目次

WarpLink SPS の利用

WarpLink SPS[1] は、通信機器(サービスアダプタ[2])をネットワークに自動接続し、遠隔から集中管理するマネージメントシステム基盤です。
設定管理などのすべての機能を専用の Web 画面を通して一元的に行えるため、SaaS アプリケーションのように手軽に機器の管理を行うことが可能です。
VPN ルータの集中監視に加えて、無人環境に多数展開している IoT ゲートウェイを一元管理することが可能となり、VPN ルータ / IoT ゲートウェイに
掛かるトータルコストを大幅に削減できます。

準備

ご利用にはユーザ登録が必要です。
弊社営業部 (TEL: 0422-37-8112 E-Mail: sales@centurysys.co.jp) へお問い合わせ下さい。

設定

設定ファイルの編集 (armsd.conf)

ファームウェア v2.7.0 より、WarpLink SPS との接続を行うエージェント(armsd)を組み込んであります。
armsd の設定ファイル /etc/armsd/armsd.conf を編集します。

armsd.conf

```
distribution-id:    0000-0123-4567-8901-2345-6789-dead-beef
ls-sa-key:      hogehogefugafuga
#sa-model-name:     arms client daemon
#sa-version:        0.00

#path-iconfig:      /etc/armsd/initial-config
#path-state-cache:  /var/cache/armsd/state
#https-proxy-url:   http://192.168.0.1:8080/

#hb-disk-usage0:    /
#hb-traffic-if0:    eth0

### see sample scripts in /usr/share/armsd-X.X/examples
#script-app-event:  /etc/armsd/scripts/app-event
#script-clear:      /etc/armsd/scripts/clear
#script-command:    /etc/armsd/scripts/command
#script-post-pull:  /etc/armsd/scripts/post-pull
#script-reconfig:   /etc/armsd/scripts/reconfig
#script-start:      /etc/armsd/scripts/start
#script-status:     /etc/armsd/scripts/status
#script-stop:       /etc/armsd/scripts/stop
#script-reboot:     /etc/armsd/scripts/reboot
#script-line-ctrl:  /etc/armsd/scripts/line
#script-state-changed:  /etc/armsd/scripts/state-changed
```

distribution-id, ls-sa-key を設定しておきます。

起動設定 (Upstart)

/etc/default/armsd を編集し、自動起動するようにします。

armsd

```
START=yes
```

起動の確認

起動できているか確認してみます。

```
root@plum:~# ps ax|grep armsd
 1025 ?        Ss     0:03 /usr/sbin/armsd -D -f /etc/armsd/armsd.conf
31798 pts/1    S+     0:00 grep --color=auto armsd
```

