Ubuntu Server 从入门到精通

第15章：监控

监控

一切尽在掌握之中

- 规模带来的挑战
  - 每当系统规模扩大10倍，原有的技术栈则需要替代
  - 更多系统、设备、应用的数量，不同的部署地点、不同的业务需求
  - 系统问题不能及时得到反馈，响应处理不及时（被动响应）
  - 设备服务间存在复杂的依赖关系，问题定位困难（分层带来的困扰）
    - 网络问题造成应用无法访问
    - 数据库问题造成应用无法访问
    - 资源抢占造成后台进程被系统杀掉
  - 来自用户的问题反馈通常存在误导因素
  - 问题出现前的性能指标预示着问题即将发生
  - 系统缺乏统一监控告警机制，失控
  - 统一的监控系统，自动告警、问题处理

监控

一切尽在掌握之中

- 监控系统
  - 周期检查系统、服务是否可用
  - Nagios
  - Zabbix
  - Manageengine
  - Solarwind
- 监控技术
  - 基于客户端
    - 适用于通用操作系统
  - 基于SNMP协议
    - 普遍适用、硬件设备
    - 私有的协议扩展

监控

一切尽在掌握之中

- 监控系统
  - Nagios
  - Zabbix
  - Manageengine
  - Solarwind
- 监控技术
  - 基于客户端
    - 适用于通用操作系统
  - 基于SNMP协议
    - 普遍适用
    - 硬件设备
    - 私有的协议扩展

Nagios

一切尽在掌握之中

- nagios是开源的系统监视工具（core、xi）
  - 自动监视系统运行状态
  - 发现问题及时通知相关人
  - 支持客户端和SNMP兼容设备
- 两类监控对象
  - Host：物理的、虚拟的、网络设备、打印机等（可分组）
  - Service：系统功能（系统服务、资源占用CPU、内存、存储）
  - Service 至少关联到一个 Host

Nagios

一切尽在掌握之中

- 检查结果的 4 种状态
  - Ok
  - Warning
  - Critical
  - Unknown
  - 具体的性能指需定义为以上状态
- 基于插件的系统框架
  - Nagios 将要检查什么，以及warning、critical的标准交给插件
  - 所有检查工作由插件完成，并分析检查结果
  - 内建插件主要由 C 语言开发（编译安装）
  - 不支持的检查功能可自行开发插件（支持所有语言）

Nagios

一切尽在掌握之中

- 清晰的对象定义系统
  - Commands：插件之上的抽象层，将类似的操作分组处理
  - Time periods：应该或不应该执行监控操作的时间跨度
  - Hosts/Groups：一个或是一组主机（一个主机可以属于多个组）
  - Services：主机上需要监视的具体功能和资源（ＣＰＵ、存储、ＷＥＢ服务）
  - Contacts/Groups：联系人
  - Notifications：不同等级事件、时间通知谁什么内容
    - 何时以何种方式通知谁具体什么信息
    - 不是严格意义的对象，其他对象的结合
  - Escalations：通知升级，告警持续一定时间后升级

Nagios

一切尽在掌握之中

- Nagios 是一个依赖系统
  - 系统和服务依赖网络设备
  - 服务之间互相依赖，被依赖的服务故障时，依赖它的服务不再检查和告警
- 计划宕机
  - 维护性宕机调度时nagios不发告警（包含依赖）
  - 也可以通知计划宕机维护
- 软硬状态
  - 为避免随机和临时性故障告警，Nagios 状态区分软硬
  - 当前与之前检测状态不同时故障为软状态，相同则为应状态（重试次数）

Nagios 安装

一切尽在掌握之中

- Ubuntu 官方库包含 Nagios 3.X 版本
  - sudo apt install nagios3 nagios-plugins
  - 官方最新版本 4.4.2（手动编译安装）
  - 提供 WEB 访问方式（历史状态）
- 编译安装
  - C 编译器、C 语言开发库、OpenSSL实现WEB加密访问、MySQL存储历史数据、PHP、SNMP、图形组件
  - apt -y install wget gcc make binutils cpp libpq-dev libmysqlclient-dev libssl1.0.0 libssl-dev pkg-config libgd-dev libgd-tools perl libperl-dev libnet-snmp-perl snmp apache2 apache2-utils libapache2-mod-php unzip tar gzip php php-gd

Nagios 安装

一切尽在掌握之中

- 创建目录结构
  - mkdir /opt/nagios　　# 二进制、插件，程序安装目录
  - mkdir /var/nagios　　# 状态数据（历史数据）
  - mkdir /etc/nagios　　# 配置文件（安装过程生成）
- 创建用户和组帐号
  - groupadd nagios　　# 后台进程运行账号
  - groupadd nagioscmd　　# 与后台进程通信（WEB）
  - useradd -g nagios -G nagioscmd -d /opt/nagios nagios　　# 建帐号并加入组
  - usermod -G nagioscmd www-data　　# WEB需与 nagios 通信

Nagios 安装

一切尽在掌握之中

- 文件系统权限
  - chown nagios:nagios /opt/nagios /etc/nagios /var/nagios
- 下载源码
  - wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.2.tar.gz
  - wget https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
  - tar -xzf nagios-4.4.2.tar.gz
  - tar -xzf nagios-plugins-2.2.1.tar.gz

Nagios 安装

一切尽在掌握之中

- 编译安装
  - cd nagios-4.4.2/
  - ./configure --prefix=/opt/nagios --sysconfdir=/etc/nagios --localstatedir=/var/nagios --libexecdir=/opt/nagios/plugins --with-command-group=nagioscmd　　# 配置
  - make all　　# 编译
  - make install　　# 安装主程序、CGI、HTML等
  - make install-commandmode　　# 安装扩展命令
  - make install-config　　# 安装例子配置文件
  - make install-init　　# 创建系统服务文件

Nagios 安装

一切尽在掌握之中

- 验证配置文件
  - su -c '/opt/nagios/bin/nagios -v /etc/nagios/nagios.cfg'
- 编译安装插件
  - cd ../nagios-plugins-2.2.1/
  - sh configure --prefix=/opt/nagios --sysconfdir=/etc/nagios --localstatedir=/var/nagios --libexecdir=/opt/nagios/plugins
  - make all
  - make install

