  `FIRST` int(32) unsigned NOT NULL default '0',
  `LAST` int(32) unsigned NOT NULL default '0',
  `ENGINE_TYPE` int(10) NOT NULL,
  `ENGINE_ID` int(15) NOT NULL,
  `SRCADDR` varchar(16) NOT NULL default '0',
  `DSTADDR` varchar(16) NOT NULL default '0',
  `NEXTHOP` varchar(16) NOT NULL default '0',
  `INPUT` int(16) unsigned NOT NULL default '0',
  `OUTPUT` int(16) unsigned NOT NULL default '0',
  `SRCPORT` int(16) unsigned NOT NULL default '0',
  `DSTPORT` int(16) unsigned NOT NULL default '0',
  `PROT` int(8) unsigned NOT NULL default '0',
  `TOS` int(2) NOT NULL,
  `TCP_FLAGS` int(8) unsigned NOT NULL default '0',
  `SRC_MASK` int(8) unsigned NOT NULL default '0',
  `DST_MASK` int(8) unsigned NOT NULL default '0',
  `SRC_AS` int(16) unsigned NOT NULL default '0',
  `DST_AS` int(16) unsigned NOT NULL default '0',
  PRIMARY KEY (FLOW_ID)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

创建数据库插入脚本

```
$ cat flow-mysql-export
#!/bin/bash

flow-export -f3 -u
"username:password:localhost:3306:netflow:flows" <
/flows/router/$1
```

获取Netflow信息，执行插入任务

```
mkdir -p /srv/flows/router
flow-capture -w /srv/flows/router -E5G 0/0/2055 -R
/srv/bin/flow-mysql-export
```

3.2. netams - Network Traffic Accounting and Monitoring Software

过程 8.1. 安装步骤

1. netams netams-web

```
$ sudo apt-get install netams netams-web
```

```
$ dpkg -s netams netams-web
```

2. NeTAMS administrator password

```
┌────────────────────┤ Configuring netams
│┌─────────────────────┐
│ Please enter password for "admin" user in NeTAMS
database.  │
│

│ NeTAMS administrator password:
│

│
*******____________________________________________________
│

│                           <Ok>
│

└──────────────────────────────────────────────────────────
───┘

┌───────────┤ Configuring netams ├────────────┐
│                                             │
│ Repeat password for NeTAMS user "admin":    │
│                                             │
│ *******__________________________________   │
```

```
                     <Ok>
```

如果你想重新配置安装过程可以运行下面命令

```
$ sudo dpkg-reconfigure netams netams-web
```

3. 基本配置

```
$ sudo vim /etc/default/netams
RUN="yes"
```

```
$ sudo cp /etc/netams/netams.conf
/etc/netams/netams.conf.old
$ sudo vim /etc/netams/netams.conf

$ sudo /etc/init.d/netams restart
```

```


$ cat /etc/apache2/conf.d/netams.conf
Alias /netams/images /usr/share/netams
Alias /netams/stat /var/lib/netams/stat

<Directory /var/lib/netams/stat/>
        Options -Indexes -FollowSymlinks

        DirectoryIndex index.html

        AllowOverride All
</Directory>

<Directory /usr/share/netams/>
```

```
        Options -Indexes -FollowSymlinks
        AllowOverride None
</Directory>
```

```
$ cat /etc/apache2/conf.d/netams-web.conf
ScriptAlias /netams/cgi-bin /usr/share/netams-web

# Uncomment the following if you have no netams package
installed
#Alias /netams/images /usr/share/netams-web/images

<Directory /usr/share/netams-web>

        Options -Indexes +FollowSymlinks

        AddHandler cgi-script .cgi

        AllowOverride None

# By default we deny access from other hosts. May be you
will need to configure
# mod_auth_basic or mod_auth_mysql.
        Order deny,allow
        Deny from All
        Allow from 127.0.0.1

</Directory>
```

4. .netamsctl.rc

```
$ vim ~/.netamsctl.rc
login=admin
password=123456
host=localhost


$ netamsctl "show version"
NeTAMS 3.4.3 (3475.1) buildd@yellow / Tue 06 Apr 2010
```

```
03:40:49 +0000
Run time  22 mins 6.5699 secs
System time:  22 mins 1.2800 secs
Average CPU/system load: 0.10%
Process ID: 23647 RES: 9212K
Memory allocated: 3640404 (23161), freed (31) (0 NULL)
[23130 used]
Total objects:
   Oids used: 9
   NetUnits: 4
   Policies: 3
   Services: 10
   Users: 1
   Connections: 1 active, 8 total

Services info:
 Storage ID=1 type mysql wr_q 0/0 rd_q 0/0
 Data-source ID=1 type LIBPCAP source eth0:0 loop 316382
average 4182 mcsec
    Perf: average skew delay 21580 mcsec, PPS: 77, BPS:
16788
Alerter 0 queue max: 255, current: 0
 Scheduled tasks: 1
```

