

# 監視ソフト Nagios をインストールする。

## 今回の目標

1 台で各ネットワーク上のサーバを監視したい。つまり ,監視側 PC には ,2 枚の LAN カードを装着し , 各 IP アドレスを付与。

そして , 2 つのネットワークとサーバの死活確認を行う。

## 実験内容

3 台の PC を用意。(うち , 1 台は , Nagios をインストール)

監視される 2 台の PC は , 各々別 IP , 別ネットワークとします。

監視側 PC ( CentOS5 , Nagios インストール )

- ・ Nagios とは
  - ・ ネットワークに接続された機器を定期的にチェックし , 障害発生に決められた手段で管理者に通知します ( メール )
  - ・ 加えて , Web から障害の状況や監視対象の状態変化が確認できます。
- ・ 特徴
  - ・ 監視時刻のスケジューリングが可能 ( 決められた時間しか監視しない )
  - ・ 監視項目には , HTTP , FTP , POP3 , SMTP 等デフォルトで用意されています。
- ・ Nagios ホームページ
- ・ <http://www.nagios.org/>

## 実験結果

- ・ 例 : Ping による死活確認
  - ・ デフォルト仕様上 , 5 分間隔で監視 , 一度 , 状況が変化すると , 1 分間隔で設定数の再試行を行う。そのあと , 改善されないと管理者宛へメール ( デフォルト )
  - ・ Ping は , RTA が 100ms 以上 , 500ms 未満 , あるいは , 20% 以上 60% 未満のパケットロスで Warning を表示し , それ以上の時は , Critical ( 赤色 ) に変化する。
  - ・ よって , 即応性は , 乏しいといえる。
  - ・ ( /usr/local/nagios/etc/localhost.cfg のしきい値を厳格にすれば , もう少し即応性が高まる )
- ・ ブラウザ側は , 90 秒毎にアップデートされる。
- ・ ( /usr/local/nagios/etc/cgi.cfg の設定値を変えれば , 10 秒でも可能 )
- ・ nagios.cfg の external command check interval の値を 1s にすれば , 1 秒間隔で設定可能かも。
- ・ 現在では , 2 分程度の遅れで監視される PC の死活確認ができます。

インストール資料 ( OS は , CentOS5 を使用。 )

(0) ユーザおよびグループの作成

- Nagios は、デーモンとして稼動します。ユーザ権限で稼動させる為、専用のユーザとインストール先ディレクトリを作成します。
- Web サーバから Nagios に対してコマンドを発行して Nagios を操作することになる為、
- Web サーバのユーザを Nagios のコマンドを処理する専用のグループのメンバーに設定します。
- httpd ユーザを新しく作成した nagcmd メンバーに加えます。

```
# /usr/sbin/groupadd nagios
# /usr/sbin/useradd -d /usr/local/nagios -g nagios -m nagios
# /usr/sbin/groupadd nagcmd
# /usr/sbin/usermod -G nagcmd httpd
# /usr/sbin/usermod -G nagcmd nagios
```

### (1)gcc のインストール (後で必要となる為)

- # yum -y install gcc

### (2)Nagios は、PNG、JPEG ライブラリ等必要な為、各種インストール。

- PNG ライブラリは、<http://www.libpng.org/pub/png> より取得 (libpng-1.2.20.tar.gz )

```
$ ./configure --prefix=/usr/local
$ cd
$ make
$ su
# make install
```

- JPEG ライブラリは、<http://www.iijg.org> より取得 (jpegsrc.v6b.tar.gz )

```
$ tar zvxvf jpegsrc.v6b.tar.gz
$ cd
$ ./configure --prefix=/usr/local --enable-shared --enable-static
$ make
$ su
# make install
```

- エラーがでたので、/usr/local/man/man1/cjpeg.1 を作成した。

### (3)GD ライブラリのインストール。

- GD ライブラリは、画像を操作するためのライブラリ群。
- 一般に、GD ライブラリ本体と、perl モジュールとしての GD.pm をひとまとめにして言うことが多い。
- ここで必要な GD ライブラリは、1.6.3 以上です。
- <http://www.boutell.com/> だったが、[http://www.libgd.org/Main\\_Page](http://www.libgd.org/Main_Page) に変更。

