**How to Connect Amazon SNS to Amazon SQS Queues**

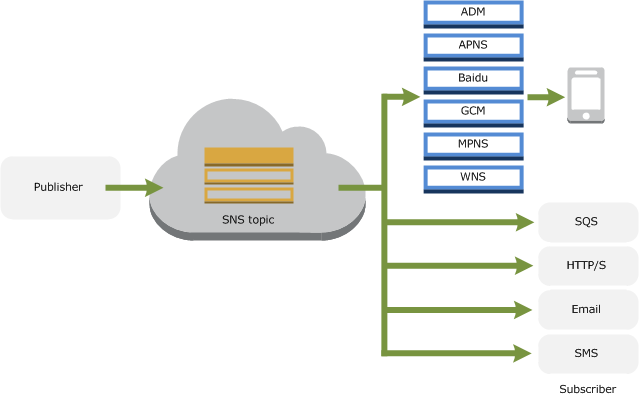
[August 7, 2015](http://budiirawan.com/how-to-connect-amazon-sns-to-amazon-sqs-queues/) Posted in [AWS](http://budiirawan.com/category/aws/) .

In this article, I want to share how to make SNS and SQS get along. Before we move further, we have a main question. What is the difference between SNS and SQS? I took the description from Amazon documentation below.

**Amazon SNS**

**Amazon SNS** allows applications to send time-critical messages to multiple subscribers through a “push” mechanism, eliminating the need to periodically check or “poll” for updates.

Here is the diagram to show how SNS works

[](http://budiirawan.com/wp-content/uploads/2015/08/sns-architecture.png)

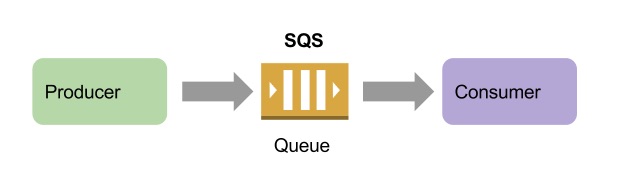
*SNS Diagram. Source: Amazon documentation*

SNS has topic which publisher can send message to. Then all subscribers that subscribe to that topic can get the messages immediately. The subscribers can be mobile apps because SNS supports mobile messaging platform e.g. GCM, APNS. In addition, it can be other things like SQS, HTTP/S url, email, lambda function or SMS.

**Amazon SQS**

**Amazon SQS** is a message queue service used by distributed applications to exchange messages through a polling model, and can be used to decouple sending and receiving components—without requiring each component to be concurrently available.

Here is a diagram to show how SQS works. It just simple implementation.

[](http://budiirawan.com/wp-content/uploads/2015/08/sqs-diagram.jpg)

*Simple SQS Diagram*

Sending email is one of implementation which SQS can be used. We can create a worker as consumer that receive messages from queue and then send the email. Another usage such as video encoding and thumbnail generator.

As summary, the main differences are:

SNS use push mechanism and SQS use polling mechanism.

SNS message can push messages to mobile devices or other subscribers directly. SQS need a worker to polling the messages.

SNS can’t persist the message but SQS can. By using SQS, you don’t need to worry to lose the message when the consumers can’t be reached due to network problem.

SNS terms are topic, publisher and subscriber. SQS terms are queue, producer and consumer.

By combining SNS and SQS together, you can deliver immediate and persisted message to applications. Very powerful!

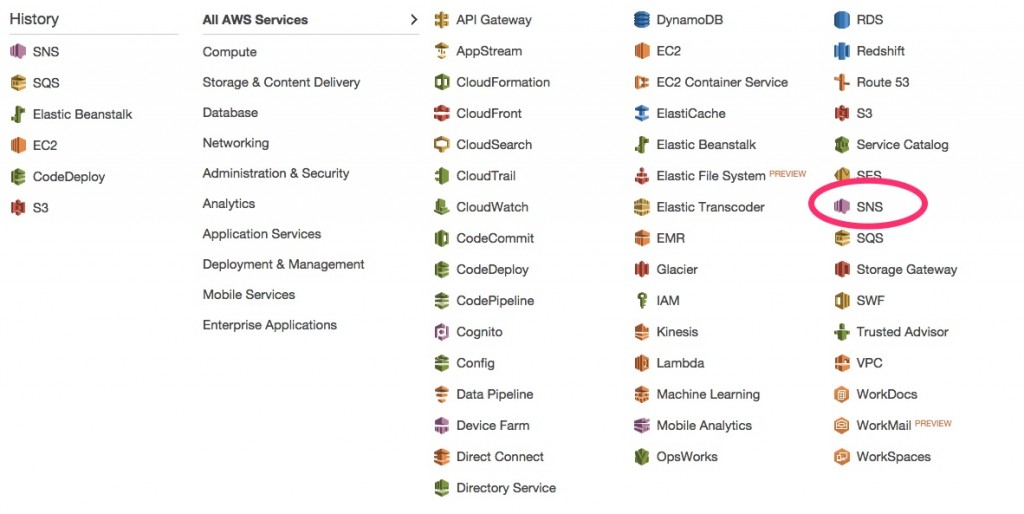
Let’s we move further how to connect them together.

