

# AWSimple PubSub

James Abel

SF Python Meetup

January 14, 2026

[j@abel.co](mailto:j@abel.co) @jamesabel [www.abel.co](http://www.abel.co)

James Abel - AWSimple

# AWSimple

- Simple Object-Oriented API on top of boto3. Emphasis on cost-effective, scale-to-zero cloud services.
  - Easy S3 access with locally cached S3 accesses. Reduces network traffic, AWS costs, and can speed up access.
  - Easy DynamoDB access with locally cached full table scans.
  - Supports moto mocking. Avoids AWS charges during testing.
  - PubSub (*today's topic*)
- 
- Featured on [PythonBytes Podcast Episode #224](#)
  - Repo: <https://github.com/jamesabel/awssimple>
  - Docs: <https://awssimple.readthedocs.io/>

# AWSimple.PubSub

- MxN PubSub
  - Any number of publishers
  - Any number of subscribers
- User provides a **channel** name (`str`) used by publishers (Pub) and subscribers (Sub)
- Automatically connects SNS → SQS in AWS
  - One Topic, one or more Queues, per **channel**
  - SNS Topic and SQS Queue names are based on channel name, but created to not collide with other existing Topics and Queue
- Unused Queues are aged out
  - SQS Queues deleted after 30 days of unuse
  - Automatically re-created if used again
- Both run on Threads for responsiveness
- Scales to Zero
- Sub “Long Poll” on SQS

```
from awssimple import Pub
```

## PUB

```
pub = Pub("awssimple_pubsub_example") # channel
```

```
pub.start()
```

```
while input_message := input(f"Message to publish (or Enter to  
exit):").strip():
```

```
    pub.publish({"message": input_message})
```

```
pub.request_exit()
```

```
from awssimple import Sub
```

## SUB

```
sub = Sub("awssimple_pubsub_example", sub_callback=print) # channel
```

```
sub.start()
```

```
input("Press Enter to exit...\n")
```

```
sub.request_exit()
```

# DEMO