

# インスタンスシャットダウン問題対応 参考資料

## 目次

1 はじめに.....	3
2 前提条件.....	3
3 WEB サーバの台数を調整する .....	3
4 NRPE SERVER を設定する.....	3
4.1. NRPE SERVER の設定ファイルを構成管理の対象にする .....	3
4.2. 監視サーバのホストの設定.....	3
4.3. NRPE SERVER の設定ファイルの確認をする .....	4
4.4. NAGIOS の設定ファイルの確認をする.....	4
4.5. CHECK_NREP の実行結果をログに出力する設定をする.....	6
4.6. NRPE SERVER の設定ファイルを更新する.....	7
5 WEB サーバの停止と自動復旧 .....	9
5.1. EXEC_WEBNODE の作成.....	9
5.2. WEB サーバを停止する.....	10
5.3. NAGIOS の管理画面から復旧ログを確認する .....	11
5.3.1. Web サーバの <i>Status</i> が <i>DOWN</i> していることを確認する.....	11
5.3.2. イベントログで監視結果を確認する.....	12
5.4. 監視サーバでログを確認する.....	14
5.4.1. <i>check_webnode</i> コマンドが実行されたことを確認する。 .....	14
5.5. デプロイサーバでログを確認する.....	14

## 図の索引

図 1 監視対象一覧 .....	10
図 2 WEB SERVER002の確認.....	11
図 3 WEB サーバの停止の確認 .....	12
図 4 WEB サーバの停止イベントの確認.....	13
図 5 EVENT HANDLER の確認 .....	13

# 1はじめに

この手順書は不意にインスタンスがシャットダウンした時、Nagios の監視によって、自動的に Web サーバが起動するための手順書です。

## 2前提条件

- 「PuppetMCollective による自動化」が完了していること
- 「NagiosGanglia を用いた監視の設定」が完了していること
- 「掲示板アプリケーションのスケールアウト」が完了していること

## 3Web サーバの台数を調整する

Web サーバの台数を2台にする。

add\_webnode.sh または、delete\_webnode.sh を利用して、**Web** サーバの台数を2台にします。

## 4NRPEServer を設定する

### 4.1.NRPE Server の設定ファイルを構成管理の対象にする

Nagios の Plugin をデプロイサーバ上で実行するために、**NREP Server** の設定をします。

以下のコマンドはデプロイサーバで実行します。

NRPE Server の設定を有効にする。site.pp の199行目付近の **include nrpe\_server** のコメントをはずす。

```
deploy# vi /etc/puppet/manifests/site.pp
```

修正前

```
include gmond  
#include nrpe_server
```

修正後

```
include gmond  
include nrpe_server
```

### 4.2.監視サーバのホストの設定

デプロイサーバが NRPEServer からの通信を許可するために、hosts の設定をします。

監視サーバの IP アドレスを取得する。

```
deploy# mco facts ipaddress -F fqdn=/^monitor/
```

実行結果

```
Report for fact: ipaddress
```

```
10.3.7.146
```

```
found 1 times
```

```
Finished processing 1 / 1 hosts in 25.26 ms
```

監視サーバの IP アドレスを hosts に設定する。

```
deploy# vi /etc/hosts
```

実行結果

```
127.0.0.1 localhost.localdomain localhost
```

```
127.0.0.1 deploy.nii.localdomain
```

```
{監視サーバの IP アドレス} monitor.nii.localdomain
```

## 4.3.NRPEServer の設定ファイルの確認をする

Nagios と通信するために NRPEServer の Config が正しく設定しているか確認をします。  
この設定をすることで、Nagios が障害を検知したとき、デプロイサーバ上の NRPEServer が Nagios の Plugin を実行することができます。

