

Amazon Web Services Simple Notification Service

This document describes an example EM7 RBA Action to send notification messages using AWS SNS. This is particularly useful for sending SMS messages without having a dedicated SMS gateway, but can be also used for other notification methods, such as publishing to AWS SQS.

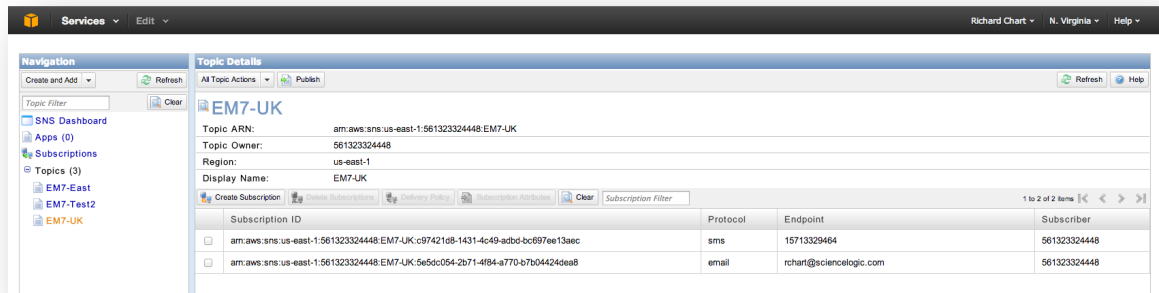
For further details of SNS, review this documentation:
<http://docs.aws.amazon.com/sns/latest/dg/welcome.html>

Important notes for sending SMS messages from SNS (as of March 2014):

- SMS messages can only be sent from SNS services deployed in the us-east-1 region
- SMS messages can only be sent to devices operating on US cellular networks
- The first 10 characters of the SNS topic Display Name are used as a prefix for the SMS message, the total message length of SMS messages can be no more than 140 single-byte characters (70 unicode).

SNS Topics

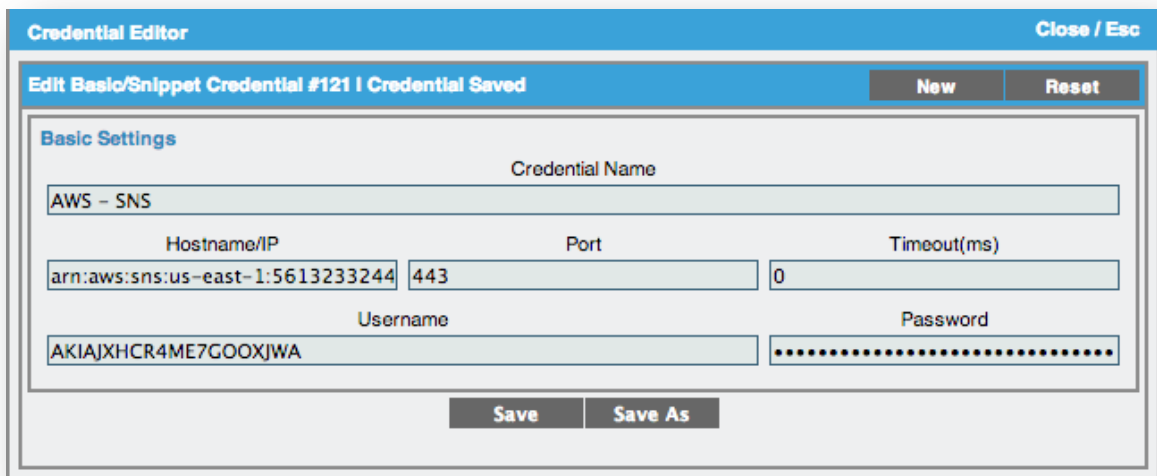
In order to use SNS for publishing notifications, you need at least one topic, with at least one subscriber, this will show in your SNS service panel on AWS something like this:



The most important piece of information for EM7 integration is the “Topic ARN”. This tells EM7 where to send notifications, it is included in the EM7 credential, along with your AWS access and AWS secret keys.