netams-web

http://localhost/netams/stat/

http://localhost/netams/cgi-bin/login.cgi

4. Ntop

ntop - display network usage in web browser

4.1. Installation

Ubuntu

```
$ sudo apt-get install ntop
$ sudo apt-get install graphviz
```

设置管理员密码

```

  ┌────────────────────────────┤ Configuring ntop
├────────────────────────────┐
  │ Please choose a password to be used for the privileged user
"admin" in   │
  │ ntop's web interface.
│
  │
│
  │ Administrator password:
│
  │
│
  │
───────────────────────────────────────────────────────────────
─────────  │
  │
│
  │                                 <Ok>
│
  │
│
└──────────────────────────────────────────────────────────────
```

```
────────────┘


  ┌──────────────────────────┤ Configuring ntop
├───────────────────────────┐
  │ Please enter the same password again to verify that you
have typed it   │
  │ correctly.
│
  │
│
  │ Re-enter password to verify:
│
  │
│
  │
───────────────────────────────────────────────────────────────
────────  │
  │
│
  │                                <Ok>
│
  │
│
└──────────────────────────────────────────────────────────────
───────────┘
```

如果你忘记密码，可以使用下面命令重置密码

```
$ sudo ntop --set-admin-password
```

```
$ sudo /etc/init.d/ntop start
```

CentOS

5.x

```
wget http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el5.rf.i386.rpm
rpm -K rpmforge-release-0.5.2-2.el5.rf.i386.rpm
rpm -i rpmforge-release-0.5.2-2.el5.rf.i386.rpm
yum install ntop
```

设置管理员密码

```
# ntop -A
Tue May 22 13:03:34 2012  NOTE: Interface merge enabled by
default
Tue May 22 13:03:34 2012  Initializing gdbm databases


ntop startup - waiting for user response!


Please enter the password for the admin user:
Please enter the password again:
Tue May 22 13:03:40 2012  Admin user password has been set
```

备份配置文件

```
# cp /etc/ntop.conf /etc/ntop.conf.old
```

/etc/sysconfig/iptables

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --
dport 3000 -j ACCEPT
service iptables restart
```

启动ntop

```
# /usr/bin/ntop -d -L -u ntop -P /var/ntop --use-syslog=daemon
```

```
or
# /usr/bin/ntop -d -L -u ntop -P /var/ntop --skip-version-check
--use-syslog=daemon
```

/etc/init.d/ntop 脚本有bug无法启动，需要如下修改

```
# vim /etc/init.d/ntop
start () {
    echo -n $"Starting $prog: "
    #daemon $prog -d -L @/etc/ntop.conf
    daemon $prog  @/etc/ntop.conf
```

4.2. Web UI

http://localhost:3000/

4.3. Plugins

NetFlow

5. MRTG

5.1. CentOS 8 Stream

```
[root@localhost ~]# dnf search mrtg
Last metadata expiration check: 3:27:52 ago on Thu 26 Aug 2021
02:14:39 PM CST.
================================================================
======================== Name Exactly Matched: mrtg
================================================================
========================
mrtg.x86_64 : Multi Router Traffic Grapher
================================================================
============================ Name Matched: mrtg
================================================================
============================
pcp-import-mrtg2pcp.x86_64 : Performance Co-Pilot tools for
importing MTRG data into PCP archive logs

[root@localhost ~]# dnf install -y mrtg
```

默认配置文件

```
[root@localhost ~]# cat /etc/mrtg/mrtg.cfg
################################################################
######
# Multi Router Traffic Grapher -- Example Configuration File
################################################################
######
# This file is for use with mrtg-2.0
#
# Note:
#
# * Keywords must start at the begin of a line.
#
# * Lines which follow a keyword line which do start
#   with a blank are appended to the keyword line
#
# * Empty Lines are ignored
#
# * Lines starting with a # sign are comments.