以下はデプロイサーバで確認します。

設定ファイルを確認する。

```
deploy# vi /etc/puppet/templates/etc/nagios/nrpe.cfg
```

nrpe.cfg

```
#78行目付近 許可しているホストに monitor.nii.localdomain を指定する。
```

```
allowed_hosts=monitor.nii.localdomain
```

```
#93行目付近 check_nrpe コマンドから引数を受け取る許可をする。
```

```
dont_blame_nrpe=1
```

```
#202行目付近 check_webnode プラグインを有効にする。
```

```
command[check_webnode]=/usr/lib/nagios/plugins/nii-
```

```
deploy/plugins/nagios/check_webnode $ARG1$ $ARG2$ $ARG3$
```

## 4.4.Nagios の設定ファイルの確認をする

Nagios から Web サーバへ Ping を送り、応答がなかった場合に、イベントが発生します。  
イベントが発生するための、event\_handler の設定の確認をおこないます。

以下はデプロイサーバで確認します。

ngaios.cfg

```
deploy# vi /etc/puppet/templates/etc/nagios/nagios.cfg
```

#### 実行結果

#52行目付近 enable\_event\_handler が1がセットしていることを確認する。  
enable\_event\_handlers=1

#### web.cfg

```
deploy# vi /etc/puppet/templates/etc/nagios/servers/web.cfg.erb
```

#### 実行結果

#47行目付近 check\_webnode コマンドが定義されていることを確認する。

```
define command{
    command_name check_webnode
    command_line $USER1$/check_nrpe -H deploy.nii.localdomain -t 60 -c
check_webnode -a $SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$ >>
/var/log/nii-monitor.log
}
```

#84行目付近 Web サーバの監視項目に PING が追加されていることを確認します。

#86行目付近 event\_handler に check\_webnode が指定されていることを確認します。

```
define service{
    use                                web-service                ; Name of service
template to use
    hostgroup_name      webservers
    service_description  PING
    check_command        check_ping!100. 0, 20%!500. 0, 60%
    event_handler        check_webnode
    max_check_attempts   1
}
```

## 4.5.check\_nrep の実行結果をログに出力する設定をする

web.cfg で定義した check\_webnode コマンドの実行結果をログに出力設定をします。  
以下のコマンドは監視サーバで実行します。

nii-monitor ログを作成する。

```
monitor# touch /var/log/nii-monitor.log
```

実行権限を付与する。

```
monitor# chmod 666 /var/log/nii-monitor.log
```

## 4.6.NRPEServer の設定ファイルを更新する

確認した nrpc.cfg ファイルを更新します。  
以下のコマンドはデプロイサーバで実行します。

MCollectivClient を使ってデプロイサーバの設定ファイルを更新する。

```
deploy# mco puppetd runonce -I deploy.nii.localdomain -v
```

実行結果

```
* [ =====> ] 1 / 1

deploy.nii.localdomain      : OK
{:output=>      "Signalled daemonized puppet agent to run (process 999);
Currently idling; last completed run 1246 seconds
ago",      :status=>"idling",      :enabled=>1,      :running=>0,      :idling=>1,
      :stopped=>0,      :lastrun=>1332688527}

---- puppetd#runonce call stats ----
      Nodes: 1 / 1
      Pass / Fail: 1 / 0
      Start Time: Sun Mar 25 15:36:13 +0000 2012
      Discovery Time: 0.00ms
      Agent Time: 54.55ms
      Total Time: 54.55ms
```

MCollectiveClient を使用して nrpe サーバを再起動する

```
deploy# mco service nagios-nrpe-server restart -I deploy.nii.localdomain -v
```

実行結果

```
* [ =====> ] 1 / 1

deploy.nii.localdomain      status=stopped
      OK

---- service summary ----
      Nodes: 1 / 1
      Statuses: stopped=1
      Elapsed Time: 0.78 s
```

MCollectiveClientを使って監視サーバの設定ファイルを更新する

```
deploy# mco puppetd runonce -I monitor.nii.localdomain -v
```

実行結果

```
* [ =====> ] 1 / 1

monitor.nii.localdomain          : OK

{:stopped=>0,      :status=>"idling",      :idling=>1,      :running=>0,      :output=>
  "Signalled daemonized puppet agent to run (process 1161); Currently
  idling;          last      completed      run      944      seconds
  ago",          :lastrun=>1337761999,      :enabled=>1}

---- rpc stats ----
      Nodes: 1 / 1
    Pass / Fail: 1 / 0
    Start Time: Wed May 23 08:49:03 +0000 2012
  Discovery Time: 0.00ms
    Agent Time: 52.03ms
    Total Time: 52.03ms
```

