

Introdução

Preparando o ambiente

Pré requisitos

Baixando o Nagios Core

Compilar o nagios core

Criando uma conta de usuário no Nagios

Manipuladores de eventos do Nagios

Reiniciando o Nagios/Apache

Instalando os plugins padrões do Nagios

Pré requisitos

Compilando os plugins

Reiniciando o daemon do Nagios

Introdução

Este tutorial aborda o processo de instalação do Nagios Core e Nagios plugins no Ubuntu 18.04. O Nagios é um sistema de monitoramento que permite os administradores de rede e pessoal do NOC identificar e resolver problemas de infraestrutura de TI antes que eles afetem processos críticos da empresa.

O Nagios é oficialmente patrocinado pela [Nagios Enterprises](#), ele tem a função de monitorar toda a sua infraestrutura de TI para garantir que sistemas, aplicativos, serviços e processos de negócios estejam funcionando corretamente. Em caso de falha, o Nagios irá alertar a equipe responsável sobre o evento ocorrido para que eles possam atuar e corrigir o problema antes que isso venha impactar os processos de negócio.

O Nagios possui algumas formas de alerta após ser detectado uma mudança de estado de um dispositivo gerenciado, o tipo mais comum é o alerta visual no dashboard do Nagios, mas você pode configurar para que seja enviado e-mail e até mesmo integrar o envio de mensagens no telegram usando Nagios.

Todo processo de instalação foi realizado seguindo a [documentação oficial](#) do Nagios.

- Links

[Nagios Core - Installing Nagios Core From Source](#)

[Table of Contents](#)

Preparando o ambiente

Inicialmente vamos baixar alguns pacotes que vamos usar durante a utilização do servidor.

Segue o comando usado:

```
sudo apt-get install -y unzip zip tcpdump openssh-server mtr nmap perl python  
python3 vim curl htop wget bc gawk snmp snmpd libnet-snmp-perl
```

```
# Descrição das aplicações que vão ser instaladas:

# UNZIP = Desarquivador para arquivos .zip;
# ZIP = Arquivador para arquivos .zip;
# TCPDUMP = Analizador de tráfego de Rede;
# openssh-server = Servidor SSH (Secure Shell), para acesso seguro a
partir de máquinas remotas;
# MTR = Ferramenta traceroute de tela cheia em ncurses e X11;
# NMAP = Mapeador de Rede (fazer verificação de portas;
# PERL = Linguagem de extração e relatórios prática de Larry Wall;
# PYTHON = Linguagem interativa de alto nível orientada a objetos (versão
2 - padrão);
# PYTHON3 = Linguagem interativa de alto nível orientada a objetos (versão
3 - padrão);
# VIM = Editor VI melhorado;
# CURL = Ferramenta de linha de comando para transferir dados com sintaxe
de URL;
# HTOP = Visualizador de processos interativo;
# WGET = Gerenciador de download;
# BC = Calculadora de linha de comando;
# GAWK = Linguagem AWK;
# SNMP = Aplicação cliente para utilizar o protocolo SNMP (Gerenciamento
de redes);
# SNMPD = Aplicação servidor para utilizar o protocolo SNMP (Gerenciamento
de redes);
# LIBNET-SNMP-PERL = Script de conexões SNMP.
```

Pré requisitos

Para podermos compilar o Nagios precisamos atender aos pré requisitos, caso contrário, teremos erro durante a compilação, abaixo segue o comando que instala as dependências do Nagios.

```
sudo apt-get install -y autoconf gcc libc6 php7.2 php make libapache2-mod-
php7.2 apache2 build-essential xinetd

# Descrição das aplicações que vão ser instaladas:
# Pré requisitos informados na documentação do Nagios
# AUTOCONF = Construtor automático de script configure (necessário
para compilar);
# GCC = Compilador C;
# LIBC6 = Biblioteca C: Bibliotecas compartilhadas;
# PHP = Linguagem de script incorporada em HTML do servidor (padrão);
# MAKE = Utilitário de compilação;
# LIBAPACHE2-MOD-PHP = linguagem de script incorporada em HTML do
servidor (módulo Apache 2) (padrão);
# APACHE = Servidor Web;
# LIBGD-DEV = Biblioteca de gráficos GD (versão de desenvolvimento).

# Outras aplicações de desenvolvimento
# XINETD = substituto para o inetd com muitas melhorias.
# BUILD-ESSENTIALS = O pacote build-essential é uma referência para
todos os pacotes necessários para compilar um pacote Debian. Geralmente inclui
os compiladores e bibliotecas GCC / g ++ e alguns outros utilitários;
```

Baixando o Nagios Core

Agora vamos baixar o pacote do Nagios para podermos compilá-lo.

```
# Entra na pasta /tmp:
cd /tmp

# Baixa o pacote do Nagios Core renomeando o arquivo baixado para
nagioscore.tar.gz:
wget -O nagioscore.tar.gz
https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.5.tar.gz

# Descompactar o nagioscore.tar.gz:
tar xzvf nagioscore.tar.gz

# Entrar no nagioscore-nagios-4.4.5:
cd nagioscore-nagios-4.4.5
```