# Where should the logfiles, and webpages be created?

# Minimal mrtg.cfg
#--------------------

HtmlDir: /var/www/mrtg
ImageDir: /var/www/mrtg
LogDir: /var/lib/mrtg
ThreshDir: /var/lib/mrtg
#Target[r1]: 2:public@myrouter.somplace.edu
#MaxBytes[r1]: 1250000
#Title[r1]: Traffic Analysis
#PageTop[r1]: <H1>Stats for our Ethernet</H1>
```

```
[root@localhost ~]# indexmaker --output=/var/www/mrtg/index.html
/etc/mrtg/mrtg.cfg
```

启用 mrtg

```
[root@localhost ~]# systemctl enable mrtg
Created symlink /etc/systemd/system/multi-
user.target.wants/mrtg.service →
/usr/lib/systemd/system/mrtg.service.
```

启动 mrtg

```
[root@localhost ~]# systemctl start mrtg
```

查看启动状态

```
[root@localhost ~]# systemctl status mrtg
● mrtg.service - Multi-router Traffic Grapher
   Loaded: loaded (/usr/lib/systemd/system/mrtg.service;
disabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-08-26 17:58:34 CST;
4s ago
 Main PID: 176231 (mrtg)
    Tasks: 1 (limit: 100608)
   Memory: 21.4M
   CGroup: /system.slice/mrtg.service
           └─176231 /usr/bin/perl -w /usr/bin/mrtg
/etc/mrtg/mrtg.cfg --lock-file /var/lock/mrtg/mrtg_l --
confcache-file /var/lib/mrtg/mrtg.ok

Aug 26 17:58:34 localhost.localdomain systemd[1]: Started Multi-
router Traffic Grapher.
```

Nginx 配置

```
[root@localhost conf.d]# cat
/etc/nginx/conf.d/monitor.netkiller.cn.conf
server {
    listen       192.168.30.13:80;
    server_name  192.168.30.13;

    access_log /var/log/nginx/monitor.netkiller.cn.access.log;
    error_log /var/log/nginx/monitor.netkiller.cn.error.log;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        root /var/www/mrtg;
        index  index.html;
```

```
            autoindex on;
    }
}
```

5.2. Ubuntu 安装

```
$ sudo apt-get install mrtg
$ sudo mkdir /etc/mrtg/
$ sudo sh -c 'cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--ifref=name --ifdesc=descr --show-op-down \
public@172.16.0.254 > /etc/mrtg/firewall.cfg'

$ sudo mkdir -p /var/www/mrtg
$ sudo indexmaker --output=/var/www/mrtg/firewall.html
/etc/mrtg/firewall.cfg
```

例 8.1. mrtg

5.3. CentOS 安装

```
# yum install mrtg
```

start

```
# env LANG=C /usr/bin/mrtg /etc/mrtg/mrtg.cfg
```

/etc/mrtg/mrtg.cfg

```
HtmlDir: /var/www/mrtg
ImageDir: /var/www/mrtg
LogDir: /var/lib/mrtg
ThreshDir: /var/lib/mrtg
#Target[r1]: 2:public@myrouter.somplace.edu
#MaxBytes[r1]: 1250000
#Title[r1]: Traffic Analysis
#PageTop[r1]: <H1>Stats for our Ethernet</H1>

Target[dell_3548_switch]:
ifInOctets.1&ifOutOctets.1:public@172.16.0.252
MaxBytes[dell_3548_switch]: 1250000
Title[dell_3548_switch]: Traffic Analysis
PageTop[dell_3548_switch]: <H1>Stats for our Ethernet</H1>
```

create mrtg.cfg

```
cp /etc/mrtg/mrtg.cfg /etc/mrtg/mrtg.cfg.old

cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--ifref=name --ifdesc=descr  --show-op-down \
public@172.16.0.252 > /etc/mrtg/mrtg.cfg
```

index.html

```
# indexmaker --output=/var/www/mrtg/index.html
/etc/mrtg/mrtg.cfg
```

5.4. 监控多个设备

```
cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--ifref=name --ifdesc=descr \
--subdirs=Dell6224 \
public@172.16.0.251 \
--ifref=name --ifdesc=descr \
--subdirs=Dell3548 \
public@172.16.0.252 \
--ifref=name --ifdesc=descr \
--subdirs=H3CS3600 \
public@172.16.0.253 > /etc/mrtg/mrtg.cfg

indexmaker --output=/var/www/mrtg/index.html /etc/mrtg/mrtg.cfg
```

5.5. 批量生成监控配置文件

```
for host in 253 252 251 250 249
do

cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
\
--ifref=name --ifdesc=descr \
--subdirs=Cisco-Switch-2960G-$host \
public@172.16.0.$host \
\
> /etc/mrtg/switch-2960-$host.cfg

indexmaker --output=/var/www/mrtg/switch-2960-$host.html
/etc/mrtg/switch-2960-$host.cfg

done
```

5.6. 图片尺寸

Xsize / Ysize

```
cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--global "Xsize[_]: 600" \
--global "Ysize[_]: 200" \
\
--ifref=name --ifdesc=descr \
--subdirs=Juniper-Firewall \
public@172.16.0.1 \
> /etc/mrtg/firewall.cfg
```

6. lvs-rrd

http://tepedino.org/lvs-rrd/

第 9 章 OpenTSDB

http://opentsdb.net/

第 10 章 Zipkin 分布式链路追踪