log でも確認してみます。

```
Jun  7 13:20:18 plum armsd[1025]: armsd version 1.0.7, with libarms 5.41
(Release)
Jun  7 13:20:18 plum armsd[1025]: no state cache file - try LS-PULL
Jun  7 13:20:18 plum armsd[1025]: log callback: 2 - Pull from LS.
Jun  7 13:20:18 plum armsd[1025]: state changed: INITIAL -> LSPULL
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - calculated. ls max
retry 1500 times.
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - calculated. ls retry
interval 60 sec.
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - line: connecting(0):
DHCP
Jun  7 13:20:18 plum armsd[1025]: log callback: 21 - Line DHCP(0) Connected.
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - [-] Start rs-
solicitation
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - RS[0]: socket
prepared. connecting...
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - RS[0]: SSL connection
established. (TLSv1.2 - ECDHE-RSA-AES256-GCM-SHA384)
Jun  7 13:20:18 plum armsd[1025]: log callback: 2 - Connecting to LS
Jun  7 13:20:18 plum armsd[1025]: log callback: 3 - LS Access Done
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - [-] End rs-
solicitation
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - line:
disconnecting(0): DHCP
Jun  7 13:20:18 plum armsd[1025]: disable web proxy
Jun  7 13:20:18 plum armsd[1025]: log callback: 25 - Line DHCP(0)
Disconnected.
Jun  7 13:20:18 plum armsd[1025]: state changed: LSPULL -> RSPULL
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - calculated. rs max
retry 5 times.
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - calculated. rs retry
interval 60 sec.
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - line: connecting(0):
DHCP
Jun  7 13:20:18 plum armsd[1025]: log callback: 21 - Line DHCP(0) Connected.
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - [-] Start config-
solicitation
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - RS[0]:
https://202.221.50.98/arms/msg
```

```
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - RS[1]:
https://202.221.50.99/arms/msg
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - RS[0]: socket
prepared. connecting...
Jun  7 13:20:18 plum armsd[1025]: log callback: 200 - RS[0]: SSL connection
established. (TLSv1.2 - AES256-GCM-SHA384)
Jun  7 13:20:18 plum armsd[1025]: log callback: 5 - Connecting to RS
Jun  7 13:20:19 plum armsd[1025]: log callback: 6 - RS Access Done
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - RS End point:
https://202.221.50.98/arms/msg
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [-] End config-
solicitation
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - line:
disconnecting(0): DHCP
Jun  7 13:20:19 plum armsd[1025]: disable web proxy
Jun  7 13:20:19 plum armsd[1025]: log callback: 25 - Line DHCP(0)
Disconnected.
Jun  7 13:20:19 plum armsd[1025]: state changed: RSPULL -> PULLDONE
Jun  7 13:20:19 plum armsd[1025]: initial configuration succeeded
Jun  7 13:20:19 plum armsd[1025]: state cache saved
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [-] Start push-method-
query
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - RS[0]: socket
prepared. connecting...
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - RS[0]: SSL connection
established. (TLSv1.2 - AES256-GCM-SHA384)
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [-] End push-method-
query
Jun  7 13:20:19 plum armsd[1025]: log callback: 91 - Push method: tunnel
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#0:
https://202.221.50.98:443/arms/tunnel
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#1:
https://202.221.50.99:443/arms/tunnel
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#2:
https://[2001:0240:bb88:0000:0000:0000:0000:0262]:443/arms/tunnel
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#3:
https://[2001:0240:bb88:0000:0000:0000:0000:0263]:443/arms/tunnel
Jun  7 13:20:19 plum armsd[1025]: log callback: 70 - Start push confirmation
Jun  7 13:20:19 plum armsd[1025]: state changed: PULLDONE -> PUSH_SENDREADY
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#0: try to
connect 202.221.50.98:443
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#0: socket
prepared. connecting...
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#0: socket
connected.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#1: try to
connect 202.221.50.99:443
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#1: socket
prepared. connecting...
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#1: socket
connected.
```

```
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#2: address
family mismatched: 2001:0240:bb88:0000:0000:0000:0000:0262
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#2: closed.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#3: address
family mismatched: 2001:0240:bb88:0000:0000:0000:0000:0263
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#3: closed.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#1: SSL
connection established. (TLSv1.2 - AES256-GCM-SHA384)
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#1: sent http
header.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#0: SSL
connection established. (TLSv1.2 - AES256-GCM-SHA384)
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#0: sent http
header.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#0: received
http header.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [-] Start push-
confirmation-start
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - Start confirmation
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - tunnel#1: received
http header.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [-] Start push-
confirmation-start
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - Start confirmation
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [70418033] End push-
confirmation-start
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - Sent confirmation
request.  wait for response.
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [-] Start push-
confirmation-done
Jun  7 13:20:19 plum armsd[1025]: log callback: 200 - [70418033] End push-
confirmation-done
Jun  7 13:20:19 plum armsd[1025]: state changed: PUSH_SENDREADY -> PUSH_WAIT
Jun  7 13:20:19 plum armsd[1025]: log callback: 80 - Start heartbeat
(interval: 300 sec)
Jun  7 13:20:19 plum armsd[1025]: log callback: 82 -  heartbeat server:
202.221.50.96
Jun  7 13:20:19 plum armsd[1025]: log callback: 82 -  heartbeat server:
202.221.50.97
```

heartbeat が送信されていますので、接続は成功しているようです。

カスタマイズ

コールバックスクリプトの作成

WarpLink SPS のシステムに接続中に発生する各種イベント(コンフィグの適用など)は□armsd に登録し

たコールバックスクリプトにより処理を行います。

コールバックスクリプトには、下記があります。

```
#script-app-event:  /etc/armsd/scripts/app-event
#script-clear:      /etc/armsd/scripts/clear
#script-command:    /etc/armsd/scripts/command
#script-post-pull:  /etc/armsd/scripts/post-pull
#script-reconfig:   /etc/armsd/scripts/reconfig
#script-start:      /etc/armsd/scripts/start
#script-status:     /etc/armsd/scripts/status
#script-stop:       /etc/armsd/scripts/stop
#script-reboot:     /etc/armsd/scripts/reboot
#script-line-ctrl:  /etc/armsd/scripts/line
#script-state-changed:  /etc/armsd/scripts/state-changed
```