**Create SNS Topic**

We have to create a topic first.

Go to **AWS Services** menu at top

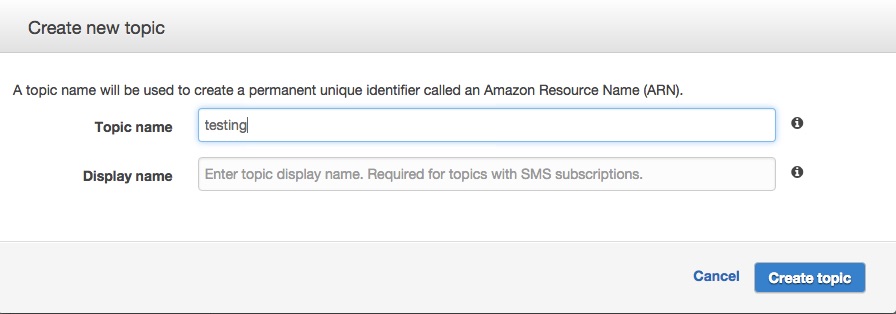
Choose **SNS**

[](http://budiirawan.com/wp-content/uploads/2015/08/sns-menu.jpg)

*Choose SNS menu*

Choose **Topics** and click **Create New Topic**

Input **Topic name** and topic **Display name** . Topic **Display name** is optional, it is used for SMS.

[](http://budiirawan.com/wp-content/uploads/2015/08/sns-create-topic-testing.jpg)

*Create new topic*

If correct, you will get your topic in topic table including the **ARN link** (the link that starts with “arn:…..”). Please note this link because it is important for next step.

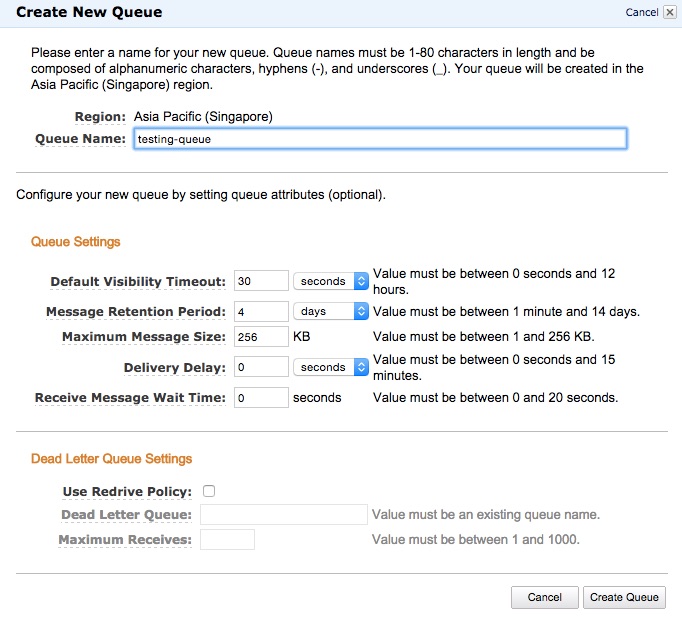
**Create SQS Queue**

Go to **AWS Services** menu at top

Choose **SQS**

Click **Create new queue.** It will display a dialog.

Just type queue name and click **Create queue**

[](http://budiirawan.com/wp-content/uploads/2015/08/sqs-create-new-queue.jpg)

*Create new queue*

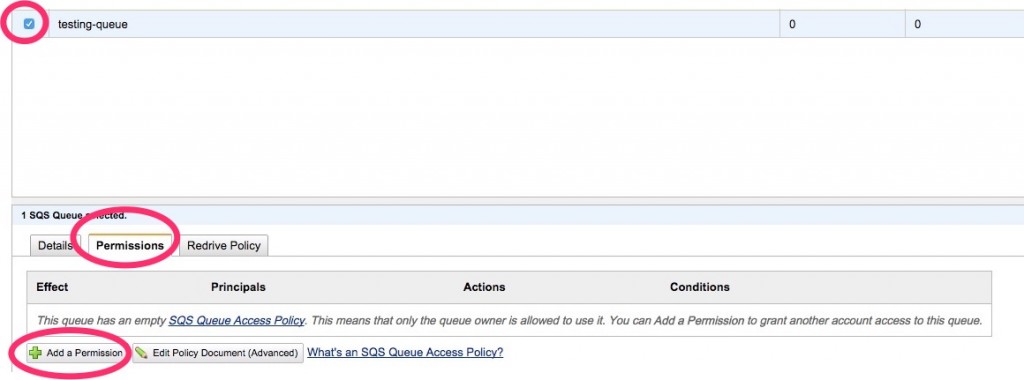
If correct, you will your queue in the list. This queue has ARN when you see in **Detail** tab at bottom.