MCollectiveClientを使用して nagios を再起動する

```
deploy# mco service nagios3 restart -I monitor.nii.localdomain -v
```

実行結果

```
* [ =====> ] 1 / 1

monitor.nii.localdomain          : OK

{:stopped=>0,      :status=>"idling",      :idling=>1,      :running=>0,      :output=>
  "Signalled daemonized puppet agent to run (process 1161); Currently
  idling;          last      completed      run      944      seconds
  ago",          :lastrun=>1337761999,      :enabled=>1}

---- rpc stats ----
      Nodes: 1 / 1
    Pass / Fail: 1 / 0
    Start Time: Wed May 23 08:49:03 +0000 2012
  Discovery Time: 0.00ms
    Agent Time: 52.03ms
    Total Time: 52.03ms
```

以上で NRPEServer の設定は完了です。Nagios がシャットダウンを検知したら、デプロイサーバ上でスクリプトが実行されます。



## 5 Web サーバの停止と自動復旧

Web サーバを停止後、Nagios が Web サーバを自動起動し復旧するまでの手順です。

1. Nagios の Hosts に表示されている **webserver002**のサーバを停止します。
2. Nagios が PING 監視しており、Web サーバが停止したため、ステータスが OK から CRITICAL に変わります。
3. ステータスが CRITICAL に変更後、Service で定義されている event\_handler が呼ばれ、デプロイサーバが NRPE プラグインを実行されます。
4. NRPE Server が check\_webnode プラグインを実行します。
5. check\_webnode プラグインは引数にとった、SERVICESTATE、SERVICESTATETYPE、SERVICEATTEMPT を評価します。
6. SERVICESTATE が CRITICAL、SERVICESTATETYPE が HARD、SERVICEATTEMPT が1の場合に、mcollective-client によって exec\_webnode コマンドが実行され、Web サーバを1台起動されます。

### 5.1.exec\_webnode の作成

自動復旧の最後で、実行される、exec\_webnode のプログラムを修正してください。  
このプログラムはサーバを1台実行するためのプログラムです。

exec\_webnode を修正する。

以下はサンプルコードです。

```
deploy# vi /root/work/deploy/plugins/nagios/exec_webnode
```

```
exec_webnode
```

```
#!/bin/bash
# LB から停止した Web サーバをとりのぞく
# 現在起動中の Web サーバの IP アドレスを取得しリストを作成する
mco facts ipaddress -F fqdn=/^web/ -j | /root/work/deploy/bin/retrieve ip mco --
format file > /var/tmp/nginx/nginx.ipset

# 設定ファイルを更新する
mco puppetd runonce -F fqdn=/^lb/ -v

# nginx をリスタートする
mco service nginx restart -F fqdn=/^lb/ -v

# Web サーバを1台起動する。
cd /root/work/deploy/task/
./add_webnode.sh >> /var/log/nii-deploy.log &
exit 0
```

add\_webnode の出力結果用ファイルを作る。

```
deploy# touch /var/log/nii-deploy.log
```

実行権限を付与する。

```
deploy# chmod 666 /var/log/nii-deploy.log
```

## 5.2.Web サーバを停止する

ブラウザから Nagios を表示する。

```
http://[監視サーバのパブリック IP アドレス]/nagios3
```

Nagios の Current Status > Hosts をクリックし現在の監視対象一覧を表示する。

The screenshot shows the Nagios 3 web interface. The sidebar on the left contains navigation links for General, Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, and System. The main content area displays the 'Host Status Details For All Host Groups' page. At the top, there are two summary boxes: 'Current Network Status' and 'Host Status Totals'. The 'Host Status Totals' box shows 7 Up, 0 Down, 0 Unreachable, and 0 Pending hosts. Below this is a table of host details with 7 entries, all showing 'UP' status. The table columns are Host, Status, Last Check, Duration, and Status Information.