Compilar o nagios core

```
# Vamos iniciar a preparação do ambiente para podermos compilar o pacote,
passamos como argumento o arquivo de configuração do Nagios para o Apache,
onde irá ficar o arquivo de apontamento do Nagios para que possamos acessá-lo
pelo navegador:
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled

# Vamos criar os binários para instalação futura:
sudo make all

# Vamos criar o usuário e grupos do nagios (o grupo deve ser nagcmd).
# O comando 'make install-groups-users' irá criar uma conta do sistema e
adicionar esse usuário no grupo do nagios:
sudo make install-groups-users

# Adicionando o usuário www-data no grupo nagios como um grupo secundário:
sudo usermod -a -G nagios www-data

# Instalando os binários, CGIs e arquivos HTML gerados pelo 'make all':
sudo make install

# Instalando os arquivos de serviço/daemon e habilitando o serviço para
iniciar no boot:
sudo make install-daemoninit

# Instalar o arquivo para termos comando externo no Nagios:
sudo make install-commandmode

# Instalando os arquivos de configuração SAMPLE, isso é necessário, pois, sem
isso, o Nagios não irá iniciar, ele instala tudo que está dentro de
'/usr/local/nagios/etc':
sudo make install-config
```

```
# Instalando os arquivos de configuração do servidor web Apache e definindo as
configurações do Apache:

# Instalando o arquivo de configuração do Apache para a interface da web
do Nagios (/etc/apache2/sites-enabled/nagios.conf):
sudo make install-webconf

# Ativando a reescrita do módulo:
sudo a2enmod rewrite

# Ativando o módulo CGI:
sudo a2enmod cgi

# Liberando o apache no firewall do Ubuntu:
sudo ufw allow Apache

# Recarregando as regras do firewall:
sudo ufw reload
```

[Explicação rewrite.](#)

[Explicação CGI.](#)

Criando uma conta de usuário no Nagios

Agora vamos criar um usuário que poderá acessar a aplicação web do Nagios.

```
# Criando um usuário chamado 'nagiosadmin'
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

# Mudando Dono e Grupo dono do arquivo 'htpasswd.users':
sudo chown nagios. /usr/local/nagios/etc/htpasswd.users
```

Manipuladores de eventos do Nagios

Vamos copiar a pasta de notificação de eventos do Nagios para pasta de produção.

```
# Copiando a pasta 'eventhandlers' para '/usr/local/nagios/libexec/':
sudo cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/

# Mudando usuário e dono da pasta 'eventhandlers':
sudo chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
```

Reiniciando o Nagios/Apache

Vamos reiniciar o apache e o Nagios para "aplicar" as mudanças.

```
# Reiniciando o Apache2:
sudo systemctl restart apache2.service

# Reiniciando o Nagios:
sudo systemctl start nagios.service
```

Agora precisamos acessar o IP do servidor do Nagios para verificar se conseguimos acessar a aplicação web do Nagios, o usuário é **nagiosadmin**, a senha é a senha que você definiu.

Se você esqueceu a senha, pode adicionar o usuário novamente usando o comando abaixo e digitar uma nova senha.

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Caso tenha conseguido acessar, você verá que todos os checks para o localhost (servidor Nagios) estarão vermelhos, como na imagem abaixo:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	CRITICAL	11-07-2019 10:34:09	0d 0h 9m 30s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
	Current Users	CRITICAL	11-07-2019 10:34:47	0d 0h 8m 52s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_users, ...) failed. errno is 2: No such file or directory
	HTTP	CRITICAL	11-07-2019 10:35:24	0d 0h 8m 15s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_http, ...) failed. errno is 2: No such file or directory
	PING	CRITICAL	11-07-2019 10:36:02	0d 0h 7m 37s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping, ...) failed. errno is 2: No such file or directory
	Root Partition	CRITICAL	11-07-2019 10:36:39	0d 0h 7m 0s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_disk, ...) failed. errno is 2: No such file or directory
	SSH	CRITICAL	11-07-2019 10:37:17	0d 0h 6m 22s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ssh, ...) failed. errno is 2: No such file or directory
	Swap Usage	CRITICAL	11-07-2019 10:37:54	0d 0h 5m 45s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_swap, ...) failed. errno is 2: No such file or directory
	Total Processes	CRITICAL	11-07-2019 10:33:32	0d 0h 5m 7s	1/4	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_procs, ...) failed. errno is 2: No such file or directory

Isso se deve ao motivo de não termos os plugins necessários para a verificação desses serviços, podemos ver isso clicando em alguns dos serviços que estão sendo verificados (vamos pegar como exemplo o Current Load), podemos verificar no campo **Status Information** que o erro é devido ao arquivo (plugin) não ter sido encontrado.

