
Boto3 Assistant

Release 0.0.1

Jun 27, 2018

1	boto3_assistant	3
1.1	boto3_assistant package	3
	Python Module Index	7

A collection of helper functions for working with Boto3!

1.1 boto3_assistant package

1.1.1 Submodules

1.1.2 boto3_assistant.account module

Functions to retrieve information about the callers account.

`boto3_assistant.account.get_account_id()`
Get the users account id.

Return: `account_id`: The callers AWS account id.

`boto3_assistant.account.get_alias()`
Get the users account alias.

Return: `alias`: The first known account alias.

`boto3_assistant.account.get_region()`
Get the current region.

Return: `region`: The region.

1.1.3 boto3_assistant.cloudformation module

Functions to retrieve information about cloudformation stacks.

`boto3_assistant.cloudformation.list_parameters(url)`
Get the parameters that are defined in a cloudformation template.

Parameters: `url` (str): The url to a cfn template in an accessible S3 bucket.

Return: `parameters`: A collection parameters in the cfn stack.

1.1.4 boto3_assistant.iam module

Functions for interacting with the IAM service.

`boto3_assistant.iam.add_policy(role_name, policy_name)`

Attach an IAM Policy to a Role.

Parameters: `role_name` (str): The name of the role to attach the policy to. `policy_name` (str): The name of the policy to attach to the role.

`boto3_assistant.iam.create_role(name, description)`

Create a role that is assumable by the callers account.

The role will be created with no policies.

Parameters: `name` (str): The name of the role to create. `description` (str): The description for the role to create.

Returns: `role`: Information about the role that is created.

`boto3_assistant.iam.list_roles(prefix)`

List all the roles for the given path prefix.

Parameters: `prefix` (str): The prefix to filter by.

`boto3_assistant.iam.put_role_policy(role_name, policy_name, policy)`

Add or update an inline role policy document.

Parameters: `role_name` (str): The name of the role to update. `policy_name` (str): The name of the policy to update. `policy` (obj): The policy document to update.

1.1.5 boto3_assistant.s3 module

`boto3_assistant.s3.create_bucket(name, region)`

Create a private versioned S3 bucket in the specified region.

Parameters: `name` (str): The name of the S3 bucket to create.

`region` (str): The name of the region to create the bucket in.

`boto3_assistant.s3.delete_bucket(name)`

Empty an S3 bucket and delete it.

Parameters: `name` (str): The name of the S3 bucket to delete.

`boto3_assistant.s3.delete_folder(name, prefix)`

Delete all objects under a given prefix.

Parameters: `name` (str): The name of the S3 bucket.

`prefix` (str): The root prefix to delete objects under.

`boto3_assistant.s3.download_file(name, prefix, download_location)`

Download a file from S3 to a local directory.

Parameters: `name` (str): The name of the S3 bucket.

`prefix` (str): The prefix of the file to download.

`download_location` (str): The path to the place to download the file to.

`boto3_assistant.s3.download_versioned_file(bucket_name, key, download_location, version)`

Download a versioned object from an S3 bucket.

Parameters: `bucket_name` (str): The name of the bucket to upload to. `key` (str): The key to upload to. `download_location` (str): The path to a local file to download to. `version` (str): The S3 version id of the object.

`boto3_assistant.s3.empty_bucket(bucket_name)`

`boto3_assistant.s3.exists(bucket_name, prefix)`

Determines if an object with the given prefix exists in a S3 bucket.

Parameters: `bucket_name` (str): The name of the S3 bucket `prefix` (str): The prefix of the object to test for.

Return Type: bool

`boto3_assistant.s3.get_file(name, prefix)`

Get the S3 Object Summary for the given prefix.

Parameters: `name` (str): The name of the S3 bucket.

`prefix` (str): The prefix of the file to get the summary for.

Returns: `object_summary`: Details about the requested file.