WEB 站点配置文件

一切尽在掌握之中

- vi /etc/apache2/conf-available/nagios.conf

```
ScriptAlias /nagios/cgi-bin /opt/nagios/sbin
Alias /nagios /opt/nagios/share
<Location "/nagios">
        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /etc/nagios/htpasswd.users
        require valid-user
</Location>
<Directory "/opt/nagios/share">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Require all granted
        Order allow,deny
        Allow from all
</Directory>
<Directory "/opt/nagios/sbin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Require all granted
        Order allow,deny
        Allow from all
</Directory>
```


WEB 站点配置文件

一切尽在掌握之中

- 加载模块并启动站点
  - a2enmod cgi
  - a2enmod auth_basic
  - a2enconf nagios
- 设置 WEB 登陆密码
  - htpasswd -c /etc/nagios/htpasswd.users nagiosadmin
  - systemctl restart apache2
  - systemctl restart nagios
- WEB 访问
  - http://ip/nagios

配置

一切尽在掌握之中

- 配置文件
  - /etc/nagios/
  - /etc/nagios/nagios.cfg　　# 主配置文件（启动时加载，其他配置的入口）
    - cfg_file　　# 单个的对象定义文件
    - cfg_dir　　# 对象定义文件目录（所有.cfg文件及子目录）
    - resource_file　　# 资源文件
  - 分类存储　　# 类型、功能、业务（易于统一开关）
- 目录结构
  - cd /etc/nagios
  - mkdir commands timeperiods contacts contactgroups hosts hostgroups services servicegroups

配置

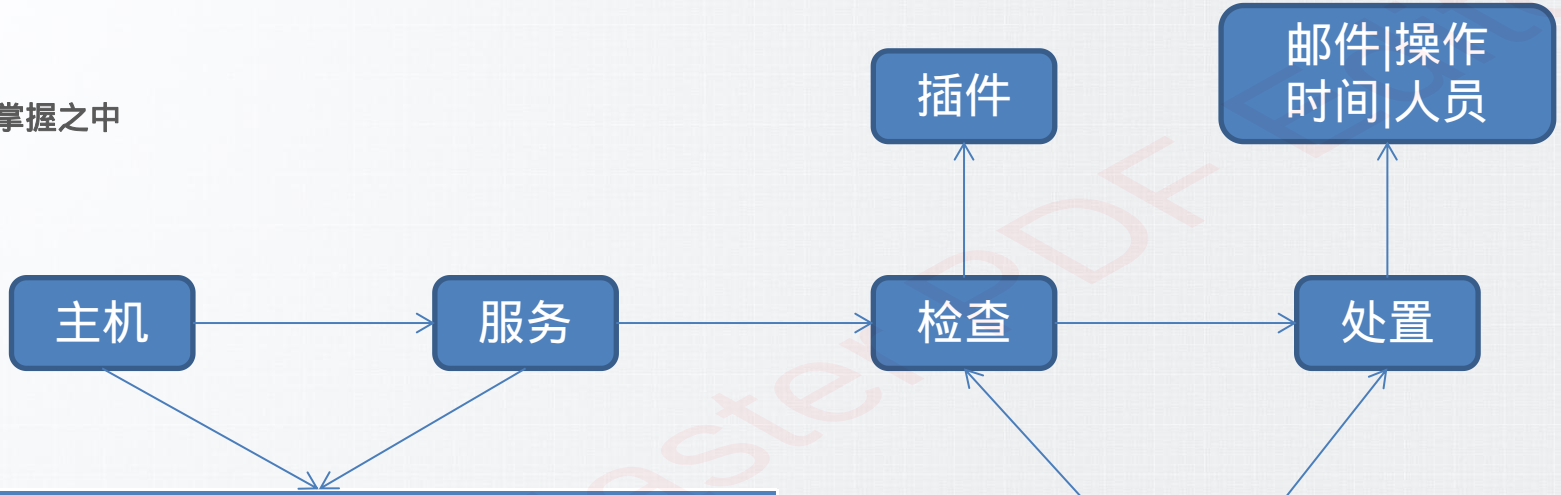
一切尽在掌握之中

- vi /etc/nagios/nagios.cfg

  cfg_dir=/etc/nagios/commands
  cfg_dir=/etc/nagios/timeperiods
  cfg_dir=/etc/nagios/contacts
  cfg_dir=/etc/nagios/contactgroups
  cfg_dir=/etc/nagios/hosts
  cfg_dir=/etc/nagios/hostgroups
  cfg_dir=/etc/nagios/services
  cfg_dir=/etc/nagios/servicegroups

配置

一切尽在掌握之中



宏（$HOSTADDRESS$）		
对象属性	自定义属性	资源文件
Host Service Contact	_起始 _MAC $_HOSTMAC$	Resource.cfg

宏

一切尽在掌握之中

```
define host{
        host_name                       somemachine
        address                         10.0.0.1
        check_command                   check-host-alive
        }
define command{
        command_name                    check-host-ssh
        command_line                    $USER1$/check_ssh -H
$HOSTADDRESS$
        }
```

- /opt/nagios/plugins/check_ssh -H 10.0.0.1

宏

一切尽在掌握之中

- 常用内建宏（全局变量）

HOSTNAME	HOSTADDRESS	HOSTDISPLAYNAME
HOSTSTATE	HOSTGROUPNAMES	LASTHOSTCHECK
LASTHOSTSTATE	SERVICEDESC	SERVICESTATE
SERVICEGROUPNAMES	CONTACTNAME	CONTACTALIAS
CONTACTEMAIL	CONTACTGROUPNAMES	

宏

一切尽在掌握之中

- 读取其他对象的宏
  - $CONTACTEMAIL:jdoe$ # jdoe 的邮件地址（非当前联系人）
- 自定义宏（对象属性字段）
  - _<variable> # 属性名称
  - $_HOST<variable>$ # 为HOST对象定义宏名
  - $_SERVICE<variable>$
  - $_CONTACT<variable>$
- 资源文件权限
  - chmode 600 /etc/nagios/resource.cfg
  - 内涵机密信息，应避免被 WEB 服务读取内容

