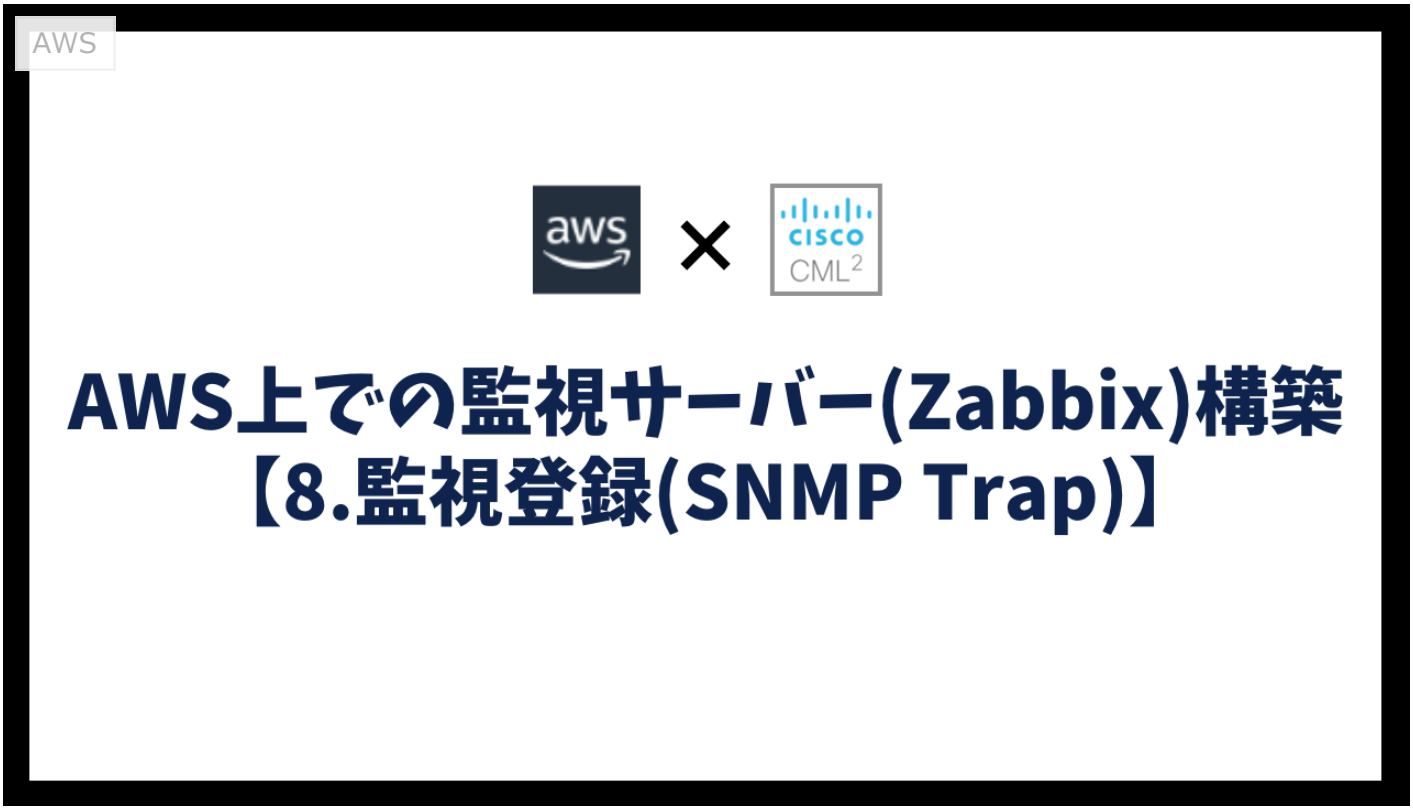
AWS上での監視サーバー(Zabbix)構築【8.監視登録(SNMP Trap)】



