
Moto Documentation

Release 4.1.3.dev

Steve Pulec

Feb 05, 2023

GETTING STARTED

1	Getting Started	3
2	Additional Resources	5
2.1	Getting Started with Moto	5
2.2	Non-Python SDK's / Server Mode	11
2.3	FAQ	14
2.4	IAM-like Access Control	15
2.5	AWS Config Support	16
2.6	Multi-Account support	19
2.7	Configuration Options	21
2.8	Implemented Services	29
2.9	Contributing	236
2.10	Development Installation	237
2.11	Architecture	237
2.12	New Features	239
2.13	PR Checklist	240
2.14	FAQ for Developers	241
2.15	Development Tips	243
2.16	Intercepting URLs	243
2.17	Writing tests	244
2.18	Utilities	245
2.19	State Transition Management	247
	Index	249

A library that allows you to easily mock out tests based on [AWS infrastructure](#).

GETTING STARTED

If you've never used `moto` before, you should read the *Getting Started with Moto* guide to get familiar with `moto` and its usage.

ADDITIONAL RESOURCES

- [Moto Source Repository](#)
- [Moto Issue Tracker](#)

2.1 Getting Started with Moto

2.1.1 Installing Moto

You can use `pip` to install the latest released version of `moto`, and specify which service(s) you will use:

```
pip install 'moto[ec2,s3,..]'
```

This will install Moto, and the dependencies required for that specific service.

If you don't care about the number of dependencies, or if you want to mock many AWS services:

```
pip install 'moto[all]'
```

If you want to install `moto` from source:

```
git clone git://github.com/getmoto/moto.git
cd moto
pip install '.[all]'
```

2.1.2 Moto usage

For example, we have the following code we want to test:

```
import boto3

class MyModel:
    def __init__(self, name, value):
        self.name = name
        self.value = value

    def save(self):
        s3 = boto3.client("s3", region_name="us-east-1")
        s3.put_object(Bucket="mybucket", Key=self.name, Body=self.value)
```

There are several ways to verify that the value will be persisted successfully.

Decorator

With a decorator wrapping, all the calls to S3 are automatically mocked out.

```
import boto3
from moto import mock_s3
from mymodule import MyModel

@mock_s3
def test_my_model_save():
    conn = boto3.resource("s3", region_name="us-east-1")
    # We need to create the bucket since this is all in Moto's 'virtual' AWS account
    conn.create_bucket(Bucket="mybucket")

    model_instance = MyModel("steve", "is awesome")
    model_instance.save()

    body = conn.Object("mybucket", "steve").get()[
        "Body"].read().decode("utf-8")

    assert body == "is awesome"
```

Context manager

Same as the Decorator, every call inside the with statement is mocked out.

```
def test_my_model_save():
    with mock_s3():
        conn = boto3.resource("s3", region_name="us-east-1")
        conn.create_bucket(Bucket="mybucket")

        model_instance = MyModel("steve", "is awesome")
        model_instance.save()

        body = conn.Object("mybucket", "steve").get()[
            "Body"].read().decode("utf-8")

        assert body == "is awesome"
```

Raw

You can also start and stop the mocking manually.

```
def test_my_model_save():
    mock = mock_s3()
    mock.start()

    conn = boto3.resource("s3", region_name="us-east-1")
    conn.create_bucket(Bucket="mybucket")

    model_instance = MyModel("steve", "is awesome")
```

(continues on next page)

(continued from previous page)

```
model_instance.save()

body = conn.Object("mybucket", "steve").get()["Body"].read().decode("utf-8")

assert body == "is awesome"

mock.stop()
```

Unittest usage

If you use `unittest` to run tests, and you want to use `moto` inside `setUp`, you can do it with `.start()` and `.stop()` like:

```
import unittest
from moto import mock_s3
import boto3

def func_to_test(bucket_name, key, content):
    s3 = boto3.resource("s3")
    object = s3.Object(bucket_name, key)
    object.put(Body=content)

class MyTest(unittest.TestCase):
    mock_s3 = mock_s3()
    bucket_name = "test-bucket"
    def setUp(self):
        self.mock_s3.start()

        # you can use boto3.client("s3") if you prefer
        s3 = boto3.resource("s3")
        bucket = s3.Bucket(self.bucket_name)
        bucket.create()

    def tearDown(self):
        self.mock_s3.stop()

    def test(self):
        content = b"abc"
        key = "/path/to/obj"

        # run the file which uploads to S3
        func_to_test(self.bucket_name, key, content)

        # check the file was uploaded as expected
        s3 = boto3.resource("s3")
        object = s3.Object(self.bucket_name, key)
        actual = object.get()["Body"].read()
        self.assertEqual(actual, content)
```

Class Decorator

It is also possible to use decorators on the class-level.

The decorator is effective for every test-method inside your class. State is not shared across test-methods.

```
@mock_s3
class TestMockClassLevel(unittest.TestCase):
    def setUp(self):
        s3 = boto3.client("s3", region_name="us-east-1")
        s3.create_bucket(Bucket="mybucket")

    def test_creating_a_bucket(self):
        # 'mybucket', created in setUp, is accessible in this test
        # Other clients can be created at will

        s3 = boto3.client("s3", region_name="us-east-1")
        s3.create_bucket(Bucket="bucket_inside")

    def test_accessing_a_bucket(self):
        # The state has been reset before this method has started
        # 'mybucket' is recreated as part of the setUp-method
        # 'bucket_inside' however, created inside the other test, no longer exists
        pass
```

Note: A `tearDown`-method can be used to destroy any buckets/state, but because state is automatically destroyed before a test-method start, this is not strictly necessary.

Stand-alone server mode

Moto also comes with a stand-alone server allowing you to mock out an AWS HTTP endpoint. For testing purposes, it's extremely useful even if you don't use Python.

```
$ moto_server -p3000
* Running on http://127.0.0.1:3000/
```

However, this method isn't encouraged if you're using `boto3`, the best solution would be to use a decorator method. See *Non-Python SDK's / Server Mode* for more information.

2.1.3 Recommended Usage

There are some important caveats to be aware of when using `moto`:

How do I avoid tests from mutating my real infrastructure

You need to ensure that the mocks are actually in place.

1. Ensure that your tests have dummy environment variables set up:

```
export AWS_ACCESS_KEY_ID='testing'
export AWS_SECRET_ACCESS_KEY='testing'
export AWS_SECURITY_TOKEN='testing'
export AWS_SESSION_TOKEN='testing'
export AWS_DEFAULT_REGION='us-east-1'
```

2. Do not embed credentials directly in your code. This is always considered bad practice, regardless of whether you use Moto. It also makes it impossible to configure fake credentials for testing purposes.
3. **VERY IMPORTANT:** ensure that you have your mocks set up *BEFORE* your *boto3* client is established. This can typically happen if you import a module that has a *boto3* client instantiated outside of a function. See the *pesky imports* section below on how to work around this.

Note: By default, the region must be one supported by AWS, see [Can I mock the default AWS region?](#) for how to change this.

Example on usage

If you are a user of *pytest*, you can leverage *pytest fixtures* to help set up your mocks and other AWS resources that you would need.

Here is an example:

```
@pytest.fixture(scope="function")
def aws_credentials():
    """Mocked AWS Credentials for moto."""
    os.environ["AWS_ACCESS_KEY_ID"] = "testing"
    os.environ["AWS_SECRET_ACCESS_KEY"] = "testing"
    os.environ["AWS_SECURITY_TOKEN"] = "testing"
    os.environ["AWS_SESSION_TOKEN"] = "testing"
    os.environ["AWS_DEFAULT_REGION"] = "us-east-1"

@pytest.fixture(scope="function")
def s3(aws_credentials):
    with mock_s3():
        yield boto3.client("s3", region_name="us-east-1")
```

In the code sample above, all of the AWS/mocked fixtures take in a parameter of *aws_credentials*, which sets the proper fake environment variables. The fake environment variables are used so that *botocore* doesn't try to locate real credentials on your system.

Next, once you need to do anything with the mocked AWS environment, do something like:

```
def test_create_bucket(s3):
    # s3 is a fixture defined above that yields a boto3 s3 client.
    # Feel free to instantiate another boto3 S3 client -- Keep note of the region though.
    s3.create_bucket(Bucket="somebucket")
```

(continues on next page)

(continued from previous page)

```
result = s3.list_buckets()
assert len(result["Buckets"]) == 1
assert result["Buckets"][0]["Name"] == "somebucket"
```

What about those pesky imports

Recall earlier, it was mentioned that mocks should be established `__BEFORE__` the clients are set up. One way to avoid import issues is to make use of local Python imports – i.e. import the module inside of the unit test you want to run vs. importing at the top of the file.

