

Nagios の Plugin を Perl で + NRPE

長野 雅広

blog.nomadscafe.jp

id:kazeburo

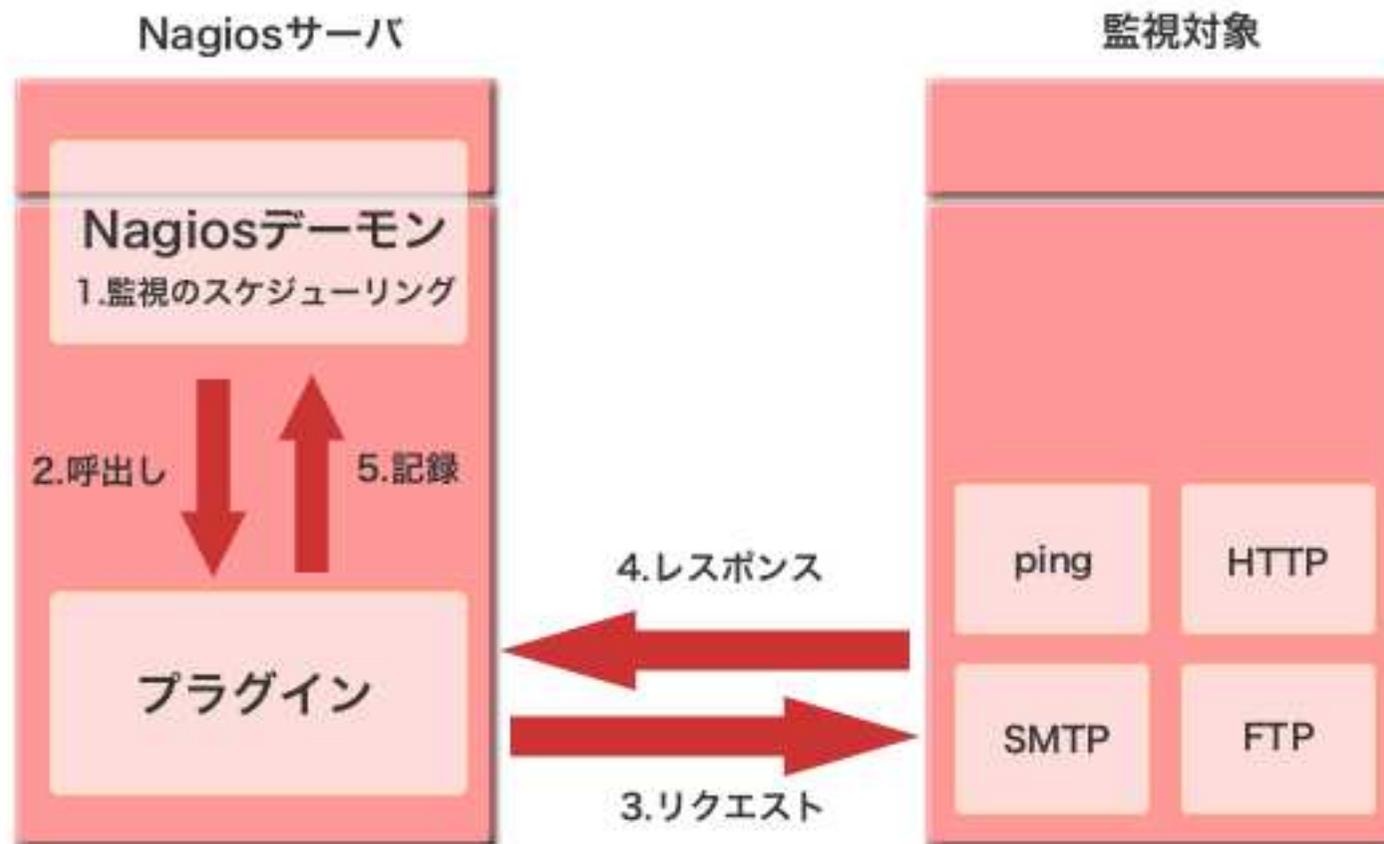



Nagios

Nagiosとは

- オープンソースで開発されている、システムとネットワークの監視ソフトウェア
- 読み方は「ナギオス」
- プラグインやWebインターフェイスの一部がPerl
- M社やH社をはじめ実績も多数