- wget でファイルをダウンロード

- wget <http://www.boutell.com/gd/http-gd-2.0.32.tar.gz> から wget <http://www.libgd.org/releases/gd-2.0.35.tar.gz> へ変更。

```
$ wget http://www.libgd.org/releases/gd-2.0.35.tar.gz
```

- ファイルを展開します。

```
$ tar xvfz gd-2.0.32.tar.gz
```

- ・展開したディレクトリに入ります。

```
$ cd gd-2.0.32
```

- ・make の準備をします。

```
$ ./configure --with-png=/usr/local/lib --with-jpeg=/usr/local/bin --prefix=/usr/local
$ make
$ make test
$ su
# make install
```

#### (4) Nagios をインストール

- ・Nagios が rpm インストールされていないかどうか確認

```
# rpm -qa | grep nagios
```

- ・( Nagios からの wget は、省略 )

```
$ tar xvfz nagios-2.9.tar.gz で展開
$ cd nagios-2.9
$ ./configure --prefix=/usr/local/nagios --with-command-group=nagcmd --with-gd-lib=/usr/local/lib
$ make all
# make install
# make install-init
# make install-commandmode
# make install-config
```

#### (5) Nagios Plugin Development のインストール

- ・ダウンロードしたファイルは、nagios-plugins-1.4.9.tar.gz でした。

```
$ ./configure --prefix=/usr/local/nagios --with-cgiurl=/usr/local/nagios/cgi-bin
$ make
# make install
```

#### (6) Nagios の Web インターフェースのための設定を行う。

- ・http.conf(/etc/httpd/conf/http.conf) に下記の内容を追記 ( CGI が動作するように設定を行う )

```
ScriptAlias /nagios/cgi-bin/ /usr/local/nagios/sbin/
<Directory "/usr/local/nagios/sbin/">
    AllowOverride AuthConfig
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

Alias /nagios/ /usr/local/nagios/share/
<Directory "/usr/local/nagios/share/">
    AllowOverride AuthConfig
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

#### (7) 管理画面用ユーザおよびパスワードの作成

- Web インターフェースに BASIC 認証を使います。
- /usr/local/nagios/share/ と /usr/local/nagios/sbin/ ディレクトリに .htaccess を作成する。
- .htaccess の設定 ( chmod 604 で設定 )

```
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
require valid-user
```

- .htaccess の「AuthUserFile」で指定した「/usr/local/nagios/etc/htpasswd.users」に htpasswd コマンドを利用してユーザを追加します。
- 今回は、「nagios」というユーザを追加します。
- ( <http://www.shtml.jp/htaccess/basic.html> で説明がありますので、そちらを見たほうが早い )

- htpasswd で Web インターフェース用アカウント登録

```
# su nagios
$ htpasswd -c /usr/local/nagios/etc/htpasswd.users nagios
New password:
Re-type new password:
Adding password for user nagios
```

## (8) Nagios 基本動作設定

- 設定するファイル群
  - メイン設定ファイル：nagios.cfg
  - CGI 設定ファイル：cgi.cfg
  - リソース設定ファイル：resource.cfg
  - 監視設定ファイル：localhost.cfg
- インストールされたサンプル設定をコピーして使用します。

```
# su nagios
$ cd /usr/local/nagios/etc
$ cd nagios.cfg-sample nagios.cfg
$ cd cgi.cfg-sample cgi.cfg
$ cd resource.cfg-sample resource.cfg
```

- 後 , localhost.cfg と commands.cfg をコピー
- nagios.cfg の修正箇所

```
37 #cfg_file=/usr/local/nagios/etc/checkcommands.cfg   これはなかった。
40 #cfg_file=/usr/local/nagios/etc/misccommands.cfg   これはなかった。
131 check_external_commands=1
```

- cgi.cfg の修正箇所

```
116 authorized_for_system_information=nagios
128 authorized_for_configuration_information=nagios
141 authorized_for_system_commands=nagios
154 authorized_for_all_services=nagios
155 authorized_for_all_hosts=nagios
168 authorized_for_all_service_commands=nagios
169 authorized_for_all_host_commands=nagios
```

- Nagios デーモンの状態を Web インターフェースから把握する設定 ( cgi.cfg )