Example:

```
def test_something(s3):
    from some.package.that.does.something.with.s3 import some_func # <-- Local import
    ↪for unit test
    # ^^ Importing here ensures that the mock has been established.

    some_func() # The mock has been established from the "s3" pytest fixture, so this
    ↪function that uses
    # a package-level S3 client will properly use the mock and not reach
    ↪out to AWS.
```

Patching the client or resource

If it is not possible to rearrange imports, we can patch the boto3-client or resource after the mock has started. See the following code sample:

```
# The client can come from an import, an __init__-file, wherever..
outside_client = boto3.client("s3")
s3 = boto3.resource("s3")

@mock_s3
def test_mock_works_with_client_or_resource_created_outside():
    from moto.core import patch_client, patch_resource
    patch_client(outside_client)
    patch_resource(s3)

    assert outside_client.list_buckets()["Buckets"] == []

    assert list(s3.buckets.all()) == []
```

This will ensure that the boto3 requests are still mocked.

Other caveats

For Tox, Travis CI, and other build systems, you might need to also perform a `touch ~/.aws/credentials` command before running the tests. As long as that file is present (empty preferably) and the environment variables above are set, you should be good to go.

2.2 Non-Python SDK's / Server Mode

Moto has a stand-alone server mode. This allows you to use Moto with any of the official AWS SDK's.

2.2.1 Installation

Install the required dependencies using:

```
pip install moto[server]
```

You can then start it like this:

```
$ moto_server
```

You can also pass the port:

```
$ moto_server -p3000
* Running on http://127.0.0.1:3000/
```

If you want to be able to use the server externally you can pass an IP address to bind to as a hostname or allow any of your external interfaces with 0.0.0.0:

```
$ moto_server -H 0.0.0.0
* Running on http://0.0.0.0:5000/
```

Please be aware this might allow other network users to access your server.

2.2.2 Start within Python

It is possible to start this server from within Python, in a separate thread. By default, this server will start on port 5000, but this is configurable.

```
from moto.server import ThreadedMotoServer
server = ThreadedMotoServer()
server.start()
# run tests
client = boto3.client("service", endpoint_url="http://localhost:5000")
...
server.stop()
```

Note that the `ThreadedMotoServer` and the decorators act on the same state, making it possible to combine the two approaches. See the following example:

```

class TestThreadedMotoServer(unittest.TestCase):

    def setUp(self):
        self.server = ThreadedMotoServer()
        self.server.start()

    def tearDown(self):
        self.server.stop()

    @mock_s3
    def test_load_data_using_decorators(self):
        server_client = boto3.client("s3", endpoint_url="http://127.0.0.1:5000")
        server_client.create_bucket(Bucket="test")

        in_mem_client = boto3.client("s3")
        buckets = in_mem_client.list_buckets()["Buckets"]
        [b["Name"] for b in buckets].should.equal(["test"])

```

This example shows it is possible to create state using the ThreadedMotoServer, and access that state using the usual decorators. Note that the decorators will destroy any resources on start, so make sure to not accidentally destroy any resources created by the ThreadedMotoServer that should be kept.

2.2.3 Run using Docker

You could also use the official Docker image from <https://hub.docker.com/r/motoserver/moto/tags>:

```
docker run motoserver/moto:latest
```

2.2.4 Example Usage

To use Moto in your tests, pass the *endpoint_url*-parameter to the SDK of your choice.

In Python:

```

boto3.resource(
    service_name='s3',
    region_name='us-west-1',
    endpoint_url='http://localhost:5000'
)

```

In Java:

```

AmazonSQS sqs = new AmazonSQSClient();
sqs.setRegion(Region.getRegion(Regions.US_WEST_2));
sqs.setEndpoint("http://localhost:5000");

```

In Scala:

```

val region = Region.getRegion(Regions.US_WEST_2).getName
val serviceEndpoint = "http://localhost:5000"
val config = new AwsClientBuilder.EndpointConfiguration(serviceEndpoint, region)
val amazonSqs = AmazonSQSClientBuilder.standard().withEndpointConfiguration(config).
↳ build

```

(continues on next page)

(continued from previous page)

In Terraform:

```
provider "aws" {  
  region                = "us-east-1"  
  skip_credentials_validation = true  
  skip_metadata_api_check   = true  
  skip_requesting_account_id = true  
  s3_force_path_style       = true  
  
  endpoints {  
    lambda = "http://localhost:5000"  
  }  
}
```

See the [Terraform Docs](#) for more information.

Other languages:

- [Java](#)
- [Ruby](#)
- [Javascript](#)

2.2.5 Use ServerMode using the decorators

It is possible to call the MotoServer for tests that were written using decorators. The following environment variables can be set to achieve this:

```
TEST_SERVER_MODE=true
```

Whenever a mock-decorator starts, Moto will:

1. Send a reset-request to `http://localhost:5000`, removing all state that was kept
2. Add the `endpoint_url` parameter to boto3, so that all requests will be made to `http://localhost:5000`.

Optionally, the `http://localhost:5000` endpoint can be overridden by:

```
TEST_SERVER_MODE_ENDPOINT=http://moto-server:5000
```

This can be used for example in case of docker-compose configuration that runs Moto server in a separate service container.

Calling the reset-API ensures the same behaviour as normal decorators, where the complete state is removed. It is possible to keep the state in between tests, using this environment variable:

```
MOTO_CALL_RESET_API=false
```

2.2.6 Dashboard

Moto comes with a dashboard to view the current state of the system:

```
http://localhost:5000/moto-api/
```

2.2.7 Reset API

An internal API endpoint is provided to reset the state of all of the backends. This will remove all S3 buckets, EC2 servers, etc.:

```
requests.post("http://motoapi.amazonaws.com/moto-api/reset")
```

2.2.8 Install with Homebrew

Moto is also available to install using [Homebrew](#), which makes it much easier to manage if you're not using Python as your primary development language.

Once Homebrew is installed, you can install Moto by running:

```
brew install moto
```

To make the Moto server start up automatically when you log into your computer, you can run:

```
brew services start moto
```

2.2.9 Caveats

The standalone server has some caveats with some services. The following services require that you update your hosts file for your code to work properly:

1. *s3-control*

For the above services, this is required because the hostname is in the form of *AWS_ACCOUNT_ID.localhost*. As a result, you need to add that entry to your host file for your tests to function properly.

2.3 FAQ

2.3.1 Is Moto concurrency safe?

No. Moto is not designed for multithreaded access/multiprocessing.

2.3.2 Why am I getting RUST errors when installing Moto?

Moto has a dependency on the pip-module *cryptography*. As of Cryptography ≥ 3.4 , this module requires Rust as a dependency. Most OS/platforms will support the installation of Rust, but if you're getting any errors related to this, see the cryptography documentation for more information: <https://cryptography.io/en/latest/installation/#rust>

2.3.3 Can I mock the default AWS region?

By default, Moto only allows valid regions, supporting the same regions that AWS supports.

If you want to mock the default region, as an additional layer of protection against accidentally touching your real AWS environment, you can disable this validation:

```
os.environ["MOTO_ALLOW_NONEXISTENT_REGION"] = True
os.environ["AWS_DEFAULT_REGION"] = "antarctica"
```

2.4 IAM-like Access Control

Moto also has the ability to authenticate and authorize actions, just like it's done by IAM in AWS. This functionality can be enabled by either setting the *INITIAL_NO_AUTH_ACTION_COUNT* environment variable or using the *set_initial_no_auth_action_count* decorator. Note that the current implementation is very basic, see [the source code](#) for more information.

2.4.1 INITIAL_NO_AUTH_ACTION_COUNT

If this environment variable is set, moto will skip performing any authentication as many times as the variable's value, and only starts authenticating requests afterwards. If it is not set, it defaults to infinity, thus moto will never perform any authentication at all.

2.4.2 set_initial_no_auth_action_count

This is a decorator that works similarly to the environment variable, but the settings are only valid in the function's scope. When the function returns, everything is restored.

```
@set_initial_no_auth_action_count(4)
@mock_ec2
def test_describe_instances_allowed():
    policy_document = {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "ec2:Describe*",
                "Resource": "*"
            }
        ]
    }
    access_key = ...
    # create access key for an IAM user/assumed role that has the policy above.
```

(continues on next page)

(continued from previous page)

```
# this part should call __exactly__ 4 AWS actions, so that authentication and  
↪authorization starts exactly after this  
  
client = boto3.client('ec2', region_name='us-east-1',  
                      aws_access_key_id=access_key['AccessKeyId'],  
                      aws_secret_access_key=access_key['SecretAccessKey'])  
  
# if the IAM principal whose access key is used, does not have the permission to  
↪describe instances, this will fail  
instances = client.describe_instances()['Reservations'][0]['Instances']  
assert len(instances) == 0
```

See the [related test suite](#) for more examples.