2021.10.08　2021.09.16

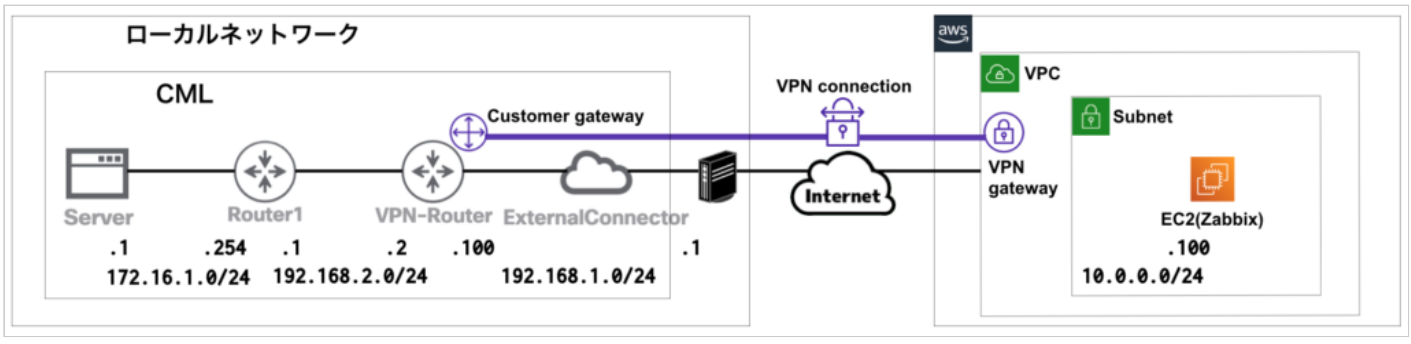
監視サーバーをAWS上で構築し、CML上のネットワーク機器/サーバーを監視します。監視ソフトウェアはZabbixを利用します。

【前回】AWS上での監視サーバー(Zabbix)構築【7.監視登録(SNMP)】
【次回】AWS上での監視サーバー(Zabbix)構築【9.メール通知(SNS)】

ネットワーク構成

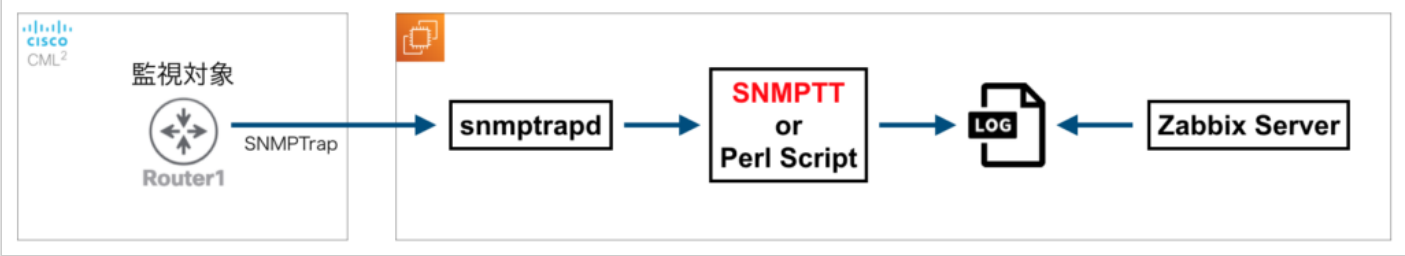
下記のネットワーク環境を構築し、AWS上のEC2(Zabbixサーバー)から、CML上のネットワーク機器/サーバーを監視できるようにしていきます。

【参考】AWSサイト間VPNの構築（1.AWSの基本設定）



SNMP Trap監視の仕組み

ZabbixでのSNMPTrapを利用しての監視は、下記の仕組みで行われます。



1. 監視対象機器がSNMPTrapを送出する。
2. サーバー上のsnmptrapdがSNMPTrapを受け取る。
3. SNMPTT or PerlScriptがSNMPTrapを加工してログファイルへ書き出す。
4. ZabbixServerがログファイルを読み取り、条件に沿って障害として検知する。

ここでは、SNMPTTを利用した監視方法を説明します。

SNMPTTのインストール・設定

EPELレポジトリのインストール

SNMPTTの利用に必要なEPELというリポジトリをインストールします。
AmazonLinuxの場合、yumでのインストールはエラーとなるため、”amazon-linux-extras”を利用します。

```
sudo amazon-linux-extras install -y epel
```

明示的にEPELを指定した時のみ、EPELリポジトリを利用するようにします。

```
sudo cp -p /etc/yum.repos.d/epel.repo /etc/yum.repos.d/epel.repo.backup
sudo vi /etc/yum.repos.d/epel.repo

(下記を編集)
[epel]
enabled=1
↓
enabled=0
```