Current Status:	CRITICAL (for 0d 0h 13m 25s)
Status Information:	(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory

Para corrigir isso, vamos instalar os plugins padrões do Nagios.

Instalando os plugins padrões do Nagios

Vamos instalar os plugins padrões do Nagios para que possamos ter uma monitoração default na ferramenta.

Pré requisitos

Boa parte dos pré requisitos do Nagios-Plugins já foi instalado anteriormente, vamos instalar apenas os que não foram instalados.

```
sudo apt install -y libmcrypt-dev libssl-dev dc gettext libmcrypt4

# libmcrypt-dev = Arquivos de desenvolvimento da biblioteca de
# decriptografia/criptografia;
# libssl-dev = Kit de ferramentas Secure Sockets Layer - arquivos de
# desenvolvimento;
# dc = calculadora de precisão arbitrária polonesa-reversa do GNU;
# gettext = Utilitários de internacionalização GNU;
# libmcrypt4 = Biblioteca de Des/Encriptação
```

Compilando os plugins

Agora vamos baixar e compilar o pacote de plugins do Nagios.

```
# Entrando no /tmp:
cd /tmp

# Baixando o pacote do Nagios-Plugins:
wget --no-check-certificate -O nagios-plugins.tar.gz
https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz

# Descompactando o pacote baixado:
tar xzf nagios-plugins.tar.gz

# Entrando na pasta descompactada do Nagios-Plugins:
cd /tmp/nagios-plugins-release-2.2.1/

# Instala alguns scripts para auxiliar na compilação:
sudo ./tools/setup

# Preparando o ambiente para a compilação:
sudo ./configure

# Vamos criar os binários para instalação futura:
sudo make

# Instalando os binários, CGIs e arquivos HTML gerados pelo 'make all':
sudo make install
```

Reiniciando o daemon do Nagios

```
# Reinicia o daemon do Nagios:
sudo systemctl restart nagios.service

# Mostra o status do serviço do Nagios:
sudo systemctl status nagios.service
```

Você pode verificar o arquivo de configuração do Nagios usando o comando

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

Re-schedule the next check of this service

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	11-07-2019 14:09:09	0d 0h 1m 44s	1/4	OK - load average: 0.09, 0.12, 0.05
	Current Users	OK	11-07-2019 14:09:47	0d 0h 1m 6s	1/4	USERS OK - 2 users currently logged in
	HTTP	OK	11-07-2019 14:10:18	0d 0h 0m 41s	1/4	HTTP OK - 11912 OK - 11912 bytes in 0.001 second response time
	INX	OK	11-07-2019 14:10:38	0d 0h 0m 19s	1/4	PING OK - Packet loss = 0% , RTT = 0.34 ms
	Root Partition	OK	11-07-2019 14:10:46	0d 0h 0m 10s	1/4	DISK OK - free space: / 16707 MB (76.69% inode=91%)
	SSH	OK	11-07-2019 14:10:28	0d 0h 0m 27s	1/4	SSH OK - OpenSSH_7.6p1 Ubuntu-4ubuntu0.3 (protocol 2.0)
	Swap Usage	OK	11-07-2019 14:07:54	0d 0h 2m 59s	1/4	SWAP OK - 100% free (1969 MB out of 1969 MB)
	Total Processes	OK	11-07-2019 14:08:32	0d 0h 2m 21s	1/4	PROCS OK 34 processes with STATE = RSZSD

Re-schedule the next check of this service

```
Service
Current Users
On Host
localhost
(localhost)
```

Member of
No servicegroups.

127.0.0.1

Service State Information	
Current Status:	OK (for 0d 0h 1m 55s)
Status Information:	USERS OK - 2 users currently logged in
Performance Data:	users=2:20:50:0
Current Attempt:	1/4 (HARD state)
Last Check Time:	11-07-2019 14:09:47
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 0.001 seconds
Next Scheduled Check:	11-07-2019 14:14:47
Last State Change:	11-07-2019 14:09:47
Last Notification:	N/A (notification 0)
Is This Service Flapping?	NO (6.25% state change)
In Scheduled Downtime?	NO
Last Update:	11-07-2019 14:11:38 (0d 0h 0m 4s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Service Commands

- ✗ Disable active checks of this service
- 🕒 Re-schedule the next check of this service
- ❓ Submit passive check result for this service
- ✗ Stop accepting passive checks for this service
- ✗ Stop observing over this service
- ✗ Disable notifications for this service
- 📧 Send custom service notification
- 🕒 Schedule downtime for this service
- ✗ Disable event handler for this service
- ✗ Disable flap detection for this service
- ✗ Clear flapping state for this service

Service Comments

 [Add a new comment](#) [Delete all comments](#)

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This service has no comments associated with it							