2.5 AWS Config Support

An experimental feature for AWS Config has been developed to provide AWS Config capabilities in your unit tests. This feature is experimental as there are many services that are not yet supported and will require the community to add them in over time. This page details how the feature works and how you can use it.

2.5.1 What is this and why would I use this?

AWS Config is an AWS service that describes your AWS resource types and can track their changes over time. At this time, moto does not have support for handling the configuration history changes, but it does have a few methods mocked out that can be immensely useful for unit testing.

If you are developing automation that needs to pull against AWS Config, then this will help you write tests that can simulate your code in production.

2.5.2 How does this work?

The AWS Config capabilities in moto work by examining the state of resources that are created within moto, and then returning that data in the way that AWS Config would return it (sans history). This will work by querying all of the moto backends (regions) for a given resource type.

However, this will only work on resource types that have this enabled.

Current enabled resource types

1. S3 (all)
2. IAM (Role, Policy)

2.5.3 Developer Guide

There are several pieces to this for adding new capabilities to moto:

1. Listing resources
2. Describing resources

For both, there are a number of pre-requisites:

Base Components

In the `moto/core/models.py` file is a class named `ConfigQueryModel`. This is a base class that keeps track of all the resource type backends.

At a minimum, resource types that have this enabled will have:

1. A `config.py` file that will import the resource type backends (from the `__init__.py`)
2. In the resource's `config.py`, an implementation of the `ConfigQueryModel` class with logic unique to the resource type
3. An instantiation of the `ConfigQueryModel`
4. In the `moto/config/models.py` file, import the `ConfigQueryModel` instantiation, and update `RESOURCE_MAP` to have a mapping of the AWS Config resource type to the instantiation on the previous step (just imported).

An example of the above is implemented for S3. You can see that by looking at:

1. `moto/s3/config.py`
2. `moto/config/models.py`

Testing

For each resource type, you will need to test write tests for a few separate areas:

- Test the backend queries to ensure discovered resources come back (ie for `IAM::Policy`, write `tests.tests_iam.test_policy_list_config_discovered_resources`). For writing these tests, you must not make use of `boto` to create resources. You will need to use the backend model methods to provision the resources. This is to make tests compatible with the moto server. You must make tests for the resource type to test listing and object fetching.
- Test the config dict for all scenarios (ie for `IAM::Policy`, write `tests.tests_iam.test_policy_config_dict`). For writing this test, you'll need to create resources in the same way as the first test (without using `boto`), in every meaningful configuration that would produce a different config dict. Then, query the backend and ensure each of the dicts are as you expect.
- Test that everything works end to end with the `boto` clients. (ie for `IAM::Policy`, write `tests.tests_iam.test_policy_config_client`). The main two items to test will be the `boto.client('config').list_discovered_resources()`, `boto.client('config').list_aggregate_discovered_resources()`, `moto.client('config').batch_get_resource_config()`, and `moto.client('config').batch_aggregate_get_resource_config()`. This test doesn't have to be super thorough, but it basically tests that the front end and backend logic all works together and returns correct resources. Beware the aggregate methods all have capital first letters (ie `Limit`), while non-aggregate methods have lowercase first letters (ie `limit`)

Listing

S3 is currently the model implementation, but it also odd in that S3 is a global resource type with regional resource residency.

But for most resource types the following is true:

1. There are regional backends with their own sets of data
2. Config aggregation can pull data from any backend region – we assume that everything lives in the same account

Implementing the listing capability will be different for each resource type. At a minimum, you will need to return a *List of Dict* that look like this:

```
[
  {
    'type': 'AWS::The AWS Config data type',
    'name': 'The name of the resource',
    'id': 'The ID of the resource',
    'region': 'The region of the resource -- if global, then you may want to have the_
↳calling logic pass in the
                    aggregator region in for the resource region -- or just us-east-1 :P'
  }
  , ...
]
```

It's recommended to read the comment for the *ConfigQueryModel*'s *list_config_service_resources* function in [base class here](moto/core/models.py).

^^ The AWS Config code will see this and format it correct for both aggregated and non-aggregated calls.

General implementation tips

The aggregation and non-aggregation querying can and should just use the same overall logic. The differences are:

1. Non-aggregated listing will specify the region-name of the resource backend *backend_region* 1. Aggregated listing will need to be able to list resource types across ALL backends and filter optionally by passing in *resource_region*.

An example of a working implementation of this is S3.

Pagination should generally be able to pull out the resource across any region so should be sharded by *region-item-name* – not done for S3 because S3 has a globally unique name space.

Describing Resources

Fetching a resource's configuration has some similarities to listing resources, but it requires more work (to implement). Due to the various ways that a resource can be configured, some work will need to be done to ensure that the Config dict returned is correct.

For most resource types the following is true:

1. There are regional backends with their own sets of data 1. Config aggregation can pull data from any backend region – we assume that everything lives in the same account

The current implementation is for S3. S3 is very complex and depending on how the bucket is configured will depend on what Config will return for it.

When implementing resource config fetching, you will need to return at a minimum *None* if the resource is not found, or a *dict* that looks like what AWS Config would return.

It's recommended to read the comment for the `ConfigQueryModel`'s `get_config_resource` function in the base class.

2.6 Multi-Account support

By default, Moto processes all requests in a default account: `12345678910`. The exact credentials provided are usually ignored to make the process of mocking requests as hassle-free as possible.

If you want to mock resources in multiple accounts, or you want to change the default account ID, there are multiple ways to achieve this.

2.6.1 Configure the default account

It is possible to configure the default account ID that will be used for all incoming requests, by setting the environment variable `MOTO_ACCOUNT_ID`.

Here is an example of what this looks like in practice:

```
# Create a bucket in the default account
client = boto3.client("s3", region_name="us-east-1")
client.create_bucket(Bucket="bucket-default-account")

# Configure another account - all subsequent requests will use this account ID
os.environ["MOTO_ACCOUNT_ID"] = "111111111111"
client.create_bucket(Bucket="bucket-in-account-2")

assert [b["Name"] for b in client2.list_buckets()["Buckets"]] == ["bucket-in-account-2"]

# Now revert to the default account, by removing the environment variable
del os.environ["MOTO_ACCOUNT_ID"]
assert [b["Name"] for b in client2.list_buckets()["Buckets"]] == ["bucket-default-account
↪"]
```

2.6.2 Configure the account ID using a request header

If you are using Moto in ServerMode you can add a custom header to a request, to specify which account should be used.

Note: Moto will only look at the request-header if the environment variable is not set.

As an example, this is how you would create an S3-bucket in another account:

```
headers = {"x-moto-account-id": "333344445555"}
requests.put("http://bucket.localhost:5000/", headers=headers)

# This will return a list of all buckets in account 333344445555
requests.get("http://localhost:5000", headers=headers)

# This will return an empty list, as there are no buckets in the default account
requests.get("http://localhost:5000")
```

2.6.3 Configure an account using STS

The `STS.assume_role()`-feature is useful if you want to temporarily use a different set of access credentials. Passing in a role that belongs to a different account will return a set of credentials that give access to that account.

Note: To avoid any chicken-and-egg problems trying to create roles in non-existing accounts, these Roles do not need to exist. Moto will only extract the account ID from the role, and create access credentials for that account.

Note: Moto will only look at the access credentials if the environment variable and request header is not set.

Let's look at some examples.

```
# Create a bucket using the default access credentials
client1 = boto3.client("s3", region_name="us-east-1")
client1.create_bucket(Bucket="foobar")

# Assume a role in our account
# Note that this Role does not need to exist
default_account = "123456789012"
sts = boto3.client("sts")
response = sts.assume_role(
    RoleArn=f"arn:aws:iam::{default_account}:role/my-role",
    RoleSessionName="test-session-name",
    ExternalId="test-external-id",
)

# These access credentials give access to the default account
client2 = boto3.client(
    "s3",
    aws_access_key_id=response["Credentials"]["AccessKeyId"],
    aws_secret_access_key=response["Credentials"]["SecretAccessKey"],
    aws_session_token=response["Credentials"]["SessionToken"],
    region_name="us-east-1",
)
client2.list_buckets()["Buckets"].should.have.length_of(1)
```

Because we assumed a role within the same account, we can see the bucket that we've just created.