Nagiosの構成



プラグイン

- プラグインの実体はOSのコマンド
 - check_http, check_ftp, check_smtp, check_tcpなど
 - check_httpを実行すると

```
# /usr/lib/nagios/plugins/check_http
check_http: Could not parse arguments
Usage: check_http -H <vhost> | -I <IP-address> [-u <uri>] [-p <port>]
      [-w <warn time>] [-c <critical time>] [-t <timeout>] [-L]
      [-a auth] [-f <ok | warn | critical | follow>] [-e <expect>]
      [-s string] [-I] [-r <regex> | -R <case-insensitive regex>] [-P string]
      [-m <min_pg_size>:<max_pg_size>] [-4|-6] [-N] [-M <age>] [-A string] [-k string]
# /usr/lib/nagios/plugins/check_http -H blog.nomadscafe.jp -u http://blog.nomadscafe.jp/
HTTP OK HTTP/1.1 200 OK - 69191 bytes in 0.019 seconds |time=0.018717s;;;0.000000 size=69191B;;;0
```

プラグインの作成

- プラグインを作成するのは簡単
- サンプル

```
#!/usr/bin/perl
use strict;
use Getopt::Long;
use IO::Socket::INET;

GetOptions(
    'H=s'    => ¥my $host, 'p|port=s' => ¥my $port
);
$port ||= 11211;
exit(3) unless $host;

my $sock = IO::Socket::INET->new( 'PeerAddr' => "$host:$port", Proto => 'tcp' );
If ( $sock ) {
    print qq(OK – connected to $host:$port¥n);
    exit(0);
} else {
    print qq(CRITICAL – cannot connect to $host:port¥n);
    exit(2);
}
```

プラグイン作成のポイント

- exitのステータスコードがサービスの状態を表す
 - OK => 0
 - WARNING => 1
 - CRITICAL => 2
 - UNKNOWN => 3
- 標準出力がプラグインからNagiosへのメッセージ

Nagios::Plugin

- CPANモジュール (<http://search.cpan.org/dist/Nagios-Plugin/>)
 - Nagios::Pluginを利用してサンプルを書き換えると、

```
use Nagios::Plugin;
my $np = Nagios::Plugin->new;
$np->add_arg( spec => 'host=s', required => 1 );
$np->add_arg( spec => 'port=s', default => 11211 );
$np->getopts;
my $sock = IO::Socket::INET->new( 'PeerAddr' => $np->opt->host.".".$np->opt->port, Proto => 'tcp' );
If ( $sock ) {
    $np->nagios_exit(OK, "connected%n");
}
else {
    $np->nagios_exit(CRITICAL, "cannot connect%n");
}
```