宏

一切尽在掌握之中

- 自定义宏

```
define host{
        host_name               somemachine
        address                 10.0.0.1
        _MAC                    12:12:12:12:12:12
        check_command           check-host-by-mac
}
define command{
        command_name            check-host-by-mac
        command_line            $USER1$/check_hostmac -H $HOSTADDRESS$ -m
$_HOSTMAC$
}
```

HOST

一切尽在掌握之中

- 描述被监视对象
  - 短名 # short name
  - 描述名 # descriptive name
  - 地址 / 主机名
  - 何时以及如何监视
  - 故障时的联系人
  - 检查频率 / 重试次数
  - 如何发出告警
  - ……

HOST

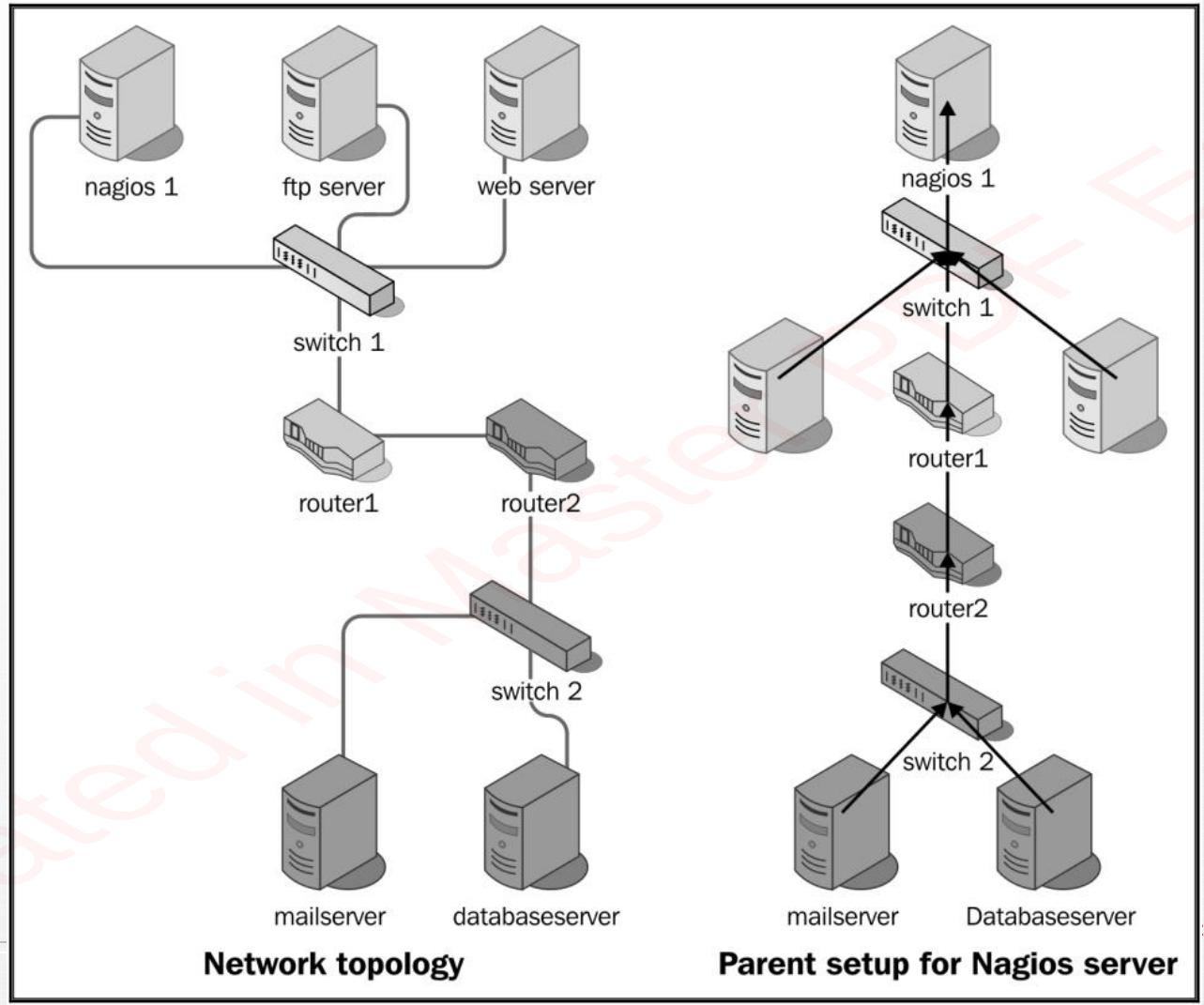
一切尽在掌握之中

```
define host{
        host_name                       linuxbox01              # 短名
        hostgroups                      linuxservers
        alias                           Linux Server 01         # 描述名
        address                         10.0.2.15
        check_command                   check-host-alive
        check_interval                  10                      # 分钟
        retry_interval                  1                       # 分钟
        max_check_attempts              5                       # 硬状态
才告警
        check_period                    24x7
        contact_groups                  linux-admins            # 或 contact
        notification_interval           30                      # 分钟
        notification_period             24x7
        notification_options            d, u, r                 # 何状态告警
        }
```

HOST

一切尽在掌握之中

- notification_options
  - d : DOWN
  - u : UNREACHABLE
  - r : Recovery (UP)
  - f : starts and stops flapping
  - s : scheduled downtime starts or ends
- 默认假设 HOST 状态为 UP
- parents # 依赖（通常为交换机 / 路由器）
  - Parent 处于hard down 状态时，host 状态为 unreachable，不再执行检测



Host Group

一切尽在掌握之中

- 每个 Host 可以同时属于多个 Group
  - 唯一不重复的短名 # short name

```
define hostgroup{
        hostgroup_name          linux-servers
        alias                   Linux servers
        members                 linux01,linux02
        hostgroup_members               group01,group02
        }
```