```
65 nagios_check_command=/usr/local/nagios/libexec/check_nagios /usr/local/nagios/var/status.dat 5
' /usr/local/nagios/bin/nagios'
```

- ・ リソース設定ファイル ( resource.cfg )

特に無し

- ・ 設定の整合性チェックと Nagios の起動

設定ファイルのチェックは、nagios コマンドに -v の引数をつけて nagios.cfg を指定します。

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

- ・ 上記コマンドを行うと、下の表示がされれば、OK。エラーが出た場合には、設定内容をチェックすること。

```
Total Warnings: 0
Total Errors: 0
```

Things look okay -No serious problems were detected during the pre-flight check

- ・ 後、各 \*.cfg の属性 ( 664 ) に設定しておくこと。

## (9) Nagios の起動

```
$ su -
# /etc/init.d/nagios start
```

## (10) Nagios を自動起動にする

```
# cd /etc/init.d/
# chmod 755 nagios
# /sbin/chkconfig --add nagios
# /sbin/chkconfig --list nagios
# /sbin/chkconfig --level 35 nagios on
# /sbin/chkconfig --list nagios
```

## (11) 監視対象 PC を追加する。

- ・ /usr/local/nagios/etc/localhost.cfg に監視対象ホストを追加する ( 以下長いですが、ご勘弁を )

```
#####
# LOCALHOST.CFG - SAMPLE OBJECT CONFIG FILE FOR MONITORING THIS MACHINE
#
# Last Modified: 11-27-2006
#
# NOTE: This config file is intended to service as an *extremely* simple
#       example of how you can create your object configuration file(s).
#
#####

#####
#####
#
```

```

#TIME PERIODS
#
#####
#####

# This defines a timeperiod where all times are valid for checks,
# notifications, etc. The classic "24x7" support nightmare. :-)

define timeperiod{
    timeperiod_name 24x7
    alias            24 Hours A Day, 7 Days A Week
    sunday           00:00-24:00
    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
}

# 'workhours' timeperiod definition
define timeperiod{
    timeperiod_name workhours
    alias           "Normal" Working Hours
    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
}

# 'nonworkhours' timeperiod definition
define timeperiod{
    timeperiod_name nonworkhours
    alias           Non-Work Hours
    sunday          00:00-24:00
    monday          00:00-09:00,17:00-24:00
    tuesday          00:00-09:00,17:00-24:00
    wednesday        00:00-09:00,17:00-24:00
    thursday         00:00-09:00,17:00-24:00
    friday           00:00-09:00,17:00-24:00
    saturday         00:00-24:00
}

# 'none' timeperiod definition
define timeperiod{
    timeperiod_name none
    alias           No Time Is A Good Time
}

#####
#####
#
# COMMANDS
#
#####
#####

# NOTE: Sample command definitions can now be found in the sample commands.cfg
#       file

#####
#####
#
# CONTACTS
#
#####
#####

# In this simple config file, a single contact will receive all alerts.
# This assumes that you have an account (or email alias) called
# "nagios-admin" on the local host.

define contact{

```

```

        contact_name      nagios-admin
        alias              Nagios Admin
        service_notification_period 24x7
        host_notification_period 24x7
        service_notification_options w,u,c,r
        host_notification_options d,r
        service_notification_commands notify-by-email
        host_notification_commands host-notify-by-email
        email              nagios-admin@localhost
    }

#####
#####
#
# 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                nagios-admin
}

#####
#####
#
# HOSTS
#
#####
#####

# Generic host definition template - This is NOT a real host, just a template!

define host{
    name                  generic-host ; The name of this host template
    notifications_enabled 1             ; Host notifications are enabled
    event_handler_enabled 1             ; Host 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_period    24x7         ; Send host notifications at any time
    register                0           ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL
HOST, JUST A TEMPLATE!
}

# Linux host definition template - This is NOT a real host, just a template!

define host{
    name                  linux-server ; The name of this host template
    use                   generic-host ; This template inherits other values from the generic-host template
    check_period          24x7         ; By default, Linux hosts are checked round the clock
    max_check_attempts    10          ; Check each Linux host 10 times (max)
    check_command          check-host-alive ; Default command to check Linux hosts
    notification_period    workhours   ; Linux admins hate to be woken up, so we only notify during the day
    ; Note that the notification_period variable is being overridden from
    ; the value that is inherited from the generic-host template!
    notification_interval  120         ; Resend notification every 2 hours
    notification_options    d,u,r      ; Only send notifications for specific host states
    contact_groups         admins      ; Notifications get sent to the admins by default
    register                0          ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
}

