

## **NAGIOS**

*Nagios* é um sistema de monitoração das aplicações de rede. Ele vigia hosts e serviços que você especificar, alertando quando o serviço ou host estiver "down" e também quando os mesmos estiverem "up", trabalhando de forma pró-ativa.

### **Preparando o ambiente para testes:**

Faça o download do último pacote do *Nagios* e do último pacote dos plugins para um diretório temporário. Usaremos /tmp/nagios.

```
# mkdir /tmp/nagios  
# cd /tmp/nagios
```

Acesse:

- <http://www.nagios.org/download/>

E baixe os pacotes [nagios-2.5.tar.gz](#) e [nagios-plugins-1.4.3.tar.gz](#).

Primeiro nós iremos instalar o gerenciador de aplicações Nagios. Vamos descomprimir o arquivo tar.gz:

```
# tar xvzf nagios-2.5.tar.gz
```

Depois de descomprimido o arquivo tar.gz, iremos para o diretório nagios-2.5:

```
# cd nagios-2.4
```

Agora precisamos decidir onde iremos instalar o nosso sistema. Você pode instalar o Nagios em qualquer lugar, mas o melhor é instalar na localização padrão (/usr/local/nagios), porque a documentação original sempre se refere para ele. Dessa forma será fácil resolver problemas que possam acontecer.

Crie o diretório onde você gostaria de instalar o Nagios:

```
# mkdir /usr/local/nagios
```

Neste ponto precisamos criar um usuário e grupo que o Nagios irá usar para carregar o serviço. Você pode usar o "root" para isto, mas não é recomendado como segurança. Em geral para uma fácil manutenção, decidi dedicar um novo usuário e grupo para essa função. O usuário e grupo terão o nome de "nagios".

```
# useradd nagios  
# groupadd nagios
```

Uma vez criado o usuário e o grupo, podemos iniciar o processo de instalação.

## **INSTALAÇÃO DO NAGIOS**

Primeiro precisamos especificar alguns parâmetros e criar o Makefile que será usado para compilar e instalar o software:

```
# ./configure --prefix=/usr/local/nagios --with-cgiurl=/nagios/cgi-bin --  
with-htmurl=/nagios/ --with-nagios-user=nagios --with-nagios-  
grp=nagios
```

Geralmente quando instalamos o *Nagios* no diretório padrão (/usr/local/nagios), não precisamos utilizar todos os parâmetros acima, mas é sempre bom direcionar para o lugar correto.

Uma vez a configuração completa, irá aparecer um sumário de todos os parâmetros que foram usados durante a configuração. Tenha certeza que tudo está OK, senão execute o "configure" novamente com as opções corretas.

Existe também uma alta probabilidade de aparecer um aviso que a biblioteca GD está faltando. Consulte o [site da Boutell](#) para instalar essa biblioteca e execute o "configure" novamente com a opção --with-gd-lib e --with-gd-inc para especificar o exato diretório da biblioteca GD.

Caso não funcione, não se preocupe, o Nagios funcionará mesmo sem essa biblioteca. Essa biblioteca é usada somente em alguns CGI's que criam imagens dinâmicas para a estatística de serviço. A aplicação é ainda muito útil sem estes gráficos.

Agora iremos compilar o software. Usaremos as seguintes opções (se você não estiver logado com o root, se logue agora):

```
# make all
```

Se durante o longo processo de compilação não acontecer nenhum erro, receberemos a mensagem "Compile Finished" no final da compilação.

Iremos executar três comandos para instalar vários componentes nos seus devidos lugares. Primeiro iremos instalar o programa principal, arquivos e

diretórios no `/usr/local/nagios`.

## # make install

Vamos agora instalar o script de inicialização para que o Nagios seja carregado automaticamente durante o boot. Esse script permite também que utilizemos a opção `start`, `stop`, `restart` e `reload`. Exemplo: `"service nagios start"`.

## # make install-init

No meu sistema (Debian 8.3) coloquei o script no diretório `/etc/rc.d/init.d`.

Se você der uma olhada dentro do diretório `/usr/local/nagios` você verá que existem quatro diretórios:

## # ls /usr/local/nagios

```
bin sbin share var
```

O diretório `"bin"` contém um simples arquivo chamado `"nagios"`, que é o centro dos pacotes. Esta aplicação não é atualmente monitorada. O diretório `"sbin"` contém o CGI script que será usado na interface web. Dentro do diretório `"share"` você encontrará o HTML da documentação e finalmente o diretório `"var"`, que é onde o Nagios armazenará informações quando iniciado.