Host	Status	Last Check	Duration	Status Information
database	UP	2012-05-22 09:49:26	4d 23h 4m 17s	PING OK - Packet loss = 0%, RTA = 2.23 ms
deployserver	UP	2012-05-22 09:48:31	4d 23h 4m 13s	PING OK - Packet loss = 0%, RTA = 0.23 ms
loadbalancer	UP	2012-05-22 09:48:46	4d 23h 4m 10s	PING OK - Packet loss = 0%, RTA = 6.46 ms
localhost	UP	2012-05-22 09:46:31	4d 23h 3m 3s	PING OK - Packet loss = 0%, RTA = 0.05 ms
monitor	UP	2012-05-22 09:49:11	4d 23h 4m 3s	PING OK - Packet loss = 0%, RTA = 0.05 ms
webserver001	UP	2012-05-22 09:49:26	4d 23h 3m 59s	PING OK - Packet loss = 0%, RTA = 0.16 ms
webserver002	UP	2012-05-22 09:48:36	0d 0h 8m 17s	PING OK - Packet loss = 0%, RTA = 0.20 ms

図 1 監視対象一覧

今回は webserver002を停止する。

webserver002をクリックし詳細情報を表示し IP アドレスを確認する。

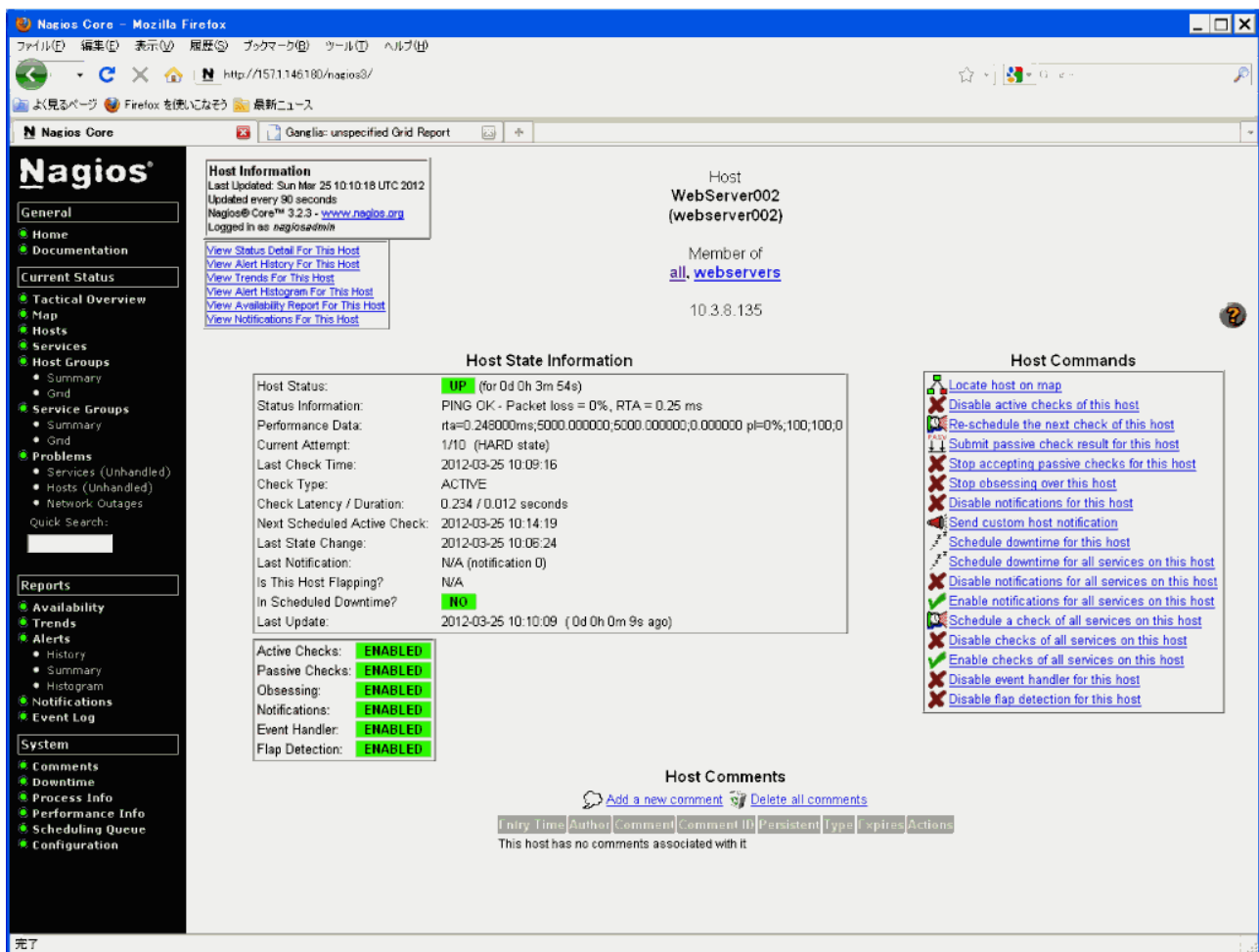


図 2 WebServer002の確認

確認した IP アドレスを元に CloudClient から webserver002 を停止する。

## 5.3.Nagios の管理画面から復旧ログを確認する

### 5.3.1.Web サーバの Status が DOWN していることを確認する

Current Status > Hosts を表示し、Web サーバの Status が DOWN していることを確認する。

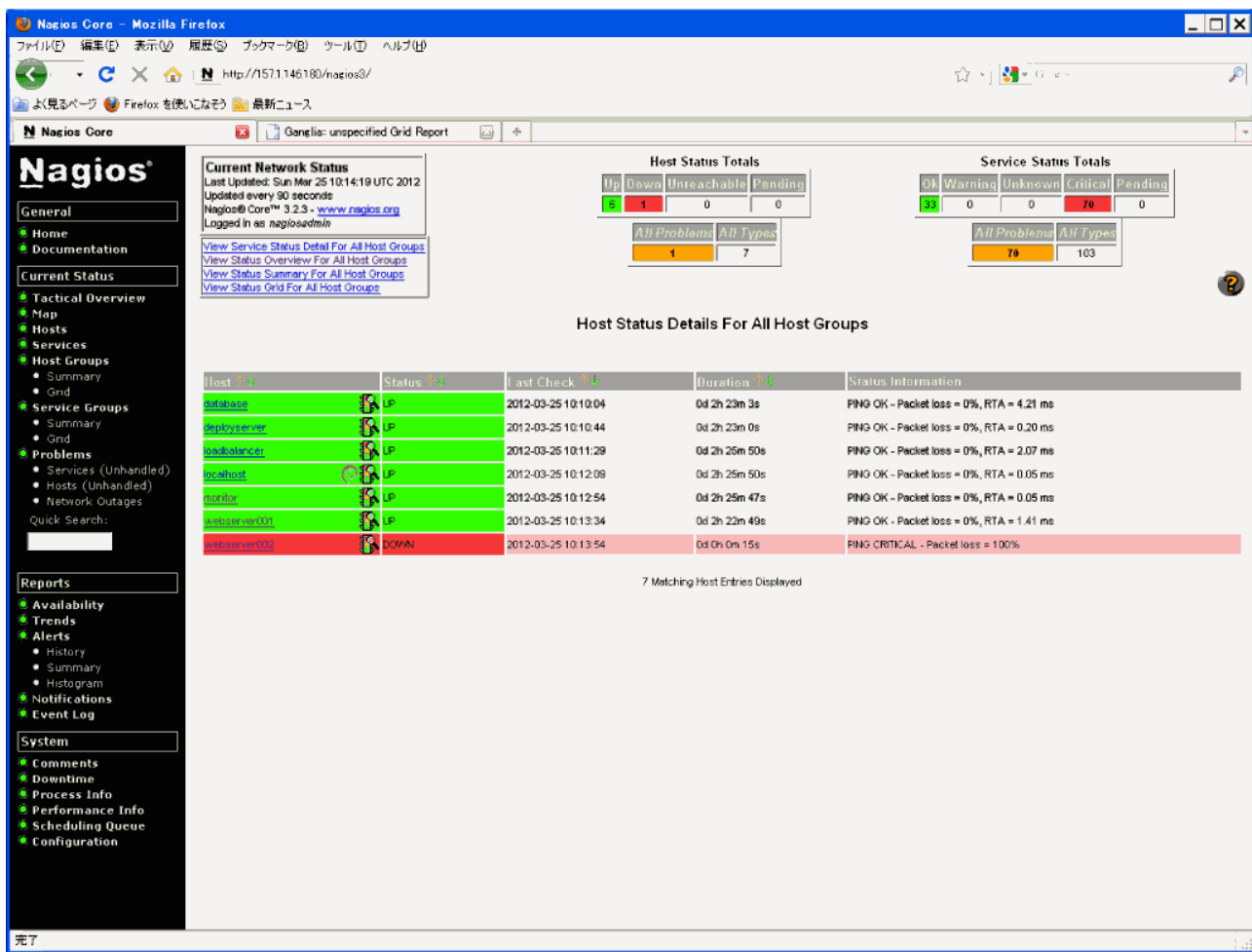


図 3 Web サーバの停止の確認

### 5.3.2. イベントログで監視結果を確認する

Reports > Eveng Log を表示し、イベントログを確認する。

Webserver002が停止したことを確認する。

```
Service Critical[2012-03-25 10:14:44] SERVICE ALERT:
webserver002:PING:CRITICAL:SOFT;2;CRITICAL - Host Unreachable (10.3.8.135)
```