Service

一切尽在掌握之中

- NFS、FTP服务、存储空间、CPU 负载等
- Service 永远绑定于一个运行状态的 HOST
- 每个 HOST 至少要定义一个 Service
- 通过唯一的描述名标识
- 定义何时、如何进行检查（running）
- 定义告警方式
- 文件命名 /etc/nagios/services/<host>-<service>.cfg

Service

一切尽在掌握之中

```
define service{
        host_name                       localhost,srv1      # <hostgroup_name>
        service_description                     www                 # 唯一标识
        check_command                   check_http
        check_interval                  10
        check_period                    24x7
        retry_interval                  1
        max_check_attempts              3
        notification_interval                   30
        notification_period                     24x7
        notification_options                    w,c,u,r
        contact_groups                  web_admins
        }
```

Service

一切尽在掌握之中

- notification_options
  - w : WARNING
  - u : UNKNOWN
  - c : CRITICAL
  - r : Recovery (OK)
  - f : starts / stops flapping
  - s : scheduled downtime starts / ends

Service

一切尽在掌握之中

- 排除检查主机（！）

```
define service{
        hostgroup_name          linux-servers
        host_name               !srv01, srv02
        service_description     SSH
        (...)
        }
```

Service group

一切尽在掌握之中

- Service Group 成员是多个 <host>,<service>对

```
define servicegroup{
        servicegroup_name       databaseservices
        alias                   All services related to databases
        members                 srv01,mysql,srv101,
pgsql,aix01,db2
        }
```

Service group

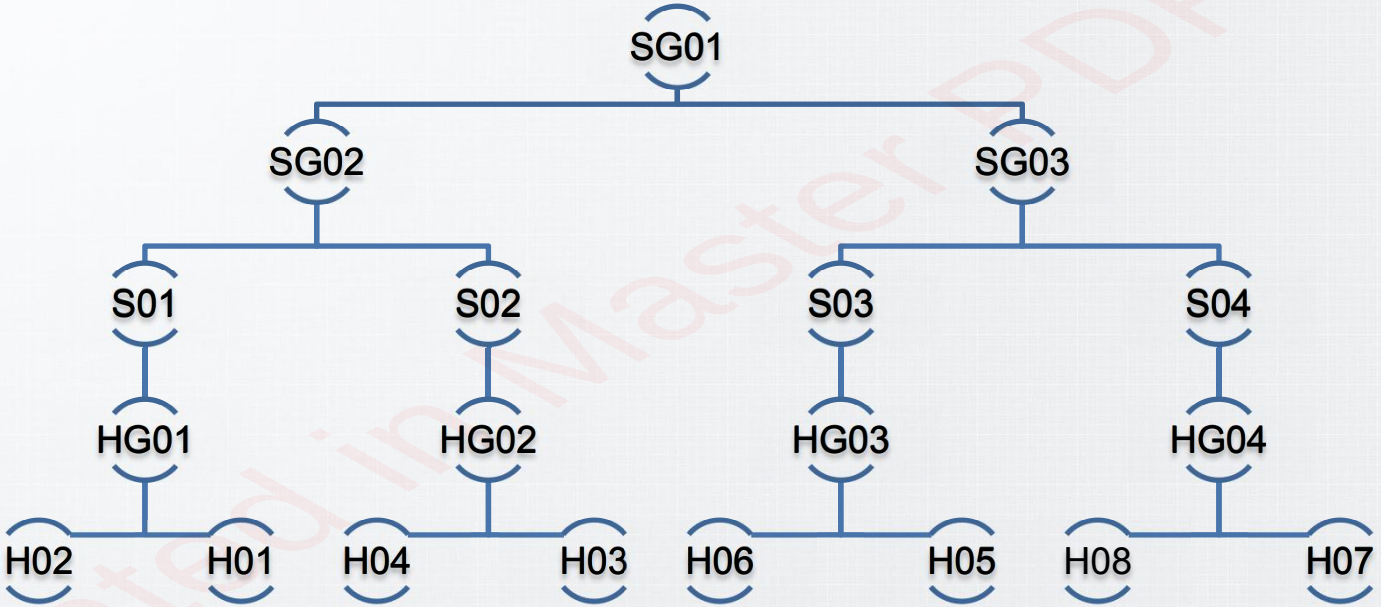
一切尽在掌握之中

- 另一定义组的方式

```
define servicegroup{
        servicegroup_name       databaseservices
        alias                   All services related to databases
        }
define service{
        host_name               linuxbox01
        service_description     mysql
        check_command           check_ssh
        servicegroups           databaseservices
        }
```


Service group

一切尽在掌握之中



Command

一切尽在掌握之中

- 如何检查 Host / Service（H/S 使用 Command）
- 如何问题告警 / 事务处理
- 两个参数：名称、命令行
- 命令执行插件或自定义命令
- 命令行会使用宏、参数等（$ARG1$, $ARG2$ ... $ARG32$ ）

Command

一切尽在掌握之中

- 定义 command

```
define command{
        command_name  check-host-alive
        command_line     $USER1$/check_ping -H $HOSTADDRESS$ -w
        3000.0,80% -c 5000.0,100% -p 5
        }
define host{
        host_name        somemachine
        address          10.0.0.1
        check_command  check-host-alive
}
```

Command

一切尽在掌握之中

- 带参数定义 command

```
define command{
        command_name  check-host-alive
        command_line    $USER1$/check_ping -H $HOSTADDRESS$
        -w $ARG1$ -c $ARG2$ -p 5
        }
define host{
        host_name       somemachine
        address         10.0.0.1
        check_command  check-host-alive-limits!3000.0,80%!5000.0,100%
        }
```

Time periods

一切尽在掌握之中

- 执行检查、告警的时间区段

```
define timeperiod{
        timeperiod_name         workinghours
        alias                   Working Hours, from Monday to Friday
        monday                  09:00-17:00
        tuesday                 09:00-17:00
        wednesday               09:00-17:00
        thursday                09:00-17:00
        friday                  09:00-17:00
        exclude                 first-mondays
        }
```