※参考※

yumを利用してインストールしようとすると下記のエラーが表示され、”amazon-linux-extras”を利用するように促されます。

```
[ec2-user@ip-10-0-0-100 ~]$ sudo yum install -y epel-release
読み込んだプラグイン:extras_suggestions, langpacks, priorities, update-motd
amzn2-core
 | 3.7 kB  00:00:00
パッケージ epel-release は利用できません。
エラー: 何もしません

epel-release is available in Amazon Linux Extra topic "epel"

To use, run
# sudo amazon-linux-extras install epel
```

SNMPTTのインストール・設定

SNMPTTをEPELリポジトリを利用してインストールします。

```
sudo yum --enablerepo=epel install -y snmptt
```

snmp/syslog関連のツールをインストールします。

```
sudo yum --enablerepo=epel install -y net-snmp net-snmp-utils net-snmp-perl perl-Sys-Syslog
```

snmptt.iniを編集します。

```
sudo cp -p /etc/snmp/snmptt.ini /etc/snmp/snmptt.ini.backup
sudo vi /etc/snmp/snmptt.ini

(下記を編集)
[General]
mode = standalone
↓
mode = daemon
```

```
net_snmp_perl_enable = 0
↓
net_snmp_perl_enable = 1

net_snmp_perl_best_guess = 0
↓
net_snmp_perl_best_guess = 2

#date_time_format =
↓
date_time_format = %Y/%m/%d %H:%M:%S

[DaemonMode]
sleep = 5
↓
sleep = 1

[Debugging]
DEBUGGING = 0
↓
DEBUGGING = 1

DEBUGGING_FILE =
# DEBUGGING_FILE = /var/log/snmptt/snmptt.debug
↓
# DEBUGGING_FILE =
DEBUGGING_FILE = /var/log/snmptt/snmptt.debug

DEBUGGING_FILE_HANDLER =
# DEBUGGING_FILE_HANDLER = /var/log/snmptt/snmptthandler.debug
↓
# DEBUGGING_FILE_HANDLER =
DEBUGGING_FILE_HANDLER = /var/log/snmptt/snmptthandler.debug

[TrapFiles]
/etc/snmp/snmptt.conf
↓
/etc/snmp/snmptt_base.conf
```

snmptt_base.confファイルに検知したいSNMPTrapの内容を記述します。今回は、リンクダウン/リンクアップを検知できるようにします。

```
sudo vi /etc/snmp/snmptt_base.conf

（下記を記述）
EVENT general .1.3.6.1.6.3.1.1.5.3 "linkDown" Critical
FORMAT ZBXTRAP $aA $ar $*

EVENT general .1.3.6.1.6.3.1.1.5.4 "linkUp" Normal
FORMAT ZBXTRAP $aA $ar $*
```

上記で定義している変数の意味は下記の通りです。ネットワーク機器からのSNMPTrapの場合、"$aA"と"$ar"は同じIPアドレスが表示されます。

変数	意味
$aA	agent-addrのIPアドレス
$ar	IPアドレス
$*	全てのSNMPTrap内メッセージ

snmptrapdの設定

snmptrapdを編集します。（起動オプションの設定）

```
sudo cp -p /etc/sysconfig/snmptrapd /etc/sysconfig/snmptrapd.backup
sudo vi /etc/sysconfig/snmptrapd

（下記を追記）
OPTIONS="-m +ALL -Lsd -On"
```

snmptrapd.confを編集します。（SNMPTTを実行するための設定）

```
sudo cp -p /etc/snmp/snmptrapd.conf /etc/snmp/snmptrapd.conf.backup
sudo vi /etc/snmp/snmptrapd.conf

（下記を追記）
```

```
authCommunity   log,execute,net public
authCommunity   log,execute,net cmlpublic
perl do "/usr/share/snmptt/snmptthandler-embedded"
```

Zabbix Sever の設定

zabbix_server.confを編集します。（読み取るログファイルの変更、SNMPTrapperの有効化）

```
sudo cp -p /etc/zabbix/zabbix_server.conf /etc/zabbix/zabbix_server.conf.backup_1
sudo vi /etc/zabbix/zabbix_server.conf

（下記を編集）
SNMPTrapperFile=/var/log/snmptrap/snmptrap.log
↓
SNMPTrapperFile=/var/log/snmptt/snmptt.log

# StartSNMPTrapper=0
↓
StartSNMPTrapper=1
```

