

---

# **Wellcome AWS Utils Documentation**

***Release 1.0.0***

**Wellcome Digital Platform**

**Jun 26, 2018**



---

## Contents

---

<b>1</b>	<b>API reference</b>	<b>3</b>
1.1	Deployment utilities . . . . .	3
1.2	DynamoDB events . . . . .	3
1.3	DynamoDB utilities . . . . .	3
1.4	ECS . . . . .	3
1.5	S3 . . . . .	3
1.6	SNS . . . . .	3
1.7	SQS . . . . .	3
<b>2</b>	<b>Changelog</b>	<b>5</b>
2.1	2.1.2 - 2018-06-26 . . . . .	5
2.2	2.1.1 - 2018-06-04 . . . . .	5
2.3	2.1.0 - 2018-06-04 . . . . .	5
2.4	2.0.2 - 2018-06-04 . . . . .	6
2.5	2.0.1 - 2018-01-12 . . . . .	6
2.6	2.0.0 - 2017-11-29 . . . . .	6
2.7	1.1.0 - 2017-11-15 . . . . .	6
2.8	1.0.0 - 2017-11-07 . . . . .	6



This package is a collection of utilities written at Wellcome for interacting with AWS.

Some of these utilities are very specific to Wellcome projects, others are more generic and should be generally useful.

The best place to start is the [API reference](#), which describes all the utilities the package provides. It's a bit sparse at the moment, but hopefully we'll expand it soon!



## **1.1 Deployment utilities**

## **1.2 DynamoDB events**

## **1.3 DynamoDB utilities**

## **1.4 ECS**

## **1.5 S3**

## **1.6 SNS**

## **1.7 SQS**





## CHAPTER 2

---

### Changelog

---

This is a record of all releases of wellcome\_aws\_utils.

#### 2.1 2.1.2 - 2018-06-26

Previously sending a message with `sns_utils.publish_sns_message` would print a message upon success. Now this message is only logged at debug level.

#### 2.2 2.1.1 - 2018-06-04

Now `@log_on_error` can be used to decorate functions with arbitrary arguments/keyword arguments.

#### 2.3 2.1.0 - 2018-06-04

This adds a new method: `lambda_utils.log_on_error`. This can be used to decorate the main function for a Lambda, and logs the event/context if the Lambda throws an unexpected exception.

For example, running the following snippet:

```
@log_on_error
def handler(event, context=None):
    if event == {1: '1', 2: '2'}:
        raise ValueError

handler(event={'foo': 'bar'})
handler(event='99 green bottles' * 99)
handler(event={1: '1', 2: '2'})
```

gives the following output:

This makes it easier to debug failed Lambdas, but without the expense of logging every event that a Lambda receives.

## 2.4 2.0.2 - 2018-06-04

Previously sending a message with `sns_utils.publish_sns_message` would log the entire SNS response.

Now the response is only logged if the SNS message is unsuccessful.

## 2.5 2.0.1 - 2018-01-12

This fixes a bug in `s3_utils.parse_s3_record`. If the key of a changed file included a character which is usually quoted in URLs (e.g. `+`), a parsed record from the S3 event stream would use the URL-quoted form of the object key.

For example, a change to `s3://example/foo+bar` would become `foo%2Bbar`.

This version unquotes the key when parsing the event.

## 2.6 2.0.0 - 2017-11-29

Replacing the `DynamoImageFactory` and `DynamoImage` classes with `DynamoEventFactory` and `DynamoEvent`

- Perform quite a bit of sanity checking on event object received
- `DynamoEvent` can: - return old and new images (if available) - return modified keys only - return deserialized or otherwise images and keys based on params

## 2.7 1.1.0 - 2017-11-15

Deprecates `sns_utils.extract_json_message` in favour of `sns_utils.extract_sns_messages_from_lambda_event`.

`extract_sns_messages_from_lambda_event` provides: - better error reporting if the event is malformed - loops over all available records from event not just the first - returns subject along with the json decoded message

This release also adds `UnWellcomeException` which will be used as the base exception for new errors.

## 2.8 1.0.0 - 2017-11-07

First production release!