Time periods

一切尽在掌握之中

- 日期格式优先级（按时间颗粒度排序）
  - YYYY-MM-DD # 具体日期 2018-11-11
  - July 4 # 每年指定日期
  - day 14 # 每月指定日期
  - Monday 1 April # 具体月指定星期（4月第一个星期一）
  - Monday 1 # 每月指定星期（每个月第一个星期一）
  - Monday # 每星期一

```
define timeperiod{
        timeperiod_name     weekends
        alias               Weekends all day long
        saturday            00:00-24:00
        sunday              00:00-24:00
        }
```

Time periods

一切尽在掌握之中

```
define timeperiod{
        timeperiod_name         24x7
        alias                   24 hours a day 7 days a week
        monday                  00:00-24:00
        tuesday                 00:00-24:00
        wednesday               00:00-24:00
        thursday                00:00-24:00
        friday                  00:00-24:00
        saturday                00:00-24:00
        sunday                  00:00-24:00
        }
```

Contacts

一切尽在掌握之中

```
define contact{
        contact_name                    zs
        alias                           zhang san
        email                           zs@lab.com
        contactgroups                   admins,nagiosadmin
        host_notification_period        workinghours
        service_notification_period     workinghours
        host_notification_options       d,u,r
        service_notification_options            w,u,c,r
        host_notification_commands              notify-host-by-email
        service_notification_commands   notify-service-by-email
        }
```

Contacts

一切尽在掌握之中

- host_notification_options
  - d : DOWN
  - u : UNREACHABLE
  - r : Recovery (UP)
  - f : starts / stops Flapping
  - s : Scheduled downtime starts / ends
  - n : Not receive any notifications

Contacts

一切尽在掌握之中

- service_notification_options
  - w : WARNING
  - u : UNKNOWN
  - c : CRITICAL
  - r : Recovery (OK)
  - f : the service starts and stops flapping
  - n : the person will not receive any service
  - notifications

Contact Group

一切尽在掌握之中

- 指定时间联系指定人群
  - 硬件问题联系硬件管理员，服务问题联系系统管理员

```
define contactgroup{
        contactgroup_name       linux-admins
        alias                   Linux Administrators
        members                 jdoe,smith
        contactgroup_members
        }
```

- 验证配置文件
  - 重启服务/系统前验证配置，避免服务宕机
  - /opt/nagios/bin/nagios -v /etc/nagios/nagios.cfg

继承和模板

一切尽在掌握之中

- 定义新对象时可重用模板定义的参数，简化管理
- 不加 register 的模板对象将被视为普通对象进行监控
- 通过 use 指令调用模板定义对象
- 定义对象时可使用多个模板，按模板先后顺序确定优先级
- 模板可层级继承调用，定义新的模板

继承和模板

一切尽在掌握之中

```
define host{
        name                    generic-server
        check_command    check-host-alive
        check_interval       5
        retry_interval       1
        max_check_attempts          5
        check_period        24x7
        notification_interval               30
        notification_period 24x7
        notification_options                d,u,r
        register             0
        }
```

继承和模板

一切尽在掌握之中

```
define host{
        use                     generic-server
        host_name               linuxbox01
        alias                   Linux Server 01
        address                 10.0.2.1
        contact_groups          linux-admins
        }
```


WEB界面

一切尽在掌握之中

- 查看性能、故障、状态、历史信息
- 支持修改部分设置
- 状态、报告、系统
- Nagios Exchange（社区站点）
  - 包含不同类型的插件和扩展
  - http://exchange.nagios.org/directory/Addons/Frontends-(GUIs-and-CLIs)/Web-Interfaces

Nagiosgraph 性能图示

一切尽在掌握之中

- 基于RRDtool实现历史数据的图形化显示（曲线图表）
- 由 perl 语言编写
- 安装依赖包
  - apt -y install libcgi-pm-perl librrds-perl libgd-gd2-perl rrdtool perl libgd-perl
- 下载源码安装
  - curl -sSL https://sourceforge.net/projects/nagiosgraph/files/latest/download | tar xzv
  - cd nagiosgraph-1.5.2
  - perl install.pl

  Modify the Nagios configuration? [n] y
  Modify the Apache configuration? [n] y
  Path of Apache configuration directory? /etc/apache2/sites-enabled

Nagiosgraph 性能图示

一切尽在掌握之中

- 修改站点配置文件
- vi /etc/apache2/sites-enabled/nagiosgraph.conf
  - 替换： Allow from all --> Require all granted
  - 删除： order allow,deny
- 访问测试
  - http://192.168.20.11/nagiosgraph/cgi-bin/show.cgi

集成

一切尽在掌握之中

- vi /etc/nagios/objects/templates.cfg

```
define service {
    name                            local-service
    use                             generic-service
    max_check_attempts              4
    check_interval                  5
    retry_interval                  1
    action_url                              /nagiosgraph/cgi-bin/show.cgi?host=$HOSTNAME$&service=$SERVICEDESC$&geom=1000x200
    register                        0
}
```

集成

一切尽在掌握之中

- 重启服务
  - systemctrl restart apache2
  - systemctrl restart nagios.service

集成

一切尽在掌握之中

- MRTG监视nagios的运行状态
  - apt install mrtg
  - cp ~/nagios-4.4.2/sample-config/mrtg.cfg /etc/nagios/
  - mkdir /opt/nagios/share/stats
  - vi /etc/nagios/mrtg.cfg
    WorkDir: /opt/nagios/share/stats
- 启始运行
  - env LANG=C mrtg /etc/nagios/mrtg.cfg
- 创建页面
  - indexmaker /etc/nagios/mrtg.cfg --output=/opt/nagios/share/stats/index.html
- 周期运行
  - vi /etc/cron.d/nagiostats
    */5 * * * * root env LANG=C /usr/bin/mrtg /etc/nagios/mrtg.cfg

集成

一切尽在掌握之中

- 访问测试
  - http://192.168.20.11/nagios/stats/
- 启始页改为service
  - vi /opt/nagios/share/index.php
    $url="cgi-bin/status.cgi?host=all&limit=0";