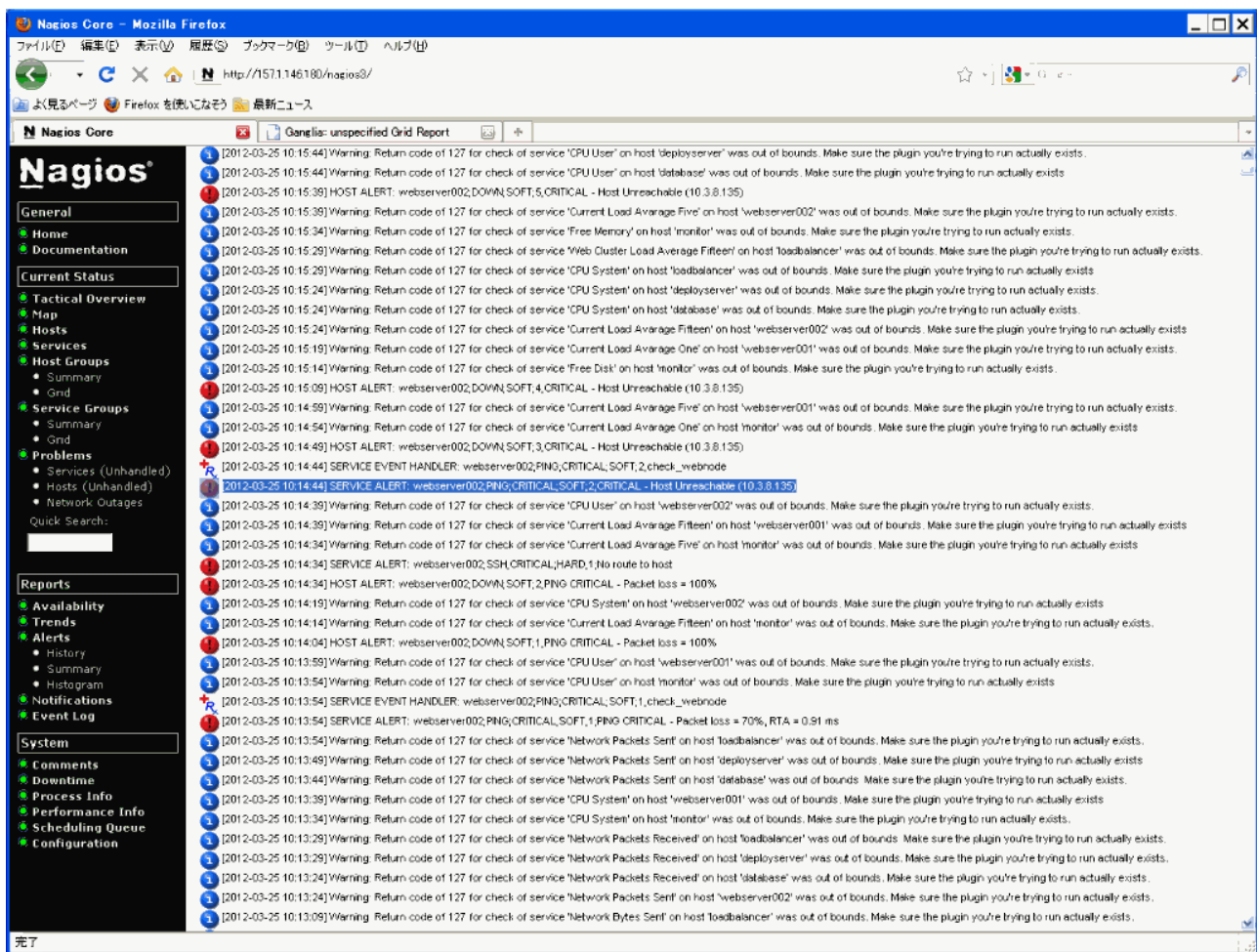


図 4 Web サーバの停止イベントの確認

Nagios が EVENT HANDLER を実行したことを確認する。

SERVICE EVENT HANDLER: webserver001;PING;CRITICAL;HARD;1;check\_webnode



図 5 EVENT HANDLER の確認

## 5.4.監視サーバでログを確認する

監視サーバ上の Nagios の check\_nrpe プラグインが動作したことを確認します。

以下は監視サーバ上で実行します。

### 5.4.1.check\_webnode コマンドが実行されたことを確認する。

Tail コマンドを使い check\_nrpe が実行されたことを確認する。

```
monitor# tail /var/log/nii-monitor.log
```

実行結果

```
1 / 1

[deploy.nii.localdomain] exit=0:
[1332693598] Success add webnode
[1332693598] Request Service: CRITICAL HARD
```

イベントハンドラーが実行されたことを確認する。

```
monitor# tail -f /var/log/nagios3/debug.log
```

実行結果

```
Running command '/usr/lib/nagios/plugins/check_nrpe -H deploy.nii.localdomain -c
check_webnode -a CRITICAL HARD 1 >> /var/log/nii-monitor.log'...
```

## 5.5.デプロイサーバでログを確認する

デプロイサーバ上で exec\_webnode が実行されたことを確認します。

Tail コマンドを使い exec\_webnode が実行されたことを確認する。

```
deploy# tail -f /var/log/nii-deploy.log
```

実行結果

```
launch instance: i-3E6A0840
I, [2012-03-25T17:07:10.304963 #6297] INFO -- : New RightAws::Ec2 using single-
threaded mode
I, [2012-03-25T17:07:20.310280 #6297] INFO -- : Opening new HTTPS connection to
vclc0006.ecloud.nii.ac.jp:8773
instance state running i-3E6A0840
running instance
mail server: 10.3.7.143
db server: 10.3.7.142
I, [2012-03-25T17:09:45.461582 #6341] INFO -- : New RightAws::Ec2 using single-
threaded mode
I, [2012-03-25T17:09:45.472120 #6341] INFO -- : Opening new HTTPS connection to
vclc0006.ecloud.nii.ac.jp:8773
```

```
retry connection 10.3.7.131
Starting mcollective: * Starting puppet agent
(中略)
monitor.nii.localdomain          status=running
                                OK

---- service summary ----
      Nodes: 1 / 1
      Statuses: started=1
Elapsed Time: 16.23 s
```