# Since this is a simple configuration file, we only monitor one host - the
# local host (this machine).

define host{
    use                   linux-server ; Name of host template to use
    ; This host definition will inherit all variables that are defined

```

```

        ; in (or inherited by) the linux-server host template definition.
        host_name      localhost
        alias           localhost
        address         127.0.0.1
    }

#
# 2007.9.20 追加しました。
# ここで、監視対象ホストの名前、IP アドレス等を定義します。
#

define host{
    use linux-server
    host_name server01
    alias server01
    address ***.***.***.***
}

define host{
    use linux-server
    host_name server02
    alias server02
    address ***.***.***.***
}

#

#####
#####
#
# HOST GROUPS
#
#####
#####

# We only have one host in our simple config file, so there is no need to
# create more than one hostgroup.

#
# こちらは、監視対象ホストのグループを設定します。
#members は、上記で設定した host_name, alias 名で登録します。
#

define hostgroup{
    hostgroup_name test
    alias Test Servers
    members localhost,server01,server02
}

#####
#####
#
# SERVICES
#
#####
#####

# Generic service definition template - This is NOT a real service, just a template!

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
    is_volatile 0 ; The service is not volatile
    register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL
SERVICE, JUST A TEMPLATE!

```



```

    }

# Local service definition template - This is NOT a real service, just a template!

define service{
    name                local-service        ; The name of this service template
    use                  generic-service      ; Inherit default values from the generic-service definition
    check_period         24x7                ; The service can be checked at any time of the day
    max_check_attempts   4                  ; Re-check the service up to 4 times in order to
determine its final (hard) state
    normal_check_interval 5                  ; Check the service every 5 minutes under normal
conditions
    #
    # デフォルトでは、5分ですが、今回は、1分に変更しています（上記は、5分のままです。）
    #
    retry_check_interval 1                  ; Re-check the service every minute until a hard state
can be determined
    contact_groups        admins            ; Notifications get sent out to everyone in the
'admins' group
    notification_options  w,u,c,r          ; Send notifications about warning, unknown, critical, and recovery
events
    notification_interval 60                ; Re-notify about service problems every hour
    notification_period   24x7             ; Notifications can be sent out at any time
    register              0                ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL
SERVICE, JUST A TEMPLATE!
}

#
#
# 以下より、各監視サービスの設定に、上記で定義した監視対象ホスト(host_name, alias 名)を追加していきま
す。
#
#

Define a service to "ping" the local machine

define service{
    use                  local-service        ; Name of service template to use
    host_name            localhost,server01,server02
    #
    # ここで、監視対象ホストを登録しています。
    #
    service_description  PING
    check_command         check_ping!100.0,20%!500.0,60%
    #
    # ping コマンドの閾値を設定しています。
    #
}

# Define a service to check the disk space of the root partition
# on the local machine. Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use                  local-service        ; Name of service template to use
    host_name            localhost,server01,server02
    service_description  Root Partition
    check_command        check_local_disk!20%!10%!/
}

# Define a service to check the number of currently logged in
# users on the local machine. Warning if > 20 users, critical
# if > 50 users.

define service{
    use                  local-service        ; Name of service template to use
    host_name            localhost,server01,server02
    service_description  Current Users
    check_command        check_local_users!20!50
}

# Define a service to check the number of currently running procs
# on the local machine. Warning if > 250 processes, critical if
# > 400 users.

define service{

```

```

        use                                local-service          ; Name of service template to use
        host_name                          localhost,server01,server02
        service_description                 Total Processes
check_command      check_local_procs!250!400!RSZDT
    }

```

# Define a service to check the load on the local machine.

```

define service{
    use                                local-service          ; Name of service template to use
    host_name                          localhost,server01,server02
    service_description                 Current Load
check_command      check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

```

以上です。2007.9.21.

## その他の監視ソフト

ping による IP 死活確認であれば , ExPing というソフトが簡単です。

- ・ 監視間隔は , 1 分単位

<http://www.woodybells.com/exping.html>

by 有限会社ケアイエム ( <http://www.keiaiemu.com/> )