集成

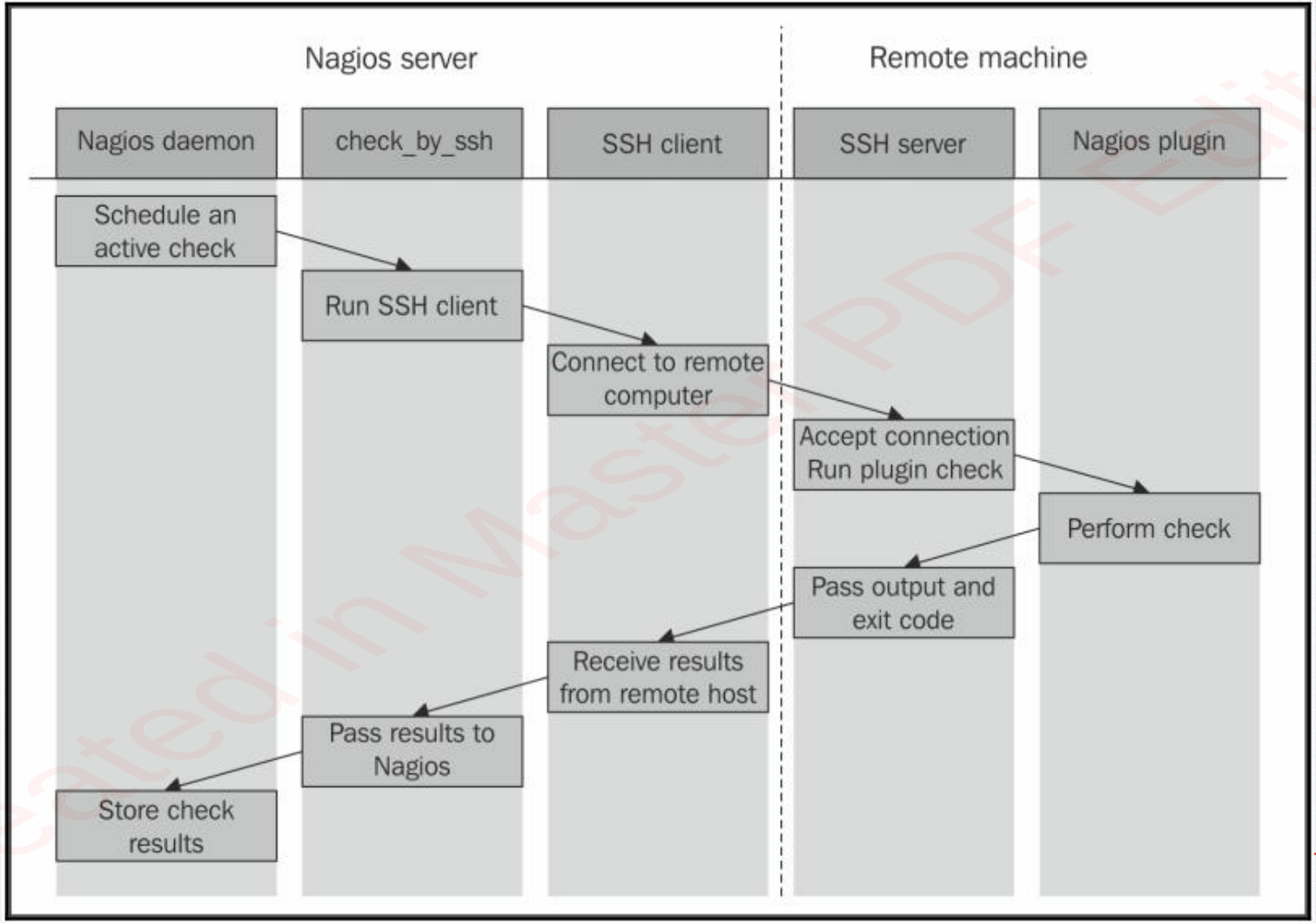
一切尽在掌握之中

- 菜单
  - vi /opt/nagios/share/side.php
  -
  - External Tools
  -
  - <ul class="navsectionlinks">
  - <a href="/nagios/stats" target="<?php echo $link_target;?>">Nagiostats
  - <a href="/nagiosgraph/cgi-bin/show.cgi" target="<?php echo $link_target;?>">Nagiosgraph
  -
  -
  -

远程监控

一切尽在掌握之中

- SSH 远程监控
  - 监视远程主机的CPU、内存、存储等
  - 利用SSH在远程主机上执行插件并返回结果及退出代码
  - SSH 基于密钥的身份认证
  - check_by_ssh 插件建立SSH连接，指定主机名和具体执行的命令



远程监控

一切尽在掌握之中

- Nagios client 安装配置
  - apt -y install gcc make binutils cpp
  - wget https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
  - tar -xzf nagios-plugins-2.2.1.tar.gz
  - cd nagios-plugins-2.2.1/
  - sh configure --prefix=/opt/nagios --sysconfdir=/etc/nagios --localstatedir=/var/nagios --libexecdir=/opt/nagios/plugins
  - make all
  - make install
- Nagios Server 生成密钥对
  - su -s /bin/bash nagios
  - ssh-keygen

远程监控

一切尽在掌握之中

- Nagios client 创建账号
  - useradd -d /opt/nagios nagios
  - chown nagios:nagios /opt/nagios
  - chmod 0700 /opt/nagios
  - passwd nagios
- Nagios Server 拷贝密钥
  - ssh -v nagios@192.168.2.1
  - ssh-copy-id nagios@192.168.1.1
- 测试（nagios_server）
  - /opt/nagios/plugins/check_by_ssh -H 192.168.20.12 -C "/opt/nagios/plugins/check_uptime"
  - /opt/nagios/plugins/check_by_ssh -H 192.168.20.12 -C "/opt/nagios/plugins/check_disk -w 15% -c 10% -p '/'"