Em geral para habilitar o uso do Nagios você precisa de um conjunto de arquivos de configuração. Estes arquivos estão dentro do diretório `"etc"`, no qual será criado quando você executar o seguinte comando:

## # make install-config

## INSTALAÇÃO DOS PLUGINS

Nesse ponto a instalação do *Nagios* está completa, mas ele não está com suas totais funções, falta a monitoração das aplicações. Esta função é responsável por checar se um particular serviço está funcionando. Para habilitarmos essa função é necessário instalar os plugins separadamente. Faça o download da última versão dos plugins no site:

- <http://www.nagios.org>

Depois do download, faça a descompactação do arquivo e entre no diretório criado pelo comando `"tar"`:

```
# cd nagiosplug-1.4.3  
# ./configure --prefix=/usr/local/nagios --with-nagios-user=nagios --  
with-nagios-group=nagios
```

Você talvez receba notificações sobre a perda de problemas ou módulos Perl enquanto esteja executando o configure. Não tem problema, a menos que você especifique uma aplicação que necessite desses módulos.

Uma vez terminada a configuração, compile todos os plugins:

```
# make all
```

Se nenhum erro for reportado, você pode instalar os plugins:

```
# make install
```

Os plugins serão instalados no diretório "libexec" dentro do diretório do Nagios /usr/local/nagios/libexec. Entre nesse diretório, depois execute "./check\_ssh -h" para saber como o "check\_ssh" trabalha.

Usando esses comandos você pode rodar manualmente a checagem de qualquer serviço, mas iremos automatizar o nosso processo. Por exemplo:

```
# ./check_ssh 200.146.2.1 (IP ilustrativo)  
SSH ok - protocol version 1.99 - server version
```

## **CONFIGURAÇÃO DO NAGIOS**

Depois da instalação do *Nagios* e dos plugins, estamos quase prontos para iniciar a monitoração dos nossos servidores, mas antes precisamos configurar alguns arquivos.

```
# cd /usr/local/nagios/etc  
# ls
```

O comando "ls" irá mostrar todos os arquivos \*.cfg-sample, precisamos renomear esses arquivos para \*.cfg. Criem um diretório "sample" e copie todos os arquivos \*.cfg-sample para esse diretório, uma cópia de segurança, em seguida renomeie todos os arquivos \*.cfg-sample para \*.cfg.

Como a nossa configuração é simples e não iremos entrar em maiores detalhes, apague os arquivos "dependencies.cfg" e "escalation.cfg", iremos criar dois arquivos em branco para substituir os mesmos:

```
# touch dependencies.cfg
# touch escalations.cfg
```

## CONTEÚDO DO ARQUIVO HOSTS.CFG

No arquivo hosts.cfg devemos colocar os servidores que desejamos monitorar. Edite o arquivo hosts.cfg com o seu editor preferido. Exemplo:

```
# vi hosts.cfg
```

```
# 'servidor1' host definition
define host{
    use          generic-host      ; Name of host template to use

    host_name    servidor1
    alias        Web Server #1
    address      192.168.0.1
    check_command check-host-alive
    max_check_attempts 5
    notification_interval 1
    notification_period 24x7
    notification_options d,u,r
}

```

Para cada "host" você irá criar um conjunto das linhas acima, identificando o "hostname" do servidor e o "IP address".

## CONTEÚDO DO ARQUIVO HOSTGROUP.CFG

Edite o arquivo hostgroup.cfg, use o seu editor preferido. Exemplo:

```
# vi hostgroup.cfg
```

```
# 'email-servers' host group definition
define hostgroup {

```

```
hostgroup_name  http-servers
alias           Web Servers
contact_groups  http-admins
members        servidor1, servidor2
}
```

Agora necessitamos adicionar os hosts para o hostgroup usando o arquivo acima. Acima nós definimos um novo "hostgroup" e associamos o "http-admins" com grupo de contato para ele. Agora iremos ver o arquivo "contactgroup".

## CONTEÚDO DO ARQUIVO CONTACTGROUP.CFG

```
# 'http-admins' contact group definition
define contactgroup {
    contactgroup_name  http-admins
    alias              Web Administrators
    members            Eduardo,Wanderson
}
```

Nós definimos o grupo de contato "http-admins" e adicionamos dois members, "Wanderson" e "Eduardo". Estas configurações asseguram que ambos os usuários serão notificados quando alguma coisa errada acontecer com os servidores que as pessoas do grupo "http-admins" são responsáveis.

A próxima etapa será configurarmos as informações dos contatos e notificações para estes usuários.