`boto3_assistant.s3.get_file_size(bucket, prefix)`

`boto3_assistant.s3.get_previous_version(bucket_name, key)`

Retrieve the previous version id of the specified key.

Parameters: `bucket_name` (str): The name of the S3 bucket to search in. `key` (str): The key of the S3 object to get the version for.

Return Type: str or None

Returns:

version_id (str): **The version id of the previous version.** If there was no previous version then None is returned.

`boto3_assistant.s3.get_sub_folders(name, prefix)`

Get what would usually be considered sub folders of an S3 prefix.

Parameters: `name` (str): The name of the S3 bucket.

`prefix` (str): The S3 prefix to search under.

`boto3_assistant.s3.is_deleted(bucket_name, prefix)`

Determines if the latest version of an object is a delete marker

Parameters: `bucket_name` (str): The name of the bucket to check in. `prefix` (str): The prefix of the object to test.

Return Type: bool

`boto3_assistant.s3.list_all_buckets()`

Get a list of all the buckets you have access to see.

Return: `buckets`: A list of all the buckets.

`boto3_assistant.s3.list_files_in_bucket(bucket_name, prefix, token=None)`

Returns a complete list of all the objects in the bucket, making recursive calls to S3 with the next continuation token.

Parameters: `bucket_name` (str): The name of the S3 bucket `prefix` (str): The prefix to list `token` (str): The optional boto3 continuation token

Return Type: list

Returns: A list of s3 objects

Syntax:

```
[
    { 'Key':          'string',      'LastModified':      datetime(2015,      1,      1),
      'ETag':         'string',      'Size':           123,      'StorageClass':    'STAN-
DARD'|'REDUCED_REDUNDANCY'|'GLACIER'|'STANDARD_IA'|'ONEZONE_IA',
      'Owner': {
          'DisplayName': 'string', 'ID': 'string'
      }
    },
]
```

`boto3_assistant.s3.upload_file(name, prefix, file_path)`

Upload a file from a local directory to an S3 bucket.

Parameters: name (str): The name of the S3 bucket.

prefix (str): The prefix to upload the file to.

file_path (str): The path to the file to upload to S3.

1.1.6 Module contents

Nothing to see here

b

- `boto3_assistant`, 6
- `boto3_assistant.account`, 3
- `boto3_assistant.cloudformation`, 3
- `boto3_assistant.iam`, 4
- `boto3_assistant.s3`, 4

A

add_policy() (in module boto3_assistant.iam), 4

B

boto3_assistant (module), 6

boto3_assistant.account (module), 3

boto3_assistant.cloudformation (module), 3

boto3_assistant.iam (module), 4

boto3_assistant.s3 (module), 4

C

create_bucket() (in module boto3_assistant.s3), 4

create_role() (in module boto3_assistant.iam), 4

D

delete_bucket() (in module boto3_assistant.s3), 4

delete_folder() (in module boto3_assistant.s3), 4

download_file() (in module boto3_assistant.s3), 4

download_versioned_file() (in module boto3_assistant.s3), 4

E

empty_bucket() (in module boto3_assistant.s3), 5

exists() (in module boto3_assistant.s3), 5

G

get_account_id() (in module boto3_assistant.account), 3

get_alias() (in module boto3_assistant.account), 3

get_file() (in module boto3_assistant.s3), 5

get_file_size() (in module boto3_assistant.s3), 5

get_previous_version() (in module boto3_assistant.s3), 5

get_region() (in module boto3_assistant.account), 3

get_sub_folders() (in module boto3_assistant.s3), 5

I

is_deleted() (in module boto3_assistant.s3), 5

L

list_all_buckets() (in module boto3_assistant.s3), 5

list_files_in_bucket() (in module boto3_assistant.s3), 5

list_parameters() (in module boto3_assistant.cloudformation), 3

list_roles() (in module boto3_assistant.iam), 4

P

put_role_policy() (in module boto3_assistant.iam), 4

U

upload_file() (in module boto3_assistant.s3), 6