远程监控

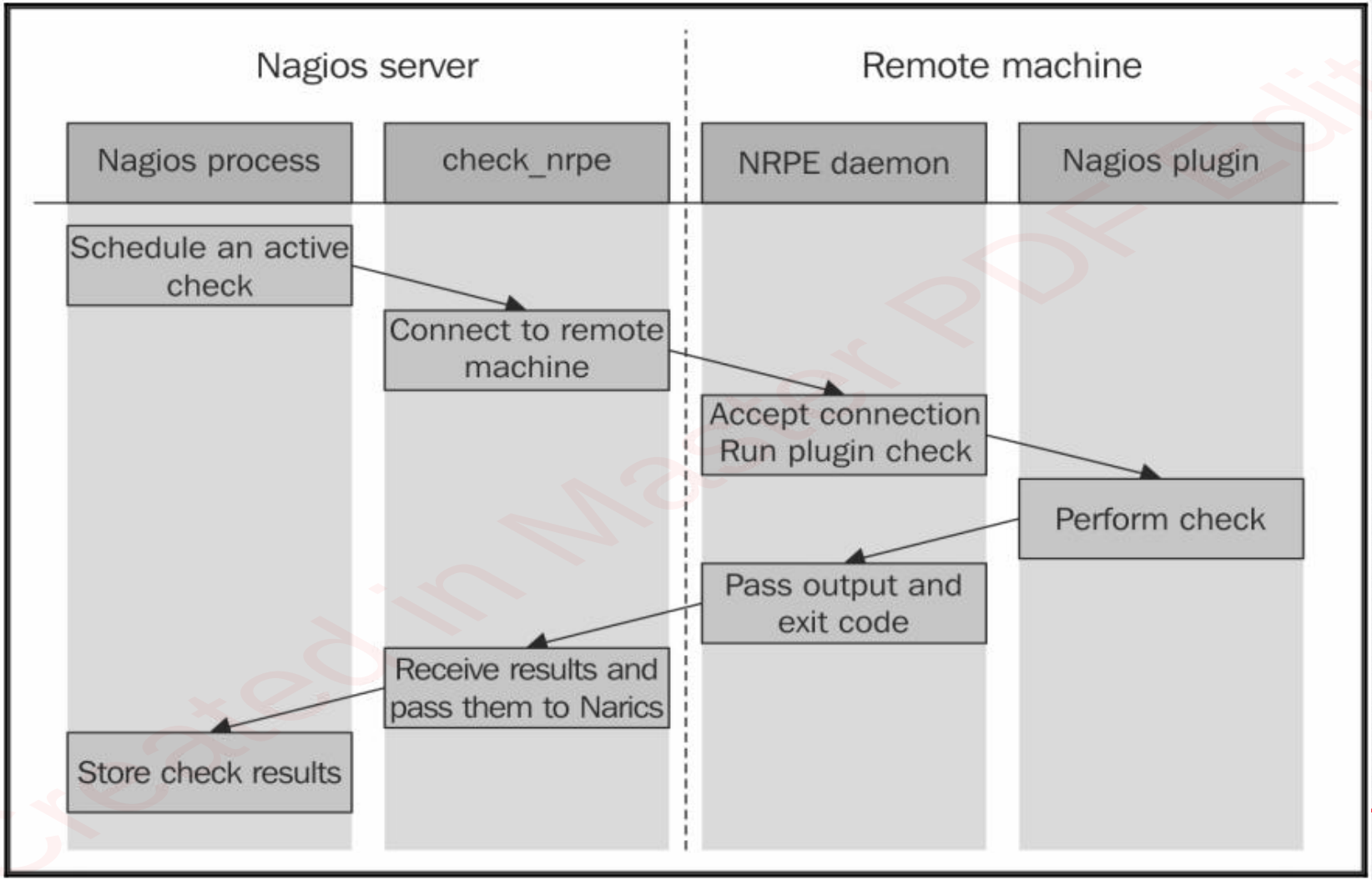
一切尽在掌握之中

- Nagios 配置
  - define command
  - {
  - command_name check_swap_by_ssh
  - command_line $USER1$/check_by_ssh -H $HOSTADDRESS$ -C "$USER1$/check_swap -w $ARG1$ -c $ARG2$"
  - }
  - define service
  - {
  - use generic-service
  - host_name !localhost
  - hostgroup_name linux-servers
  - service_description SWAP
  - check_command check_swap_by_ssh
  - }

远程监控

一切尽在掌握之中

- NRPE 远程监控
  - 客户端/服务器架构（check_nrpe 插件、NRPE daemon）
  - 通信支持加密(SSL over TCP)
  - 通信流量小于 SSH、节省CPU
  - 只允许运行特定命令，不会造成任意指令执行(SSH会)。
  - 默认工作端口 TCP 5666
- 安装依赖包
  - apt install gcc make binutils cpp pkg-config libc6-dev libssl-dev openssl
- 库安装（S / C）
  - apt install nagios-nrpe-server nagios-nrpe-plugin



远程监控

一切尽在掌握之中

- 编译安装
  - wget https://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz
  - tar zxvf nrpe-2.15.tar.gz
  - cd nrpe-2.15
  - sh configure \
  - --sysconfdir=/etc/nagios \
  - --libexecdir=/opt/nagios/plugins \
  - --prefix=/opt/nagios \
  - --localstatedir=/var/nagios \
  - --with-nrpe-user=nagios \
  - --with-nrpe-group=nagios \
  - --with-nagios-user=nagios \
  - --with-nagios-group=nagios \
  - --with-ssl-lib=/usr/lib/x86_64-linux-gnu \
  - --enable-ssl

远程监控

一切尽在掌握之中

- make all
- Nagios 服务器（NRPE 客户端）
  - make install-plugin
- Nagios 客户端（NRPE 服务端）
  - make install-daemon
  - mkdir /etc/nagios
  - make install-daemon-config
- 配置NRPE服务器
  - useradd -d /opt/nagios nagios
  - mkdir /opt/nagios
  - chown nagios:nagios /opt/nagios
  - passwd -l nagios

远程监控

一切尽在掌握之中

- 启动脚本
  - vi /etc/init.d/nrpe
  - chmod 0755 /etc/init.d/nrpe
  - update-rc.d nrpe
- 配置文件
  - vi /etc/nagios/nrpe.cfg
    - allowed_hosts=192.168.20.11
    - command[check_disk_sys]=/opt/nagios/plugins/check_disk -w 20% -c 10% -p /
    - 命令只在被监控的目标上配置，因此难于维护