**Configure SQS Queue Permission**

This step is very important to allow SNS to connect to SQS.

Click the queue name checkbox

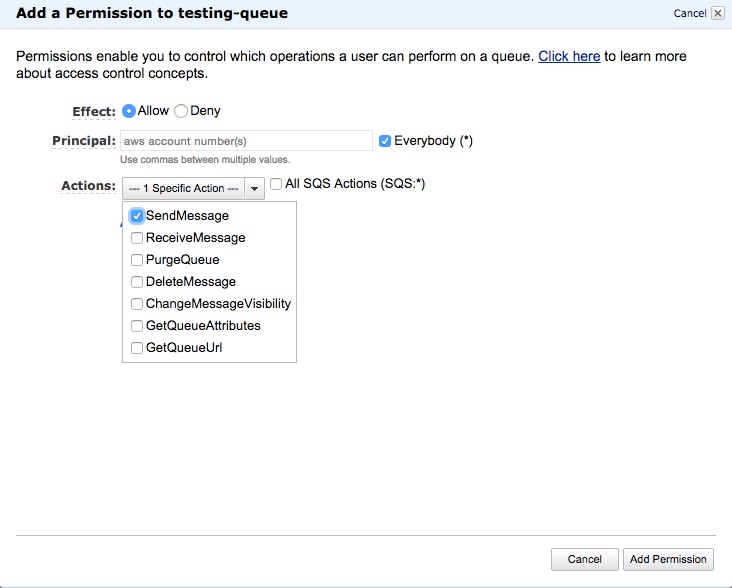
At the bottom, click **Permission** tab and click **Add a Permission**

[](http://budiirawan.com/wp-content/uploads/2015/08/SQS-add-permission.jpg)

*Add permission button*

In **Principal** , click **Everybody(\*)**

In **Actions** , just choose **Send Message** . We don’t need other SQS Actions beside this.

[](http://budiirawan.com/wp-content/uploads/2015/08/sqs-add-permission-to-queue.jpg)

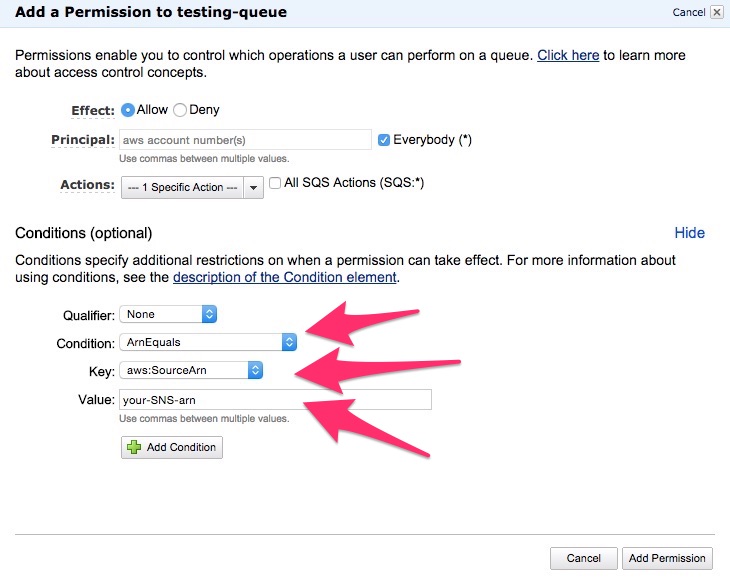
*Add permission dialog*

Click **Add Conditions** then you’ll see condition form.

Choose **Condition** to **ArnEquals**

Choose **Key** to **aws:SourceArn**

In **Value** , copy paste SNS ARN here.

[](http://budiirawan.com/wp-content/uploads/2015/08/sqs-add-permission-with-conditional.jpg)

*Specify conditions*

Click **Add Condition**

Then click **Add Permission**

**Create a SQS Subscriber**

We need to go back to SNS page.