Name	Function	Arguments
app-event	IP アドレス変更の通知	
clear	ステータス初期化 (PUSH)	
command	任意コマンド	command <id> <requestfile> <resultfile>
post-pull	Pull 完了後に実行	
reconfig	コンフィグ変更 (PUSH)	reconfig <id> <version> <info> <configfile>
start	Pull 時実行	start <id> <version> <info> <configfile>
stop	モジュール停止	stop <id>
reboot	再起動	
line-ctrl	Pull 時の回線接続[3)]	line-ctrl <action> <line-type> <ifindex> [<id> <password> [<cid> <apn> <pdp>]]
state-changed	ステータス変化通知	state-changed <old-status> <new-status>

機器の起動時には、モジュールの個数分(5) “start” スクリプトが呼ばれます。
例として、次のような “start” スクリプトにすると、

start

```
#!/bin/bash

id=$1
version=$2
info=$3
config=$4

echo start script invoked at `date`: $0 $*

echo module-id=$id
echo version=$version
echo info=$info
echo config file is: $config:
echo -----
cat $config
```

```
echo -----
```

次のようなログが(Upstart のログに)記録されます。

```
start script invoked at Wed Jun 7 13:20:18 JST 2017:
/etc/armsd/scripts/start 2 0.0.0-0 /tmp/armsd.1025/tmp/0
module-id=2
version=0.0.0-0
info=
config file is: /tmp/armsd.1025/tmp/0:
-----
hogehoge
-----
start script invoked at Wed Jun 7 13:20:18 JST 2017:
/etc/armsd/scripts/start 3 0.0.0-0 /tmp/armsd.1025/tmp/1
module-id=3
version=0.0.0-0
info=
config file is: /tmp/armsd.1025/tmp/1:
-----
-----
start script invoked at Wed Jun 7 13:20:18 JST 2017:
/etc/armsd/scripts/start 4 0.0.0-0 /tmp/armsd.1025/tmp/2
module-id=4
version=0.0.0-0
info=
config file is: /tmp/armsd.1025/tmp/2:
-----
-----
start script invoked at Wed Jun 7 13:20:18 JST 2017:
/etc/armsd/scripts/start 0 0.0.0-0 /tmp/armsd.1025/tmp/3
module-id=0
version=0.0.0-0
info=
config file is: /tmp/armsd.1025/tmp/3:
-----
hoge1
http://www.yahoo.co.jp?file=hogehoge.img&hoge=fuga
-----
start script invoked at Wed Jun 7 13:20:19 JST 2017:
/etc/armsd/scripts/start 1 0.0.0-0 /tmp/armsd.1025/tmp/4
module-id=1
version=0.0.0-0
info=
config file is: /tmp/armsd.1025/tmp/4:
-----
-----
```