远程监控

一切尽在掌握之中

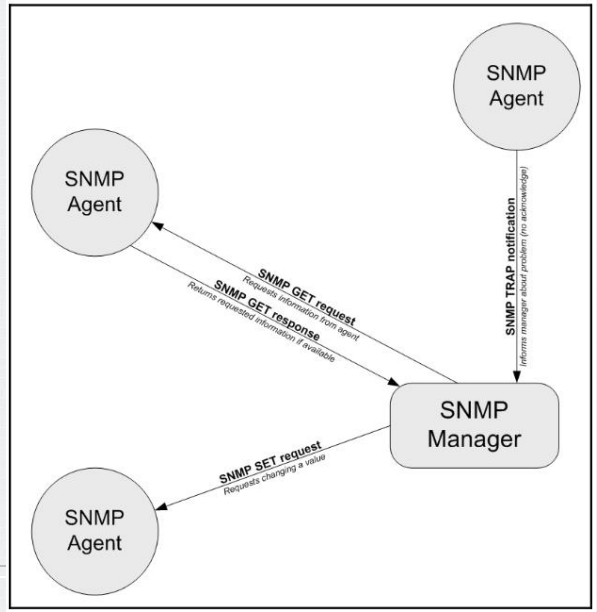
- 手工测试
  - /opt/nagios/plugins/check_nrpe -H 192.168.2.52 -c check_users
  - /usr/lib/nagios/plugins/check_nrpe -H 192.168.2.52 -c check_load
- 配置 nagios 使用 NRPE

```
define command
{
        command_name    check_swap_nrpe
        command_line    $USER1$/check_nrpe -H "$HOSTADDRESS$" -c
        "check_swap"
}
```

远程监控

一切尽在掌握之中

- SNMP（Simple Network Management Protocol）
  - 工业标准，（原则上）所有设备厂商都支持
  - 适合非通用操作系统的硬件设备（无法安装插件、nrpe）
  - 统一的方法获取和设置设备参数
  - 标准的层次化信息分组访问方式，称为管理信息库
  - Management Information Base(MIB)
  - MIB定义可访问的属性，即标准OID的对应关系
  - 厂商可自定义OID（不兼容）
  - OID （Object identifier）
  - 协议端口：UDP 161 / 162



远程监控

一切尽在掌握之中

- OID

Identifier	Description
1	**iso**: iso standard tree
3	**org**: Organizations; this node is a placeholder for all national and international organizations
6	**dod**: Department of Defense; this is the node for the U.S. Department of Defense
1	**internet**: Subnode for the Internet; since originally the Internet was a project for U.S. military defense, its placeholder is under the **dod** subtree
2	**mgmt**: Systems management node
1	**mib-2**: Management Information Base, version 2 root node
1	**system**: Operating system information
5	**sysName**: Name of this machine; usually a fully qualified domain name
0	Index of the elements; in this case it is always 0

远程监控

一切尽在掌握之中

- 被监视端运行 agent 进程侦听端口，监视端称为 SNMP manager
- get / set 是从 manager 向 agent 的通信过程
- trap 是 agent 主动向 manager 通告信息的通信过程
- 版本：v1、v2、v2c、v2u、v3
  - v1 基于 IP / community（public/private）的安全机制
  - v2、v2c 增加了 getbulk（获取节点下所有数据）、inform（trap——请求manager确认ACK）
  - v2u 基于用户的身份验证安全机制（但不包含v1、v2c其他安全机制，很少用）
- v3改进的安全模型，认证、隐私、访问控制
- 检查设备支持SNMP什么版本（proxy用于多版本之间的协调转换）

远程监控

一切尽在掌握之中

- 安装 Net-SNMP 工具包、MIB 文件库（manager）
  - apt install snmp snmp-mibs-downloader
  - snmpget -v 1 -c public 192.168.2.2 iso.org.dod.internet.mgmt.mib-2.system.sysName.0
  - snmpwalk -v 1 -c public 192.168.2.2 1.3.6.1.2.1.1
  - 图形化工具 TkIned、 mbrowse
- 安装 SNMP Agent
  - apt install snmpd
  - vi /etc/snmp/snmpd.conf
    - community / IP
- 监控 windows 系统
  - snmpwalk -v 2c -c public 192.168.1.1

远程监控

一切尽在掌握之中

- nagios 基于 snmp 监控
  - 插件 check_snmp
  - /opt/nagios/plugins/check_snmp -H 10.0.0.1 -P 2c -C public -o SNMPv2-MIB::sysLocation.0 -s "yuanfh"
  - /opt/nagios/plugins/check_snmp -H 10.0.0.1 -P 2c -C public -o SNMPv2-MIB::sysContact.0 -r "@"
  - /opt/nagios/plugins/check_snmp -H 10.0.0.1 -P 2c -C public -o HOST-RESOURCES-MIB::hrSystemProcesses.0 -w 0:20 -c 0:30
  - /opt/nagios/plugins/check_ifoperstatus -H 10.0.0.1 -C public1 -k 65539
  - /opt/nagios/plugins/check_ifstatus -H 10.0.0.1 -v 2c -C public1 -u 65539
  - route print

nagios 配置文件

一切尽在掌握之中

```
define command
{
        command_name            check_snmp
        command_line            $USER1$/check_snmp -P 1 -H $HOSTADDRESS$ -o
        $ARG1$ $ARG2$
}
define service
{
        use                     generic-service
        hostgroup_name          snmp-aware
        service_description     Processes
        check_command           check_snmp!HOST-RESOURCES-
        MIB::hrSystemProcesses.0!-w 0:50 -c 0:100
}
```

远程监控

一切尽在掌握之中

```
define host
{
        use                             generic-host
        host_name                       linuxbox01
        address                         10.0.0.1
        _SNMPVERSION                    2c
        _SNMPCOMMUNITY                  public
}

define command
{
        command_name check_snmp
        command_line $USER1$/check_snmp -H $HOSTADDRESS$ -o $ARG1$ -P
$_HOSTSNMPVERSION$ -C $_HOSTSNMPCOMMUNITY$ $ARG2$
}
```