Things get interesting when assuming a role within a different account.

```
# Create a bucket with default access credentials
client1 = boto3.client("s3", region_name="us-east-1")
client1.create_bucket(Bucket="foobar")

# Assume a role in a different account
# Note that the Role does not need to exist
sts = boto3.client("sts")
response = sts.assume_role(
    RoleArn="arn:aws:iam::111111111111:role/role-in-another-account",
    RoleSessionName="test-session-name",
    ExternalId="test-external-id",
```

(continues on next page)

(continued from previous page)

```

)
# Retrieve all buckets in this new account - this will be completely empty
client2 = boto3.client(
    "s3",
    aws_access_key_id=response["Credentials"]["AccessKeyId"],
    aws_secret_access_key=response["Credentials"]["SecretAccessKey"],
    aws_session_token=response["Credentials"]["SessionToken"],
    region_name="us-east-1",
)
client2.list_buckets()["Buckets"].should.have.length_of(0)

```

Because we've assumed a role in a different account, no buckets were found. The *foobar*-bucket only exists in the default account, not in *IIIIIIIIIIII*.

2.7 Configuration Options

Moto has a variety of ways to configure the mock behaviour.

2.7.1 Environment Variables

The following is a non-exhaustive list of the environment variables that can be used to configure Moto.

Key	Value	De- fault	Explanation
TEST_SERVER_MODE	bool	False	Useful when you want to run decorated tests against an existing Moto-Server. All boto3-clients/resources created within the test will point to <i>http://localhost:5000</i> .
INITIAL_NO_AUTH_ACTION_COUNT	int	0	See <i>IAM-like Access Control</i> .
DEFAULT_CONTAINER_REGISTRY	str	docker	Registry that contains the Docker containers. Used by AWSLambda and Batch.
MOTO_ALLOW_NONEXISTENT_REGION	bool	True	
MOTO_S3_CUSTOM_ENDPOINTS	list		See <i>s3</i> .

2.7.2 Recorder

The Moto Recorder is used to log all incoming requests, which can be replayed at a later date. This is useful if you need to setup an initial state, and ensure that this is the same across developers/environments.

Usage

Usage in decorator mode:

```
from moto.moto_api import recorder

# Start the recorder
recorder.start_recording()
# Make some requests using boto3

# When you're ready..
recorder.stop_recording()
log = recorder.download_recording()

# Later on, upload this log to another system
recorder.upload_recording(log)
# And replay the contents
recorder.replay_recording()

# While the recorder is active, new requests will be appended to the existing log
# Reset the current log if you want to start with an empty slate
recorder.reset_recording()
```

Usage in ServerMode:

```
# Start the recorder
requests.post("http://localhost:5000/moto-api/recorder/start-recording")
# Make some requests

# When you're ready..
requests.post("http://localhost:5000/moto-api/recorder/stop-recording")
log = requests.get("http://localhost:5000/moto-api/recorder/download-recording").content

# Later on, upload this log to another system
requests.post("http://localhost:5000/moto-api/recorder/upload-recording", data=log)
# and replay the contents
requests.post("http://localhost:5000/moto-api/recorder/replay-recording")

# While the recorder is active, new requests will be appended to the existing log
# Reset the current log if you want to start with an empty slate
requests.post("http://localhost:5000/moto-api/recorder/reset-recording")
```

Note that this feature records and replays the incoming HTTP request. Randomized data created by Moto, such as resource ID's, will not be stored as part of the log.

Recorder Configuration

The requests are stored in a file called *moto_recording*, in the directory that Python is run from. You can configure this location using the following environment variable: *MOTO_RECORDER_FILEPATH=/whatever/path/you/want*

The recorder is disabled by default. If you want to enable it, use the following environment variable: *MOTO_ENABLE_RECORDING=True*

Deterministic Identifiers

Moto creates random identifiers for most resources, just like AWS. The Recorder will recreate the same resources every time, but with different identifiers.

It is possible to seed Moto and ensure that the ‘random’ identifiers are always the same for subsequent requests.

Example invocation:

```
# Ensure the provided parameter `a` is an integer
requests.post("http://motoapi.amazonaws.com/moto-api/seed?a=42")

# To try this out, generate a EC2 instance
client = boto3.client("ec2", region_name="us-east-1")
resp = client.run_instances(ImageId="ami-12c6146b", MinCount=1, MaxCount=1)

# The resulting InstanceId will always be the same
instance_id = resp["Instances"][0]["InstanceId"]
assert instance_id == "i-d1026706d7e805da8"
```

To seed Moto in ServerMode:

```
requests.post(f"http://localhost:5000/moto-api/seed?a=42")
```

Because the seeding API is only exposed as a request, it will be recorded just like any other request. Seed Moto at the beginning of a recording to ensure the resulting state will always be the same:

```
requests.post("http://localhost:5000/moto-api/recorder/start-recording")
requests.post("http://localhost:5000/moto-api/seed?a=42")

client = boto3.client("ec2", region_name="us-east-1")
resp = client.run_instances(ImageId="ami-12c6146b", MinCount=1, MaxCount=1)

requests.post("http://localhost:5000/moto-api/recorder/stop-recording")
```

2.7.3 Prettify responses

This option allows to prettify responses from moto. Pretty responses are more readable (eg. for debugging purposes). It also makes moto better in mocking AWS as AWS returns prettified responses.

Ugly output:

```
<DeleteLaunchTemplatesResponse xmlns="http://ec2.amazonaws.com/doc/2016-11-15/">
  ↪<requestId>178936da-50ad-4d58-8871-22d9979e8658example</requestId><launchTemplate>
  ↪<defaultVersionNumber>1</defaultVersionNumber><launchTemplateId>lt-d920e32b0cccd6adb</
  ↪launchTemplateId><launchTemplateName>example-name</launchTemplateName></launchTemplate>
  ↪</DeleteLaunchTemplatesResponse>
```

(continues on next page)

Prettified output:

```
<DeleteLaunchTemplatesResponse xmlns="http://ec2.amazonaws.com/doc/2016-11-15/">
  <requestId>178936da-50ad-4d58-8871-22d9979e8658example</requestId>
  <launchTemplate>
    <defaultVersionNumber>1</defaultVersionNumber>
    <launchTemplateId>lt-d920e32b0cccd6adb</launchTemplateId>
    <launchTemplateName>example-name</launchTemplateName>
  </launchTemplate>
</DeleteLaunchTemplatesResponse>
```

Enabling Pretty responses

As changing responses can interfere with some external tools, it is disabled by default. If you want to enable it, use environment variable: `MOTO_PRETTIFY_RESPONSES=True`

2.7.4 State Transitions

When developing against AWS, many API calls are asynchronous. Many resources will take some time to complete, and you'll need to write business logic to ensure the application can deal with all possible states. What is the desired behaviour when the status is *initializing*? What should happen when the status is finally *ready*? What should happen when the resource is still not *ready* after an hour?

Let's look at an example. Say you want to create a DAX cluster, and wait until it's available - or throw an error if this takes too long.

```
def create_and_wait_for_cluster(name):
    client.create_cluster(ClusterName=name, ...)

    cluster_status = get_cluster_status(name)
    while cluster_status != "available":
        sleep()

        if five_minutes_have_passed():
            error()

    cluster_status = get_cluster_status(name)
```

Because Moto handles everything in-memory, and no actual servers are created, there is no need to wait until the cluster is ready - it could be ready immediately. Not having to wait for a resource to be ready is of course the major benefit of using Moto, but it also means that the entire example above is impossible to test.

Moto exposes an API that can artificially delay these state transitions, allowing you to let Moto resemble the asynchronous nature of AWS as closely as you need.

Sticking with the example above, you may want to test what happens if the cluster takes 5 seconds to create:

```
from moto.moto_api import state_manager

state_manager.set_transition(model_name="dax::cluster", transition={"progression": "time
↪", "seconds": 5})
```

(continues on next page)

(continued from previous page)

```
create_and_wait_for_cluster("my_new_cluster")
```

In order to test what happens in the event of a timeout, we can order the cluster to only be ready after 10 minutes:

```
from moto.moto_api import state_manager

state_manager.set_transition(model_name="dax::cluster", transition={"progression": "time
↪", "seconds": 600})

try:
    create_and_wait_for_cluster("my_new_cluster")
except:
    verify_the_correct_error_was_thrown()
```

In other tests, you may simply want the cluster to be ready as quickly as possible:

```
from moto.moto_api import state_manager

state_manager.set_transition(model_name="dax::cluster", transition={"progression":
↪ "immediate"})
```

So far we've seen two possible transitions:

- The state progresses immediately
- The state progresses after x seconds

There is a third possibility, where the state progresses after calling *describe_object* a specific number of times. This can be useful if you want to verify that the state does change, but you don't want your unit test to take too long.