- Getoptのwrapper、Performance Dataの書き出しなどをサポートしている
- シンプルな使い方をするのであれば必要ないかも

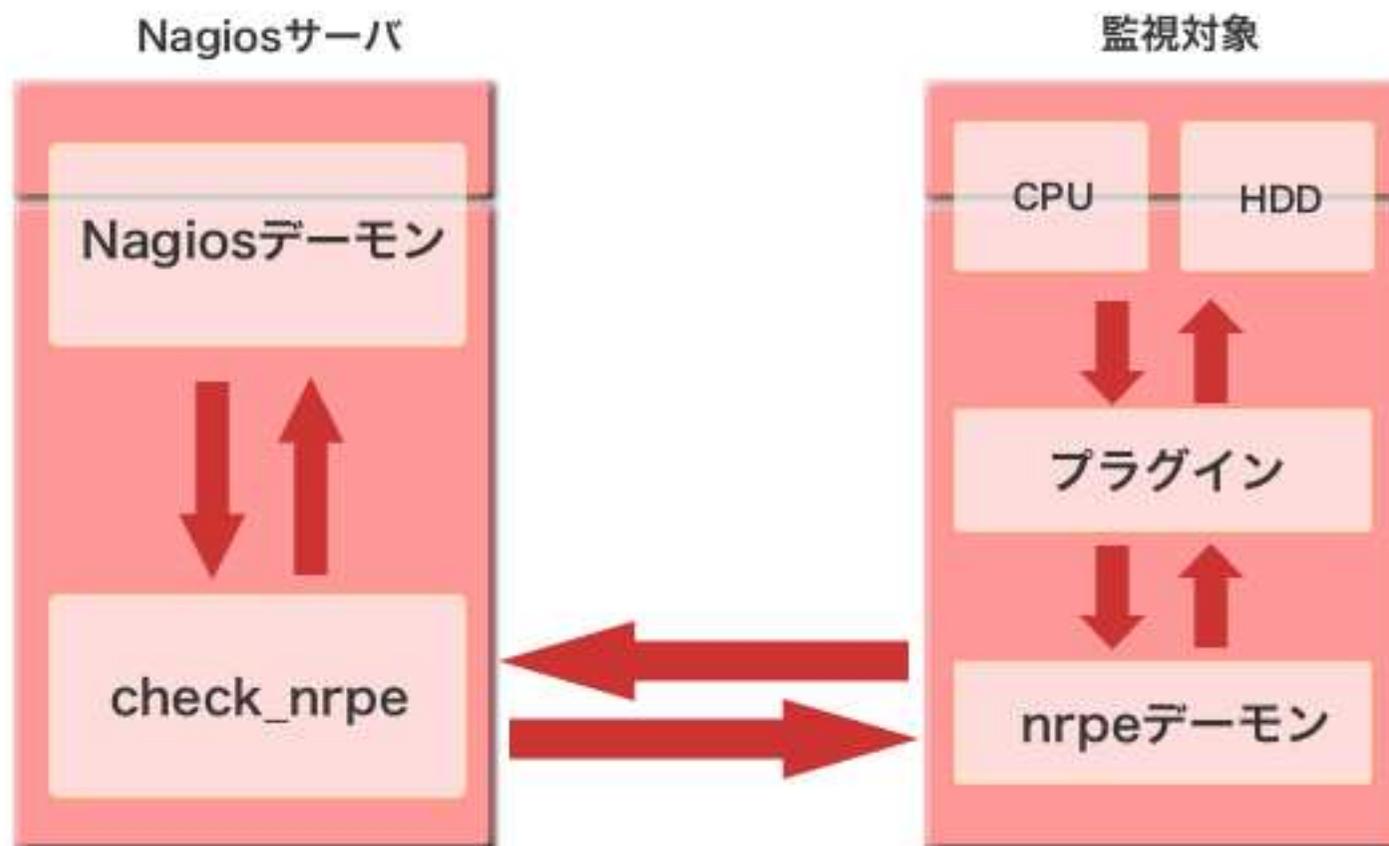



NRPE

NRPEとは

- Nagios Remote Plugin Executer
- Nagiosのアドオン
- check_nrpeプラグインとnrpeデーモンを利用して、リモートのcheck_commandプラグインを実行できる

NRPEの構成



NRPEでできること

- HDDやCPU使用率、プロセス数などリソース監視
- アプリケーション依存の監視
 - DBやログ、キャッシュなどの確認
 - nagiosがアプリケーションの詳細を知る必要がない
 - ⇒ アプリケーションと監視の分離が可能
- それcheck_snmpでもできるよ

まとめ

- プラグインを作成するのは簡単
- nrpeを用いてリモートのリソース監視も可能
- Nagiosは設定が複雑ですが、単に監視だけではなくプラグインしだいで、アプリケーションと密接な使い方ができます。

ご清聴ありがとうございました