Go to **AWS Services** menu at top

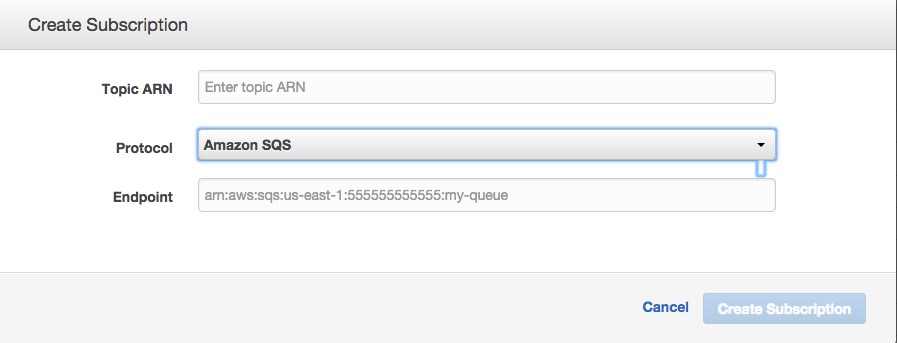
Choose **SNS**

Click **Subscriptions** in left menu

Create **Subscription**

Input your **Topic ARN**

In **Protocol** , choose **Amazon SQS**

in **Endpoint** , copy paste SQS queue ARN. (you need go back to SQS page to get this information)   
[](http://budiirawan.com/wp-content/uploads/2015/08/sns-create-subscription.jpg)

If success, you will see it in subscription list.

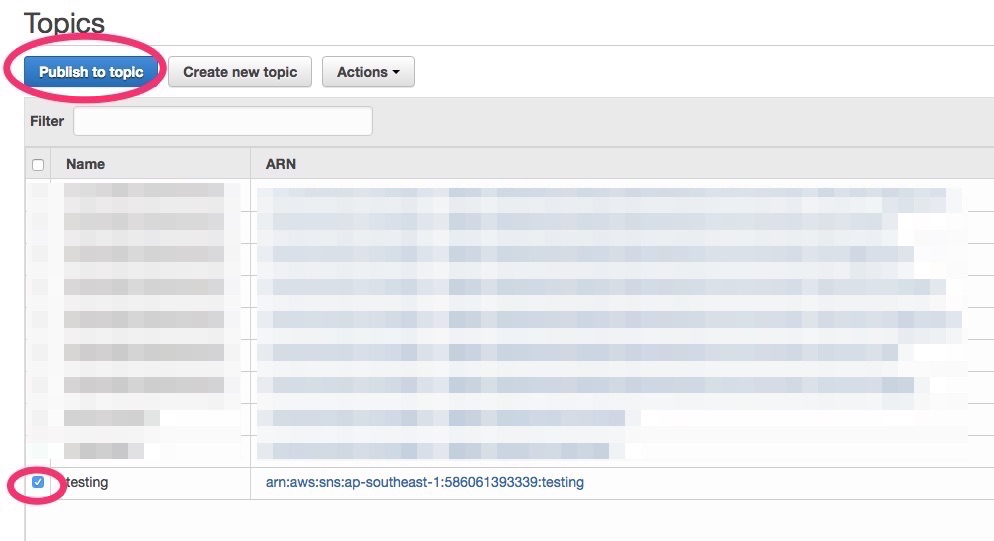
**Testing**

Let’s we test our setup by publishing a message to topic that we created. As result, the message is supposed to be delivered to our queue.

in SNS page, click **Topic** in left menu

Select the checkbox in topic name

Click **Publish to Topic**

[](http://budiirawan.com/wp-content/uploads/2015/08/sns-publish-topic.jpg)

*Publish topic*

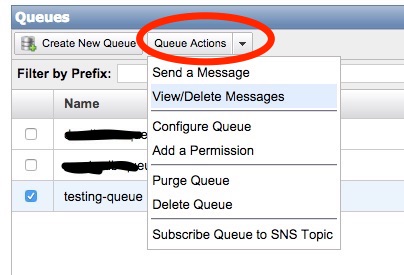
Insert **Subject** and **Message**

Click **Publish Message**

Let’s go to SQS page, if success, you will see a new created message in the queue. To see the message, follow steps below:

Click the queue checkbox

Choose **Queue Actions** to **View/Delete Messages**

[](http://budiirawan.com/wp-content/uploads/2015/08/sqs-view-message.jpg)

*View/Delete Queue Messages*

Click **Start Polling** for Messages

You will see the message there. Click **More Details** to see it as full message.

**Summary**

This article describes how to connect SNS and SQS together. By code implementation, you need AWS SDK to publish message to SNS and retrieve the message from SQS.