サービスの起動

SNMPTTとsnmptrapdを起動し、自動起動設定を追加します。

```
sudo systemctl start snmptt snmptrapd
sudo systemctl enable snmptt snmptrapd
```

ZabbixServerを再起動します。

```
sudo systemctl restart zabbix-server
```

SNMPTTの動作確認

サーバーから手動でSNMPTrapを送出し動作確認を行います。

```
sudo snmptrap -v 2c -c public 127.0.0.1 '' .1.3.6.1.6.3.1.1.5.3
```

snmptt.logに表示されることを確認します。

```
sudo cat /var/log/snmptt/snmptt.log
```

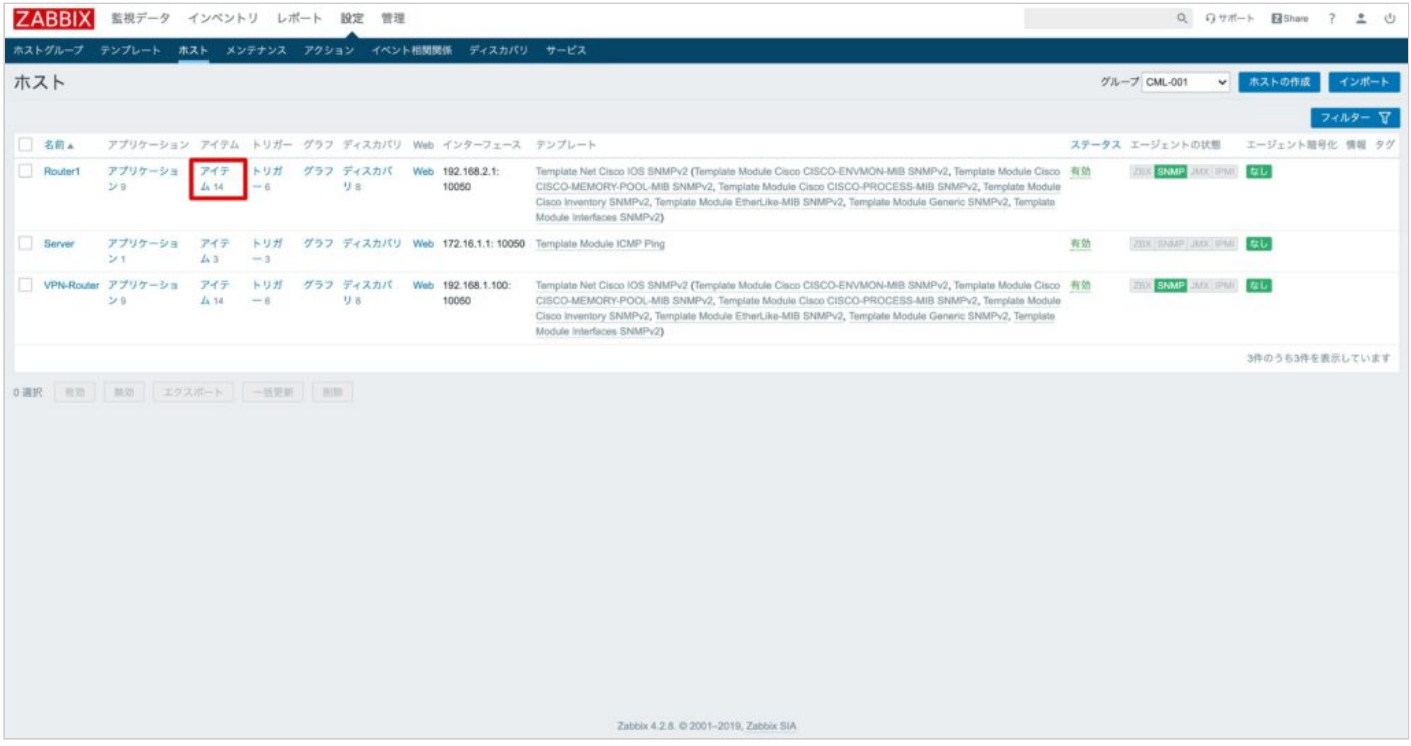
```
[ec2-user@ip-10-0-0-100 ~]$ sudo cat /var/log/snmptt/snmptt.log
2021/09/14 02:07:41 .1.3.6.1.6.3.1.1.5.3 Critical "linkDown" 127.0.0.1 -
ZBXTRAP 127.0.0.1 127.0.0.1
```