ファームウェア更新機能の実装

モジュール ID: 0 には “ ファームウェア(URL)” を割り当てていますので、ファームウェア更新機能を作成するには、

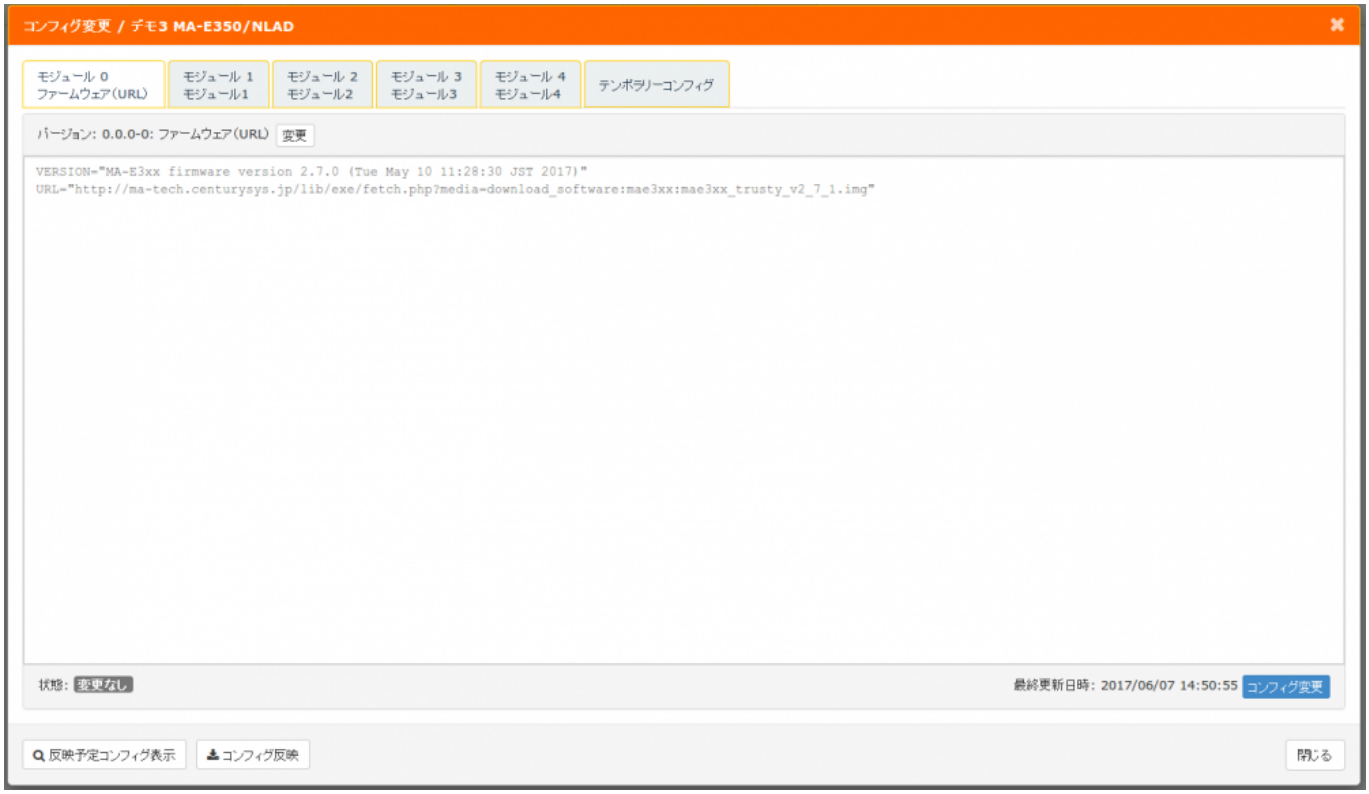
- VERSION[4] および URL を config に記述するようにする
- start / reconfig スクリプトで、config ファイルをパースしてファームウェア更新が必要か判定する

という実装が考えられます。

サンプルは次のようになるでしょうか。

reconfig

```
#! /bin/bash


firmup_check() {
  local config=$1
  local ma_version=`cat /etc/version`
  local VERSION="None"
  local URL=""

  . $config

  echo "MA version(current): \"$ma_version\""
  echo "Version in config: \"$VERSION\""

  if [ "$ma_version" != "$VERSION" ]; then
    echo "version unmatched, need to update firmware, firmware URL is
\"$URL\""
    # wget $URL -O /tmp/firmware.img  # (download firmware)
    # firmup /tmp/firmware.img        # (update firmware)
  else
    echo "version matched, not need to firm-update."
  fi
}

echo reconfig script invoked at `date`: $0 $*

# $1 is module-id
case "$1" in
  "0" ) firmup_check $4
      ;;
esac
```



実行例は下記のようになります。

syslog

```
reconfig script invoked at Wed Jun 7 14:51:02 JST 2017:
/etc/armsd/scripts/reconfig 0 0.0.0-0 /tmp/armsd.1025/candidate-
config/0
MA version(current): "MA-E3xx firmware version 2.7.0 (Tue May 9
11:28:30 JST 2017)"
Version in config: "MA-E3xx firmware version 2.7.1 (Tue May 10 11:28:30
JST 2017)"
version unmatched, need to update firmware, firmware URL is
"http://ma-tech.centurysys.jp/lib/exe/fetch.php?media=download_software
:mae3xx:mae3xx_trusty_v2_7_1.img"
```

1)
IIJ様 SACM の OEM
2)
SA
3)
SMF SDK を用いて独自 RS を構築している事業者のみ設定可能です。WarpLink SPS/SACM などのサービスを直接利用する場合には利用できません。
4)
無条件に実行されるため、バージョンをチェックして無駄なダウンロードを避けるため