SNS Credential

We do not have an exactly matching credential form to include the “Topic ARN” in credentials, so we use an Basic/Snippet credential, with the Topic ARN stored in the Hostname field:



The screenshot shows a 'Credential Editor' window with a blue header bar containing 'Close / Esc'. Below the header is a title bar 'Edit Basic/Snippet Credential #121 | Credential Saved' with 'New' and 'Reset' buttons. The main area is titled 'Basic Settings' and contains the following fields:

Credential Name		
AWS - SNS		
Hostname/IP	Port	Timeout(ms)
arn:aws:sns:us-east-1:5613233244	443	0
Username		Password
AKIAJXHCR4ME7GOOXJWA		

At the bottom are 'Save' and 'Save As' buttons.

The username and password fields contain the AWSAccessKeyId and AWSSecretKey respectively.

RBA Action

The basic structure of an RBA action to publish an SNS message is simple. Using the boto Python package only two calls are required, firstly to connect to the SNS in the appropriate region, and then to publish the notification.

In the example shown below and in the PowerPack the remaining statements are for variable extraction and message formatting.

In the version of boto included in EM7 prior to 7.3.6, there is no ability to send explicitly different message structures to different subscriber types, however SMS devices will just receive the subject text, prefixed with the SNS topic display name. The version of boto included in 7.3.6 and beyond has more flexibility via the added `message_structure` argument to the `publish` method.

```
#####
#
# Use the AWS Simple Notification Service to send a notification message
#
# The message is published with both a message body and subject.
# SMS devices will receive the SNS Topic display name followed by
# the message subject.
#
#####

import boto.sns

AWSAccessKeyId = EM7_ACTION_CRED['cred_user']
AWSSecretKey = EM7_ACTION_CRED['cred_pwd']
topic_arn = EM7_ACTION_CRED['cred_host']
region_name = 'us-east-1'

# Message body structure matches default email structure
msg_body = "Severity: %s\nFirst Occurred: %s\nLast Occurred: %s\nOccurrences: %s\nSource: %s\nOrganization: %s\nDevice: %s\n\nMessage: %s\n\nSent by Automation Action: %s\n\nView this event at: %s\n" \
% (EM7_VALUES['%S'], EM7_VALUES['%D'], EM7_VALUES['%d'], EM7_VALUES['%c'], EM7_VALUES['%Z'], EM7_VALUES['%O'], EM7_VALUES['%X'], EM7_VALUES['%M'], EM7_VALUES['%N'], EM7_VALUES['%H'])

# Subject structure is <severity> on <device> in <org>: <event>
msg_subject = "%s on %s in %s: %s" % (EM7_VALUES['%S'], EM7_VALUES['%X'], EM7_VALUES['%O'], EM7_VALUES['%M'])

# Subject max len is 100 chars, otherwise error code 400 (undocumented)
msg_subject = msg_subject[:100]

sns_conn = boto.sns.connect_to_region(region_name, aws_access_key_id=AWSAccessKeyId, aws_secret_access_key=AWSSecretKey)

sns_conn.publish(topic=topic_arn, message=msg_body, subject=msg_subject)

EM7_RESULT = msg_subject
```

RBA Action Images

RBA action with Snippet credential:

The screenshot shows the 'Policy Editor' interface with the title 'Policy Editor | Action Saved | Editing Action [57]'. A 'Reset' button is in the top right. The form contains the following fields:

- Action Name:** Send notification using AWS SNS
- Action State:** [Enabled]
- Description:** (empty text area)
- Organization:** [System]
- Action Type:** Run a Snippet
- Snippet Credential:** [AWS - SNS]
- Action Run Context:** [Database]
- Snippet Code:** A text area containing the following code:

```
#####  
#  
# Use the AWS Simple Notification Service to send a notification message  
#  
# The message is published with both a message body and subject.  
# SMS devices will receive the SNS Topic display name followed by  
# the message subject.  
#  
#####  
  
import boto.sns  
  
AWSAccessKeyId = EM7_ACTION_CRED['cred_user']  
AWSSecretKey = EM7_ACTION_CRED['cred_pwd']
```

Typical log message from a successful run showing message subject line:

The screenshot shows the 'Event Actions Log | For Event [2185713]' with 'Refresh' and 'Guide' buttons. The log entry for '2014-03-10 14:24:12' reads:

Automation Policy SNS Automation action Send notification using AWS SNS ran Successfully
Message:Snippet action executed without incident
Result:MINOR on TL008-HQ-AP-01.MSTL008.local in Chicago: System or agent has recently restarted'