Zabbixの設定

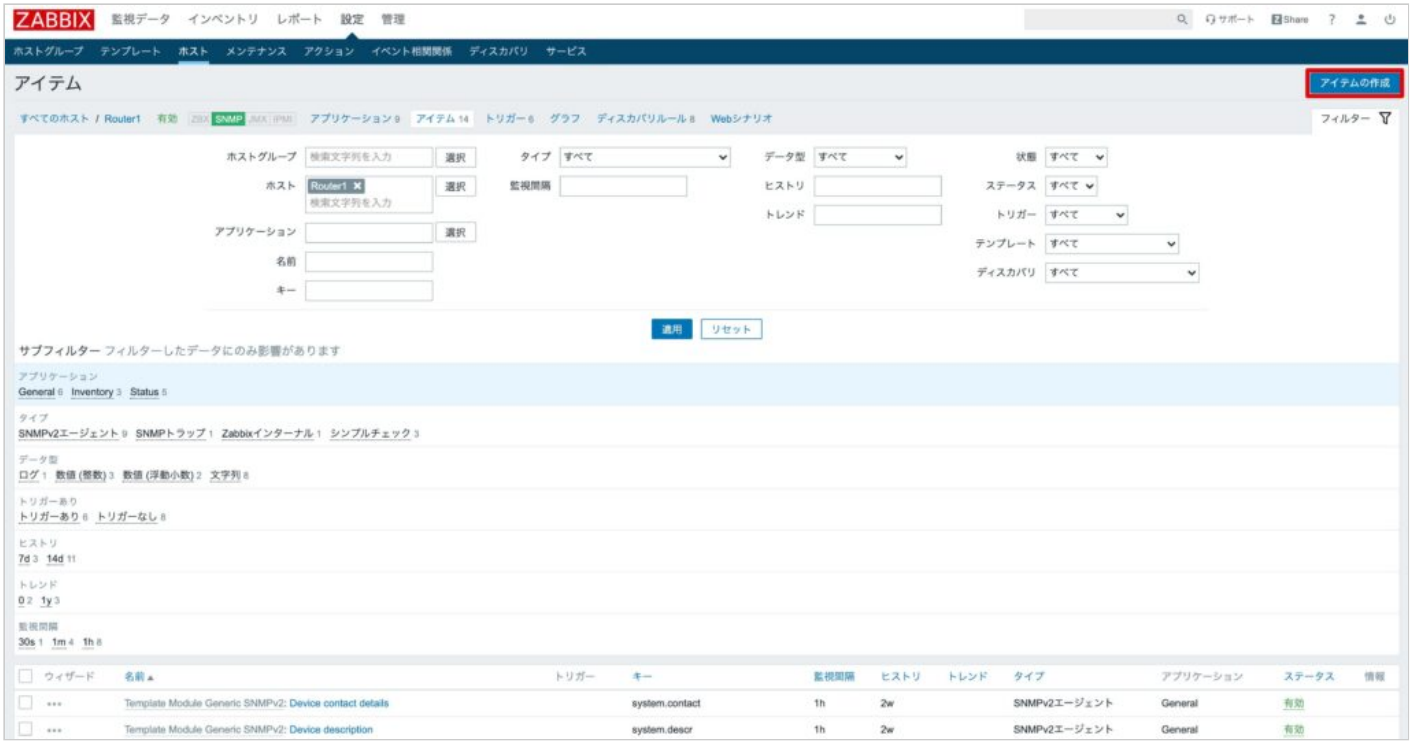
Router1のGigabitEthernet0/3のリンクダウンを検知するための設定を行ないます。（汎用的な設定ではありませんが、動作確認として検知対象を絞って設定します。）