Note: We will use the *boto3.client(..).describe_object* method as an example throughout this page. This should be seen as a agnostic version of service-specific methods to verify the status of a resource, such as *boto.client("dax").describe_clusters()* or *boto.client("support").describe_cases()*.

Changing the state after a certain number of invocations can be done like this:

```
state_manager.set_transition(model_name="dax::cluster", transition={"progression":
↪ "manual", "times": 3})
```

The transition is called *manual* because it requires you to manually invoke the *describe_object*-method before the status is progressed. To show how this would work in practice, let's look at an example test:

```
client.create_cluster(ClusterName=name, ...)
# The first time we retrieve the status
status = client.describe_clusters(ClusterNames=[name])["Clusters"][0]["Status"]
assert status == "creating"
# Second time we retrieve the status
status = client.describe_clusters(ClusterNames=[name])["Clusters"][0]["Status"]
assert status == "creating"
# This is the third time that we're retrieving the status - this time it will advance to
↪ the next status
status = client.describe_clusters(ClusterNames=[name])["Clusters"][0]["Status"]
assert status == "available"
```

This should be done cleanly in a while-loop of-course, similar to the `create_and_wait_for_cluster` defined above - but this is a good way to showcase the behaviour.

Registered models

A list of all supported models can be found here.

Older versions of Moto may not support all models that are listed here. To see a list of supported models for your Moto-version, call the `get_registered_models`-method:

```
with mock_all():
    print(state_manager.get_registered_models())
```

Note the `mock_all`-decorator! Models are registered when the mock for that resource is started. If you call this method outside of a mock, you may see an empty list.

If you'd like to see state transition support for a resource that's not yet supported, feel free to open an issue or PR.

State Transitions in ServerMode

Configuration state transitions can be done in ServerMode as well, by making a HTTP request to the MotoAPI. This is an example request for `dax::cluster` to wait 5 seconds before the cluster becomes ready:

```
post_body = dict(model_name="dax::cluster", transition={"progression": "time", "seconds": 5})
resp = requests.post("http://localhost:5000/moto-api/state-manager/set-transition",
                    data=json.dumps(post_body))
```

An example request to see the currently configured transition for a specific model:

```
requests.get("http://localhost:5000/moto-api/state-manager/get-transition?model_name=dax::cluster")
```

We will not list all configuration options here again, but all models and transitions types (as specified above) follow the same format.

Reset

It is possible to reset the state manager, and undo any custom transitions that were set. Using Python:

```
from moto.moto_api import state_manager

state_manager.unset_transition(model_name="dax::cluster")
```

Or if you're using Moto in ServerMode:

```
post_body = dict(model_name="dax::cluster")
resp = requests.post("http://localhost:5000/moto-api/state-manager/unset-transition",
                    data=json.dumps(post_body))
```

2.7.5 Supported Models for State Transitions

Service: Batch

Model: *batch::job* Available States:

“SUBMITTED” → “PENDING” → “RUNNABLE” → “STARTING” → “RUNNING” “RUNNING” → SUCCEEDED|FAILED

Transition type: *immediate* Advancement:

When a user calls *submit_job*, Moto will go through a few steps to prepare the job, and when ready, execute that job in a Docker container. There are some steps to go through while the status is *SUBMITTED*, there are some steps to follow when the status is *PENDING*, etcetera.

Moto will try to advance the status itself - the moment this succeeds, the next step is executed. As the default transition is *immediate*, the status will advance immediately, and these steps will be executed as quickly as possible. This ensures that the job will be executed as quickly as possible.

Delaying the execution can be done as usual, by forcing Moto to wait x seconds before transitioning to the next stage. This can be useful if you need to ‘catch’ a job in a specific stage.

Service: Cloudfront

Model: *cloudfront::distribution* Available States:

“InProgress” → “Deployed”

Transition type: Manual - describe the resource 1 time before the state advances Advancement:

Call `boto3.client("cloudfront").get_distribution(..)` to advance a single distribution, or `boto3.client("cloudfront").list_distributions(..)` to advance all distributions.

Service: DAX

Model: *dax::cluster* Available States:

“creating” → “available” “deleting” → “deleted”

Transition type: Manual - describe the resource 4 times before the state advances Advancement:

Call `boto3.client("dax").describe_clusters(..)`.

Service: Glue

Model: *glue::job_run* Available States:

“STARTING” → “RUNNING” → “SUCCEEDED”

Transition type: *immediate* Advancement:

Call `boto3.client("glue").get_job_run(..)`

Service: S3 (Glacier Restoration)

Model: *s3::keyrestore* Available States:

None → “IN_PROGRESS” → “RESTORED”

Transition type: Immediate - transitions immediately

Service: Support

Model: *support::case* Available states:

“opened” → “pending-customer-action” → “reopened” → “resolved” → “unassigned” → “work-in-progress” → “opened”

Transition type: Manual - describe the resource 1 time before the state advances Advancement:

Call `boto3.client("support").describe_cases(..)`

Service: Transcribe

Model: *transcribe::vocabulary* Available states:

None → “PENDING → “READY”

Transition type: Manual - describe the resource 1 time before the state advances Advancement:

Call `boto3.client("transcribe").get_vocabulary(..)`

Model: *transcribe::medicalvocabulary* Available states:

None → “PENDING → “READY”

Transition type: Manual - describe the resource 1 time before the state advances Advancement:

Call `boto3.client("transcribe").get_medical_vocabulary(..)`

Model: *transcribe::transcriptionjob* Available states:

None → “QUEUED” → “IN_PROGRESS” → “COMPLETED”

Transition type: Manual - describe the resource 1 time before the state advances Advancement:

Call `boto3.client("transcribe").get_transcription_job(..)`

Model: *transcribe::medicaltranscriptionjob* Available states:

None → “QUEUED” → “IN_PROGRESS” → “COMPLETED”

Transition type: Manual - describe the resource 1 time before the state advances Advancement:

Call `boto3.client("transcribe").get_medical_transcription_job(..)`

2.8 Implemented Services

Please see a list of all currently supported services. Each service will have a list of the endpoints that are implemented. Each service will also have an example on how to mock an individual service.

Note that you can mock multiple services at the same time:

```
@mock_s3
@mock_sqs
def test_both_s3_and_sqs():
    ...
```

```
@mock_all()
def test_all_supported_services_at_the_same_time():
    ...
```

2.8.1 acm

Example usage

```
@mock_acm
def test_acm_behaviour():
    boto3.client("acm")
    ...
```

Implemented features for this service

- [X] add_tags_to_certificate
- [X] delete_certificate
- [] describe_certificate
- [X] export_certificate
- [] get_account_configuration
- [X] get_certificate
- [] import_certificate
- [] list_certificates
- [] list_tags_for_certificate
- [] put_account_configuration
- [X] remove_tags_from_certificate
- [] renew_certificate
- [X] request_certificate

The parameter DomainValidationOptions has not yet been implemented

- [] resend_validation_email
- [] update_certificate_options

2.8.2 acm-pca

`class moto.acmpca.models.ACMPCABackend(region_name: str, account_id: str)`
Implementation of ACMPCA APIs.

Example usage

```
@mock_acmpca
def test_acmpca_behaviour:
    boto3.client("acm-pca")
    ...
```

Implemented features for this service

- [X] create_certificate_authority

The following parameters are not yet implemented: IdempotencyToken, KeyStorageSecurityStandard, UsageMode

- [] create_certificate_authority_audit_report
- [] create_permission
- [X] delete_certificate_authority
- [] delete_permission
- [] delete_policy
- [X] describe_certificate_authority
- [] describe_certificate_authority_audit_report
- [X] get_certificate

The CertificateChain will always return None for now

- [X] get_certificate_authority_certificate
- [X] get_certificate_authority_csr
- [] get_policy
- [X] import_certificate_authority_certificate
- [X] issue_certificate

The following parameters are not yet implemented: ApiPassthrough, SigningAlgorithm, TemplateArn, Validity, ValidityNotBefore, IdempotencyToken Some fields of the resulting certificate will have default values, instead of using the CSR

- [] list_certificate_authorities
- [] list_permissions
- [X] list_tags

Pagination is not yet implemented

- [] put_policy
- [] restore_certificate_authority
- [X] revoke_certificate

This is currently a NO-OP

- [X] tag_certificate_authority
- [X] untag_certificate_authority
- [X] update_certificate_authority

2.8.3 amp

class moto.amp.models.PrometheusServiceBackend(*region_name: str, account_id: str*)
Implementation of PrometheusService APIs.