## CONTEÚDO DO ARQUIVO CONTACTS.CFG

```

# 'Wanderson' contact definition
define contact {
    contact_name      hugo
    alias             Wanderson Nunes
    service_notification_period  24x7
    host_notification_period  24x7
    service_notification_options  w,u,c,r
    host_notification_options  d,u,r
    service_notification_commands  notify-by-email
    host_notification_commands  host-notify-by-email
    email             wandnrosa@gmail.com
}

# 'Eduardo' contact definition
define contact{
    contact_name      Eduardo
    alias             Eduardo
    service_notification_period  24x7
    host_notification_period  24x7
    service_notification_options  w,u,c,r
    host_notification_options  d,u,r
    service_notification_commands  notify-by-email
    host_notification_commands  host-notify-by-email
    email             eduardo.avila.ortiz@gmail.com
}

```

Para criarmos os detalhes dos usuários o arquivo é o contacts.cfg, conforme exemplo acima.

Depois de criados os hosts, grupo dos hosts, contatos e grupos dos contatos, iremos identificar qual o serviço que queremos gerenciar em cada host.

No exemplo abaixo estamos habilitando o gerenciamento através do PING para saber se o servidor está no ar e o serviço HTTP para sabermos se o serviço Web está funcionando corretamente.

## CONTEÚDO DO ARQUIVO SERVICES.CFG

```
# Service definition
```

```
define service {  
    use      generic-service ; Name of service template to use  
  
    host_name      servidor1  
    service_description  HTTP  
    is_volatile    0  
    check_period   24x7  
    max_check_attempts  3  
    normal_check_interval  1  
    retry_check_interval  1  
    contact_groups    http-admins  
    notification_interval  2  
    notification_period  24x7  
    notification_options    w,u,c,r  
    check_command      check_http  
}
```

```
# Service definition
```

```
define service {  
    use      generic-service ; Name of service template to use  
  
    host_name      servidor1  
    service_description  PING  
    is_volatile    0  
    check_period   24x7  
    max_check_attempts  5  
    normal_check_interval  1  
    retry_check_interval  1  
    contact_groups    http-admins  
    notification_interval  2  
    notification_period  24x7  
    notification_options    c,r  
    check_command      check_ping!100.0,20%!500.0,60%
```



```
}
```

Agora que terminamos a configuração dos hosts, contatos e serviços, podemos iniciar o serviço do Nagios para iniciarmos a monitoração dos nossos servidores.

Por experiência própria com o Red Hat 8.0, carregue o serviço com a opção "reload" e não "start". Ainda não sei porque, mas a opção "start" quando carregada pela primeira vez gera alguns erros que não permite carregar o serviço.

## INTERFACE WEB

Bem, embora o *Nagios* esteja iniciado e monitorando os nossos servidores e enviando notificações por e-mail (o seu MTA deverá estar configurado corretamente para enviar mensagens) quando ocorrer algum problema, é muito válido configurarmos a interface web para uma melhor interação com essa monitoração.

Para isso precisamos ter um Web Server instalado na máquina que o Nagios esteja instalado. Em nosso exemplo será utilizando o *Apache*, pois é um dos mais Web Servers utilizados no mundo inteiro.

### # vim /etc/httpd/conf/httpd.conf

Adicione as linhas abaixo ao final do arquivo:

```
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">
  Options None
  AllowOverride AuthConfig
  Order allow,deny
  Allow from all
</Directory>
```

Está configuração cria um alias "/nagios/cgi-bin/" e direciona ele para um script CGI no diretório "sbin" do Nagios. Dessa forma podemos carregar a interface web do Nagios assumindo que a página principal do seu web

server está em `http://localhost`, digitando no nosso browser `http://localhost/nagios`.

Mas ainda não carrega a página do Nagios, pois ainda não criamos os usuários que podem acessá-lo. Crie um arquivo `".htaccess"` no diretório `/usr/local/nagios/sbin` com o seguinte conteúdo:

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

Não esqueça de criar o arquivo com atributo de oculto, com um "ponto" antes do nome do arquivo.

Agora precisamos criar os usuários e as suas respectivas senhas, execute o seguinte comando:

```
# htpasswd -c /usr/local/nagios/etc/htpasswd.user hugo  
New password: *****  
Re-type new password: *****
```

Para criar outros usuários você não deve utilizar o parâmetro `"-c"`, pois ele é utilizado para criar o arquivo `htpasswd.user` se o mesmo não existir. Se você utilizar o parâmetro `"-c"` e o arquivo já existir, o mesmo será substituído.

Ótimo agora você poderá entrar na interface web do Nagios para monitorar online todos os serviços dos seus servidores.

Uma dica muito importante, todos os acessos aos scripts CGI são controlados pelo arquivo `/usr/local/nagios/etc/cgi.cfg`. Lá você determina o que cada usuário pode acessar.

Pronto, o seu software de monitoração de hosts e serviços está funcionando. Para controle de mais serviços, aconselho você entrar no diretório `/usr/local/nagios/libexec` e testar os scripts (`./check_ftp -h`).