アイテムの登録

Router1の「アイテム」をクリックします。

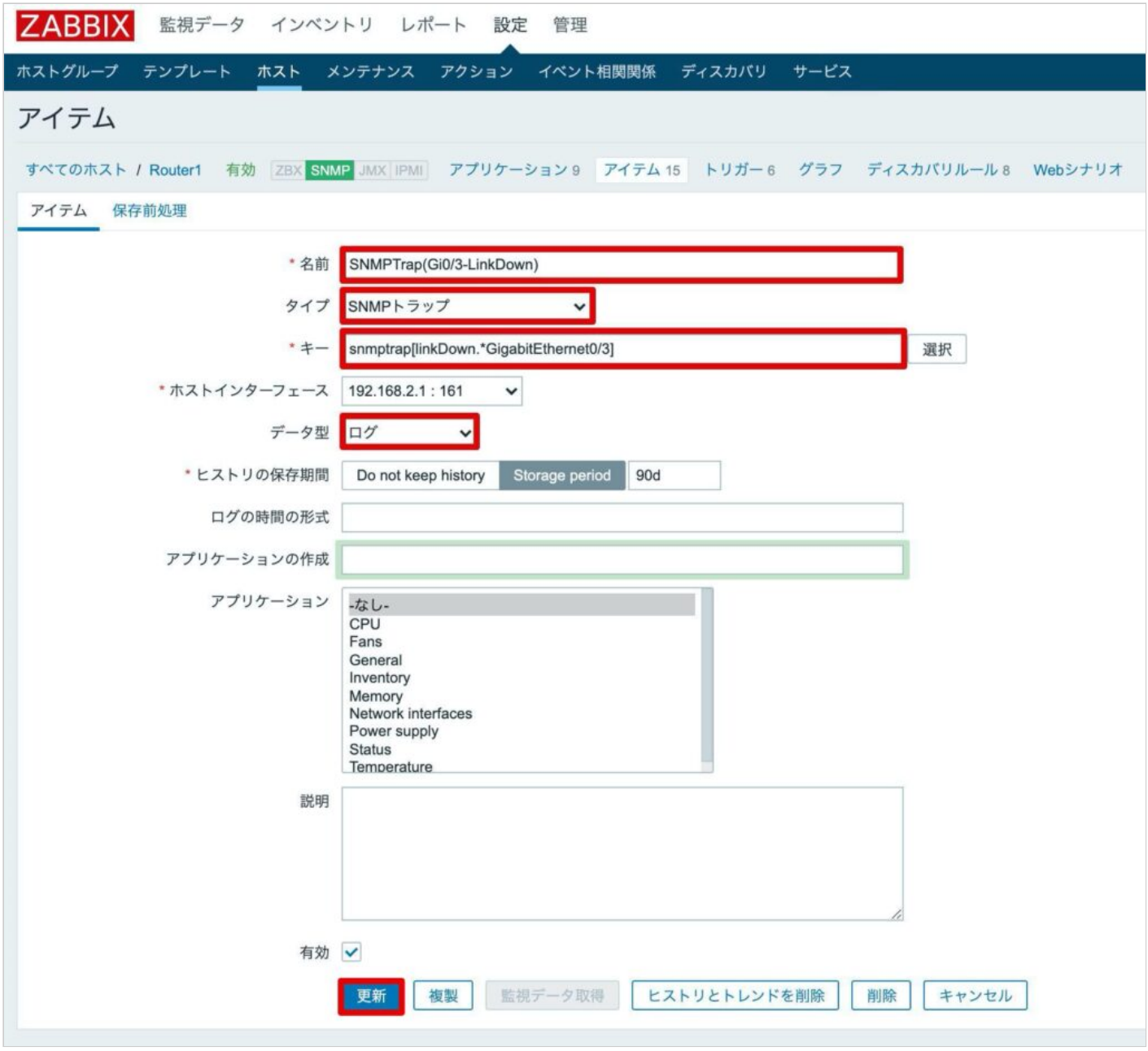


「アイテムの作成」をクリックします。



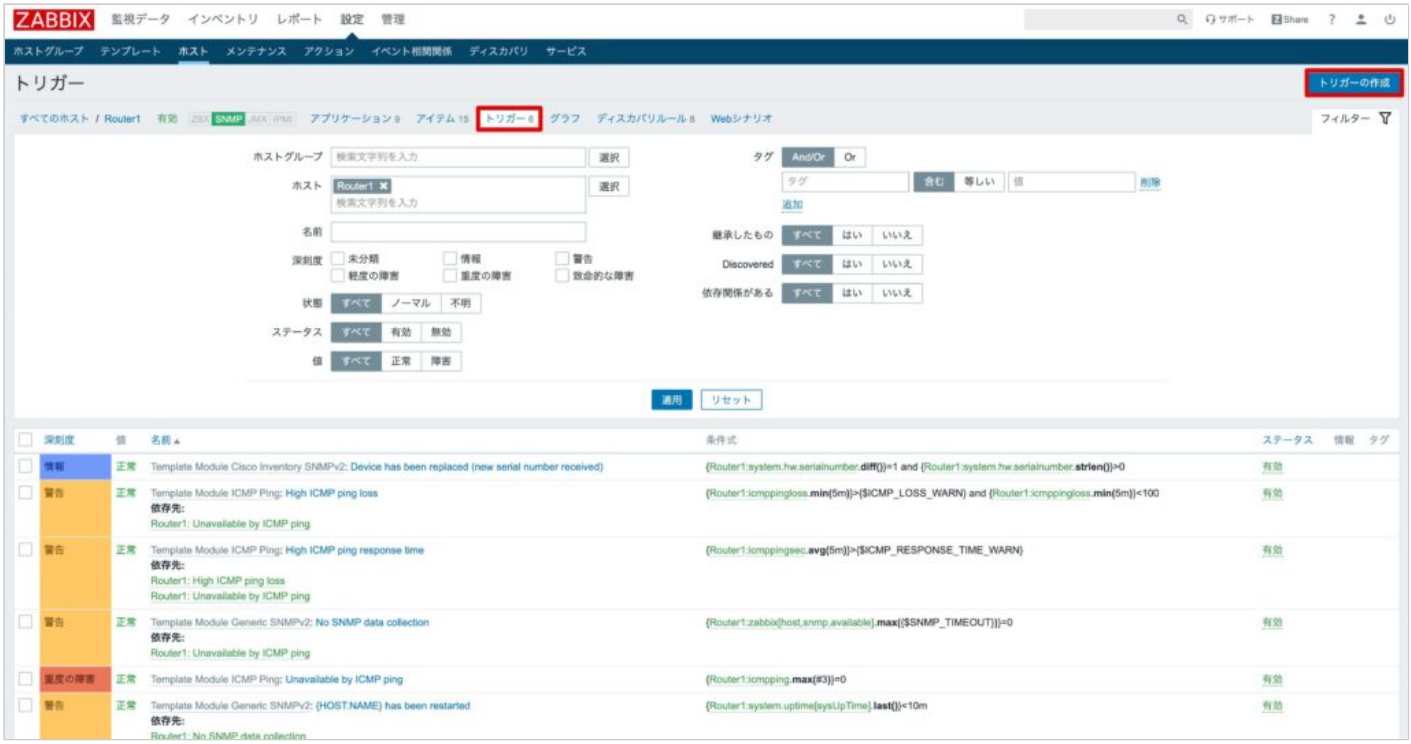
下記の通り設定し、更新をクリックします。

項目	内容
名前	SNMPTrap(Gi0/3-LinkDown) ※任意の名前
タイプ	SNMPトラップ
キー	snmptrap[linkDown.*GigabitEthernet0/3]
データ型	ログ