Example usage

```
@mock_aws
def test_amp_behaviour():
    boto3.client("amp")
    ...
```

Implemented features for this service

- [] create_alert_manager_definition
- [X] create_logging_configuration
- [X] create_rule_groups_namespace
 - The ClientToken-parameter is not yet implemented
- [X] create_workspace
 - The ClientToken-parameter is not yet implemented
- [] delete_alert_manager_definition
- [X] delete_logging_configuration
- [X] delete_rule_groups_namespace
 - The ClientToken-parameter is not yet implemented
- [X] delete_workspace
 - The ClientToken-parameter is not yet implemented
- [] describe_alert_manager_definition
- [X] describe_logging_configuration
- [X] describe_rule_groups_namespace
- [X] describe_workspace
- [X] list_rule_groups_namespaces
- [X] list_tags_for_resource
- [X] list_workspaces
- [] put_alert_manager_definition
- [X] put_rule_groups_namespace
 - The ClientToken-parameter is not yet implemented
- [X] tag_resource
- [X] untag_resource

- [X] update_logging_configuration
- [X] update_workspace_alias

The ClientToken-parameter is not yet implemented

2.8.4 apigateway

class moto.apigateway.models.**APIGatewayBackend**(region_name: str, account_id: str)
API Gateway mock.

The public URLs of an API integration are mocked as well, i.e. the following would be supported in Moto:

```
client.put_integration(  
    restApiId=api_id,  
    ...,  
    uri="http://httpbin.org/robots.txt",  
    integrationHttpMethod="GET"  
)  
deploy_url = f"https://{api_id}.execute-api.us-east-1.amazonaws.com/dev"  
requests.get(deploy_url).content.should.equal(b"a fake response")
```

Limitations:

- Integrations of type HTTP are supported
- Integrations of type AWS with service DynamoDB are supported
- Other types (AWS_PROXY, MOCK, etc) are ignored
- Other services are not yet supported
- The BasePath of an API is ignored
- TemplateMapping is not yet supported for requests/responses
- This only works when using the decorators, not in ServerMode

Example usage

```
@mock_apigateway  
def test_apigateway_behaviour:  
    boto3.client("apigateway")  
    ...
```

Implemented features for this service

- [X] create_api_key
- [X] create_authorizer
- [X] create_base_path_mapping
- [X] create_deployment
- [] create_documentation_part
- [] create_documentation_version
- [X] create_domain_name
- [X] create_model

- create_request_validator
- create_resource
- create_rest_api
- create_stage
- create_usage_plan
- create_usage_plan_key
- create_vpc_link
- delete_api_key
- delete_authorizer
- delete_base_path_mapping
- delete_client_certificate
- delete_deployment
- delete_documentation_part
- delete_documentation_version
- delete_domain_name
- delete_gateway_response
- delete_integration
- delete_integration_response
- delete_method
- delete_method_response
- delete_model
- delete_request_validator
- delete_resource
- delete_rest_api
- delete_stage
- delete_usage_plan
- delete_usage_plan_key
- delete_vpc_link
- flush_stage_authorizers_cache
- flush_stage_cache
- generate_client_certificate
- get_account
- get_api_key
- get_api_keys
- get_authorizer
- get_authorizers

- [X] `get_base_path_mapping`
- [X] `get_base_path_mappings`
- [] `get_client_certificate`
- [] `get_client_certificates`
- [X] `get_deployment`
- [X] `get_deployments`
- [] `get_documentation_part`
- [] `get_documentation_parts`
- [] `get_documentation_version`
- [] `get_documentation_versions`
- [X] `get_domain_name`
- [X] `get_domain_names`
- [] `get_export`
- [X] `get_gateway_response`
- [X] `get_gateway_responses`
 - `Pagination is not yet implemented`
- [X] `get_integration`
- [X] `get_integration_response`
- [X] `get_method`
- [X] `get_method_response`
- [X] `get_model`
- [] `get_model_template`
- [X] `get_models`
- [X] `get_request_validator`
- [X] `get_request_validators`
- [X] `get_resource`
- [X] `get_resources`
- [X] `get_rest_api`
- [] `get_rest_apis`
- [] `get_sdk`
- [] `get_sdk_type`
- [] `get_sdk_types`
- [X] `get_stage`
- [X] `get_stages`
- [] `get_tags`
- [] `get_usage`

- [X] get_usage_plan
- [X] get_usage_plan_key
- [X] get_usage_plan_keys
- [X] get_usage_plans
- [X] get_vpc_link
- [X] get_vpc_links

 Pagination has not yet been implemented

- [] import_api_keys
- [] import_documentation_parts
- [X] import_rest_api

 Only a subset of the OpenAPI spec 3.x is currently implemented.

- [X] put_gateway_response
- [X] put_integration
- [X] put_integration_response
- [X] put_method
- [X] put_method_response
- [X] put_rest_api

 Only a subset of the OpenAPI spec 3.x is currently implemented.

- [] tag_resource
- [] test_invoke_authorizer
- [] test_invoke_method
- [] untag_resource
- [] update_account
- [X] update_api_key
- [X] update_authorizer
- [X] update_base_path_mapping
- [] update_client_certificate
- [] update_deployment
- [] update_documentation_part
- [] update_documentation_version
- [] update_domain_name
- [] update_gateway_response
- [] update_integration
- [] update_integration_response
- [] update_method
- [] update_method_response

- [] update_model
- [X] update_request_validator
- [] update_resource
- [X] update_rest_api
- [X] update_stage
- [] update_usage
- [X] update_usage_plan
- [] update_vpc_link

2.8.5 apigatewayv2

class moto.apigatewayv2.models.**ApiGatewayV2Backend**(*region_name: str, account_id: str*)
Implementation of ApiGatewayV2 APIs.

Example usage

```
@mock_apigatewayv2
def test_apigatewayv2_behaviour:
    boto3.client("apigatewayv2")
    ...
```

Implemented features for this service

- [X] create_api
The following parameters are not yet implemented: CredentialsArn, RouteKey, Tags, Target
- [X] create_api_mapping
- [X] create_authorizer
- [] create_deployment
- [X] create_domain_name
- [X] create_integration
- [X] create_integration_response
- [X] create_model
- [X] create_route
- [X] create_route_response
The following parameters are not yet implemented: ResponseModels, ResponseParameters
- [] create_stage
- [X] create_vpc_link
- [] delete_access_log_settings
- [X] delete_api
- [X] delete_api_mapping
- [X] delete_authorizer
- [X] delete_cors_configuration

- [] delete_deployment
- [X] delete_domain_name
- [X] delete_integration
- [X] delete_integration_response
- [X] delete_model
- [X] delete_route
- [X] delete_route_request_parameter
- [X] delete_route_response
- [] delete_route_settings
- [] delete_stage
- [X] delete_vpc_link
- [] export_api
- [X] get_api
- [X] get_api_mapping
- [X] get_api_mappings
- [X] get_apis
 - Pagination is not yet implemented
- [X] get_authorizer
- [] get_authorizers
- [] get_deployment
- [] get_deployments
- [X] get_domain_name
- [X] get_domain_names
 - Pagination is not yet implemented
- [X] get_integration
- [X] get_integration_response
- [X] get_integration_responses
- [X] get_integrations
 - Pagination is not yet implemented
- [X] get_model
- [] get_model_template
- [] get_models
- [X] get_route
- [X] get_route_response
- [] get_route_responses
- [X] get_routes

Pagination is not yet implemented

- [] `get_stage`
- [] `get_stages`
- [X] `get_tags`
- [X] `get_vpc_link`
- [X] `get_vpc_links`
- [] `import_api`
- [X] `reimport_api`

Only YAML is supported at the moment. Full OpenAPI-support is not guaranteed. Only limited validation is implemented

- [] `reset_authorizers_cache`
- [X] `tag_resource`
- [X] `untag_resource`
- [X] `update_api`

The following parameters have not yet been implemented: `CredentialsArn`, `RouteKey`, `Target`

- [] `update_api_mapping`
- [X] `update_authorizer`
- [] `update_deployment`
- [] `update_domain_name`
- [X] `update_integration`
- [X] `update_integration_response`
- [X] `update_model`
- [X] `update_route`
- [] `update_route_response`
- [] `update_stage`
- [X] `update_vpc_link`

2.8.6 application-autoscaling

Example usage

```
@mock_applicationautoscaling
def test_applicationautoscaling_behaviour:
    boto3.client("application-autoscaling")
    ...
```

Implemented features for this service

- [X] `delete_scaling_policy`
- [X] `delete_scheduled_action`
- [X] `deregister_scalable_target` Registers or updates a scalable target.