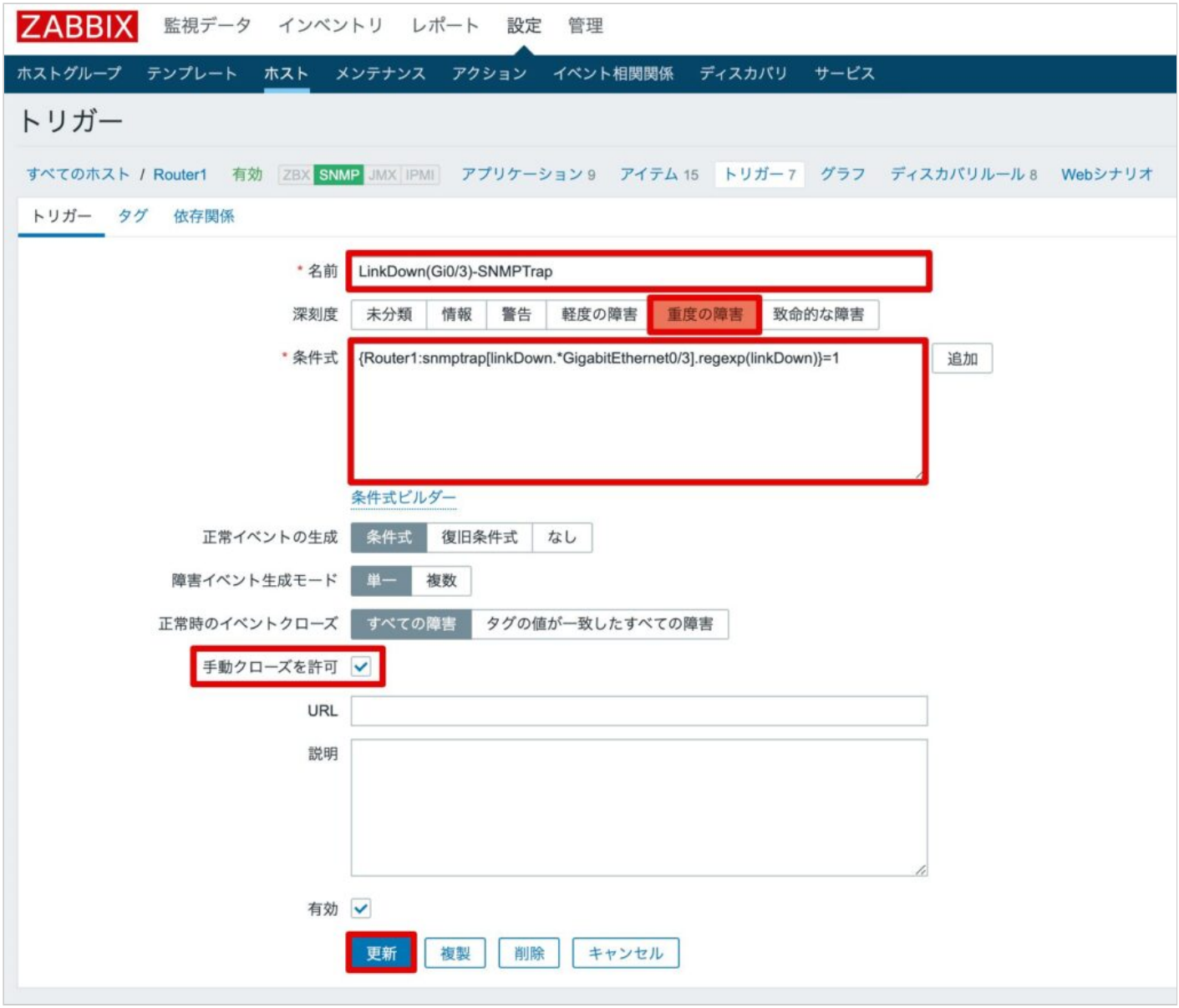
トリガーの登録

「トリガー」→「トリガーの作成」をクリックします。



下記の通り設定し、更新をクリックします。

項目	内容
名前	LinkDown(Gi0/3)-SNMPTrap ※任意の名前
深刻度	重度の障害 ※任意の深刻度
条件式	{Router1:snmptrap[linkDown.*GigabitEthernet0/3].regexp(linkDown)}=1
手動クローズを許可	チェックを入れる



ネットワーク機器の設定

Router1でTrapを送出するための設定を行います。

```
[Router1]
snmp-server community cmlpublic RO
snmp-server host 10.0.0.100 version 2c cmlpublic
snmp-server enable traps
```

監視検知の確認

サーバー側でsnmptt.logを確認できるようにしておきます。

```
sudo tail -f /var/log/snmptt/snmptt.log
```

Router1のGi0/3をshutdownします。

```
[Router1]
int Gi0/3
shut
```

snmptt.logに表示されたことを確認します。

```
02:13:49 .1.3.6.1.6.3.1.1.5.3 Critical "linkDown" 192.168.2.1 - ZBXTRAP 192.168.2.1
192.168.2.1 4 GigabitEthernet0/3 ethernetCsmacd administratively down
```

「監視データ」→「障害」の画面で、Router1の障害として検知されていることを確認します。

以上で、AWS上での監視サーバー(Zabbix)構築【8.監視登録(SNMP Trap)】の説明は完了です！