- [X] describe_scalable_targets Describe scalable targets.
- [] describe_scaling_activities
- [X] describe_scaling_policies
- [X] describe_scheduled_actions
 - Pagination is not yet implemented
- [X] put_scaling_policy
- [X] put_scheduled_action
- [X] register_scalable_target Registers or updates a scalable target.

2.8.7 appsync

class moto.appsync.models.**AppSyncBackend**(*region_name: str, account_id: str*)
Implementation of AppSync APIs.

Example usage

```
@mock_appsync
def test_appsync_behaviour():
    boto3.client("appsync")
    ...
```

Implemented features for this service

- [] associate_api
- [] create_api_cache
- [X] create_api_key
- [] create_data_source
- [] create_domain_name
- [] create_function
- [X] create_graphql_api
- [] create_resolver
- [] create_type
- [] delete_api_cache
- [X] delete_api_key
- [] delete_data_source
- [] delete_domain_name
- [] delete_function
- [X] delete_graphql_api
- [] delete_resolver
- [] delete_type
- [] disassociate_api
- [] evaluate_code

- [] evaluate_mapping_template
- [] flush_api_cache
- [] get_api_association
- [] get_api_cache
- [] get_data_source
- [] get_domain_name
- [] get_function
- [X] get_graphql_api
- [] get_introspection_schema
- [] get_resolver
- [X] get_schema_creation_status
- [X] get_type
- [X] list_api_keys

 Pagination or the maxResults-parameter have not yet been implemented.

- [] list_data_sources
- [] list_domain_names
- [] list_functions
- [X] list_graphql_apis

 Pagination or the maxResults-parameter have not yet been implemented.

- [] list_resolvers
- [] list_resolvers_by_function
- [X] list_tags_for_resource
- [] list_types
- [X] start_schema_creation
- [X] tag_resource
- [X] untag_resource
- [] update_api_cache
- [X] update_api_key
- [] update_data_source
- [] update_domain_name
- [] update_function
- [X] update_graphql_api
- [] update_resolver
- [] update_type

2.8.8 athena

Example usage

```
@mock_athena
def test_athena_behaviour:
    boto3.client("athena")
    ...
```

Implemented features for this service

- [] batch_get_named_query
- [] batch_get_prepared_statement
- [] batch_get_query_execution
- [X] create_data_catalog
- [X] create_named_query
- [] create_notebook
- [] create_prepared_statement
- [] create_presigned_notebook_url
- [X] create_work_group
- [] delete_data_catalog
- [] delete_named_query
- [] delete_notebook
- [] delete_prepared_statement
- [] delete_work_group
- [] export_notebook
- [] get_calculation_execution
- [] get_calculation_execution_code
- [] get_calculation_execution_status
- [X] get_data_catalog
- [] get_database
- [X] get_named_query
- [] get_notebook_metadata
- [] get_prepared_statement
- [X] get_query_execution
- [X] get_query_results

Queries are not executed, so this call will always return 0 rows by default.

When using decorators, you can use the internal API to manually set results:

```
from moto.athena.models import athena_backends, QueryResults
from moto.core import DEFAULT_ACCOUNT_ID

backend = athena_backends[DEFAULT_ACCOUNT_ID]["us-east-1"]
rows = [{'Data': [{'VarCharValue': '..'}]}]
column_info = [{
    'CatalogName': 'string',
    'SchemaName': 'string',
    'TableName': 'string',
    'Name': 'string',
    'Label': 'string',
    'Type': 'string',
    'Precision': 123,
    'Scale': 123,
    'Nullable': 'NOT_NULL',
    'CaseSensitive': True
}]
results = QueryResults(rows=rows, column_info=column_info)
backend.query_results["test"] = results

result = client.get_query_results(QueryExecutionId="test")
```

- [] get_query_runtime_statistics
- [] get_session
- [] get_session_status
- [] get_table_metadata
- [X] get_work_group
- [] import_notebook
- [] list_application_dpu_sizes
- [] list_calculation_executions
- [X] list_data_catalogs
- [] list_databases
- [] list_engine_versions
- [] list_executors
- [] list_named_queries
- [] list_notebook_metadata
- [] list_notebook_sessions
- [] list_prepared_statements
- [X] list_query_executions
- [] list_sessions
- [] list_table_metadata
- [] list_tags_for_resource
- [X] list_work_groups

- [] start_calculation_execution
- [X] start_query_execution
- [] start_session
- [] stop_calculation_execution
- [X] stop_query_execution
- [] tag_resource
- [] terminate_session
- [] untag_resource
- [] update_data_catalog
- [] update_named_query
- [] update_notebook
- [] update_notebook_metadata
- [] update_prepared_statement
- [] update_work_group

2.8.9 autoscaling

Example usage

```
@mock_autoscaling
def test_autoscaling_behaviour:
    boto3.client("autoscaling")
    ...
```

Implemented features for this service

- [X] attach_instances
- [X] attach_load_balancer_target_groups
- [X] attach_load_balancers
- [] attach_traffic_sources
- [] batch_delete_scheduled_action
- [] batch_put_scheduled_update_group_action
- [] cancel_instance_refresh
- [] complete_lifecycle_action
- [X] create_auto_scaling_group
- [X] create_launch_configuration
- [X] create_or_update_tags
- [X] delete_auto_scaling_group
- [X] delete_launch_configuration
- [X] delete_lifecycle_hook

- [] delete_notification_configuration
- [X] delete_policy
- [X] delete_scheduled_action
- [X] delete_tags
- [] delete_warm_pool
- [] describe_account_limits
- [] describe_adjustment_types
- [X] describe_auto_scaling_groups
- [X] describe_auto_scaling_instances
- [] describe_auto_scaling_notification_types
- [] describe_instance_refreshes
- [X] describe_launch_configurations
- [] describe_lifecycle_hook_types
- [X] describe_lifecycle_hooks
- [X] describe_load_balancer_target_groups
- [X] describe_load_balancers
- [] describe_metric_collection_types
- [] describe_notification_configurations
- [X] describe_policies
- [] describe_scaling_activities
- [] describe_scaling_process_types
- [X] describe_scheduled_actions
- [X] describe_tags

 Pagination is not yet implemented. Only the *auto-scaling-group* and *propagate-at-launch* filters are implemented.

- [] describe_termination_policy_types
- [] describe_traffic_sources
- [] describe_warm_pool
- [X] detach_instances
- [X] detach_load_balancer_target_groups
- [X] detach_load_balancers
- [] detach_traffic_sources
- [] disable_metrics_collection
- [X] enable_metrics_collection
- [] enter_standby
- [X] execute_policy

- [] `exit_standby`
- [] `get_predictive_scaling_forecast`
- [] `put_lifecycle_hook`
- [] `put_notification_configuration`
- [X] `put_scaling_policy`
- [X] `put_scheduled_update_group_action`
- [] `put_warm_pool`
- [] `record_lifecycle_action_heartbeat`
- [X] `resume_processes`
- [X] `set_desired_capacity`
- [X] `set_instance_health`

The `ShouldRespectGracePeriod`-parameter is not yet implemented

- [X] `set_instance_protection`
- [] `start_instance_refresh`
- [X] `suspend_processes`
- [] `terminate_instance_in_auto_scaling_group`
- [X] `update_auto_scaling_group`

The parameter `DefaultCooldown`, `PlacementGroup`, `TerminationPolicies` are not yet implemented

2.8.10 batch

class `moto.batch.models.BatchBackend`(*region_name: str, account_id: str*)

Batch-jobs are executed inside a Docker-container. Everytime the `submit_job`-method is called, a new Docker container is started. A job is marked as 'Success' when the Docker-container exits without throwing an error.

Use `@mock_batch_simple` instead if you do not want to use a Docker-container. With this decorator, jobs are simply marked as 'Success' without trying to execute any commands/scripts.

Example usage

```
@mock_batch
def test_batch_behaviour:
    boto3.client("batch")
    ...
```

Implemented features for this service

- [X] `cancel_job`
- [X] `create_compute_environment`
- [X] `create_job_queue`
- [X] `create_scheduling_policy`
- [X] `delete_compute_environment`
- [X] `delete_job_queue`

- [X] delete_scheduling_policy
- [X] deregister_job_definition
- [X] describe_compute_environments
 - Pagination is not yet implemented
- [X] describe_job_definitions
 - Pagination is not yet implemented
- [X] describe_job_queues
 - Pagination is not yet implemented
- [X] describe_jobs
- [X] describe_scheduling_policies
- [X] list_jobs
 - Pagination is not yet implemented
- [X] list_scheduling_policies
 - Pagination is not yet implemented
- [X] list_tags_for_resource
- [X] register_job_definition
- [X] submit_job
 - Parameters RetryStrategy and Parameters are not yet implemented.
- [X] tag_resource
- [X] terminate_job
- [X] untag_resource
- [X] update_compute_environment
- [X] update_job_queue
- [X] update_scheduling_policy

2.8.11 budgets

class moto.budgets.models.**BudgetsBackend**(*region_name: str, account_id: str*)
Implementation of Budgets APIs.

Example usage

```
@mock_budgets
def test_budgets_behaviour:
    boto3.client("budgets")
    ...
```

Implemented features for this service

- [X] create_budget
- [] create_budget_action
- [X] create_notification

- [] create_subscriber
- [X] delete_budget
- [] delete_budget_action
- [X] delete_notification
- [] delete_subscriber
- [X] describe_budget
- [] describe_budget_action
- [] describe_budget_action_histories
- [] describe_budget_actions_for_account
- [] describe_budget_actions_for_budget
- [] describe_budget_notifications_for_account
- [] describe_budget_performance_history
- [X] describe_budgets
 - [] Pagination is not yet implemented
- [X] describe_notifications_for_budget
 - [] Pagination has not yet been implemented
- [] describe_subscribers_for_notification
- [] execute_budget_action
- [] update_budget
- [] update_budget_action
- [] update_notification
- [] update_subscriber

2.8.12 ce

class moto.ce.models.**CostExplorerBackend**(*region_name: str, account_id: str*)
Implementation of CostExplorer APIs.

Example usage

```
@mock_ce
def test_ce_behaviour:
    boto3.client("ce")
    ...
```

Implemented features for this service

- [] create_anomaly_monitor
- [] create_anomaly_subscription
- [X] create_cost_category_definition
 - [] The EffectiveOn and ResourceTags-parameters are not yet implemented
- [] delete_anomaly_monitor

- [] delete_anomaly_subscription
- [X] delete_cost_category_definition
 - The EffectiveOn-parameter is not yet implemented
- [X] describe_cost_category_definition
 - The EffectiveOn-parameter is not yet implemented
- [] get_anomalies
- [] get_anomaly_monitors
- [] get_anomaly_subscriptions
- [] get_cost_and_usage
- [] get_cost_and_usage_with_resources
- [] get_cost_categories
- [] get_cost_forecast
- [] get_dimension_values
- [] get_reservation_coverage
- [] get_reservation_purchase_recommendation
- [] get_reservation_utilization
- [] get_rightsizing_recommendation
- [] get_savings_plans_coverage
- [] get_savings_plans_purchase_recommendation
- [] get_savings_plans_utilization
- [] get_savings_plans_utilization_details
- [] get_tags
- [] get_usage_forecast
- [] list_cost_allocation_tags
- [] list_cost_category_definitions
- [] list_savings_plans_purchase_recommendation_generation
- [X] list_tags_for_resource
- [] provide_anomaly_feedback
- [] start_savings_plans_purchase_recommendation_generation
- [X] tag_resource
- [X] untag_resource
- [] update_anomaly_monitor
- [] update_anomaly_subscription
- [] update_cost_allocation_tags_status
- [X] update_cost_category_definition
 - The EffectiveOn-parameter is not yet implemented

2.8.13 cloudformation

class `moto.cloudformation.models.CloudFormationBackend`(*region_name: str, account_id: str*)

CustomResources are supported when running Moto in ServerMode. Because creating these resources involves running a Lambda-function that informs the MotoServer about the status of the resources, the MotoServer has to be reachable for outside connections. This means it has to run inside a Docker-container, or be started using `moto_server -h 0.0.0.0`.

Example usage

```
@mock_cloudformation
def test_cloudformation_behaviour:
    boto3.client("cloudformation")
    ...
```

Implemented features for this service

- [] `activate_type`
- [] `batch_describe_type_configurations`
- [] `cancel_update_stack`
- [] `continue_update_rollback`
- [X] `create_change_set`
- [X] `create_stack`

The functionality behind `EnableTerminationProtection` is not yet implemented.

- [X] `create_stack_instances`

The following parameters are not yet implemented: `DeploymentTargets.AccountFilterType`, `DeploymentTargets.AccountsUrl`, `OperationPreferences`, `CallAs`

- [X] `create_stack_set`

The following parameters are not yet implemented: `StackId`, `AdministrationRoleARN`, `AutoDeployment`, `ExecutionRoleName`, `CallAs`, `ClientRequestToken`, `ManagedExecution`

- [] `deactivate_type`
- [X] `delete_change_set`
- [X] `delete_stack`
- [X] `delete_stack_instances`

The following parameters are not yet implemented: `DeploymentTargets`, `OperationPreferences`, `RetainStacks`, `OperationId`, `CallAs`

- [X] `delete_stack_set`
- [] `deregister_type`
- [] `describe_account_limits`
- [X] `describe_change_set`
- [] `describe_change_set_hooks`
- [] `describe_publisher`
- [] `describe_stack_drift_detection_status`

- describe_stack_events
- describe_stack_instance
- describe_stack_resource
- describe_stack_resource_drifts
- describe_stack_resources
- describe_stack_set
- describe_stack_set_operation
- describe_stacks
- describe_type
- describe_type_registration
- detect_stack_drift
- detect_stack_resource_drift
- detect_stack_set_drift
- estimate_template_cost
- execute_change_set
- get_stack_policy
- get_template
- get_template_summary
- import_stacks_to_stack_set
- list_change_sets
- list_exports
- list_imports
- list_stack_instances

 Pagination is not yet implemented. The parameters StackInstanceAccount/StackInstanceRegion are not yet implemented.

- list_stack_resources
- list_stack_set_operation_results
- list_stack_set_operations
- list_stack_sets
- list_stacks
- list_type_registrations
- list_type_versions
- list_types
- publish_type
- record_handler_progress
- register_publisher

- [] register_type
- [] rollback_stack
- [X] set_stack_policy

Note that Moto does no validation/parsing/enforcement of this policy - we simply persist it.

- [] set_type_configuration
- [] set_type_default_version
- [] signal_resource
- [X] stop_stack_set_operation
- [] test_type
- [X] update_stack
- [X] update_stack_instances

Calling this will update the parameters, but the actual resources are not updated

- [X] update_stack_set
- [] update_termination_protection
- [X] validate_template

2.8.14 cloudfront

Example usage

```
@mock_cloudfront
def test_cloudfront_behaviour:
    boto3.client("cloudfront")
    ...
```

Implemented features for this service

- [] associate_alias
- [] copy_distribution
- [] create_cache_policy
- [] create_cloud_front_origin_access_identity
- [] create_continuous_deployment_policy
- [X] create_distribution

Not all configuration options are supported yet. Please raise an issue if we're not persisting/returning the correct attributes for your use-case.

- [X] create_distribution_with_tags
- [] create_field_level_encryption_config
- [] create_field_level_encryption_profile
- [] create_function
- [X] create_invalidation
- [] create_key_group

- [] create_monitoring_subscription
- [] create_origin_access_control
- [] create_origin_request_policy
- [] create_public_key
- [] create_realtime_log_config
- [] create_response_headers_policy
- [] create_streaming_distribution
- [] create_streaming_distribution_with_tags
- [] delete_cache_policy
- [] delete_cloud_front_origin_access_identity
- [] delete_continuous_deployment_policy
- [X] delete_distribution

The IfMatch-value is ignored - any value is considered valid. Calling this function without a value is invalid, per AWS' behaviour

- [] delete_field_level_encryption_config
- [] delete_field_level_encryption_profile
- [] delete_function
- [] delete_key_group
- [] delete_monitoring_subscription
- [] delete_origin_access_control
- [] delete_origin_request_policy
- [] delete_public_key
- [] delete_realtime_log_config
- [] delete_response_headers_policy
- [] delete_streaming_distribution
- [] describe_function
- [] get_cache_policy
- [] get_cache_policy_config
- [] get_cloud_front_origin_access_identity
- [] get_cloud_front_origin_access_identity_config
- [] get_continuous_deployment_policy
- [] get_continuous_deployment_policy_config
- [X] get_distribution
- [X] get_distribution_config
- [] get_field_level_encryption
- [] get_field_level_encryption_config

- [] `get_field_level_encryption_profile`
- [] `get_field_level_encryption_profile_config`
- [] `get_function`
- [] `get_invalidation`
- [] `get_key_group`
- [] `get_key_group_config`
- [] `get_monitoring_subscription`
- [] `get_origin_access_control`
- [] `get_origin_access_control_config`
- [] `get_origin_request_policy`
- [] `get_origin_request_policy_config`
- [] `get_public_key`
- [] `get_public_key_config`
- [] `get_realtime_log_config`
- [] `get_response_headers_policy`
- [] `get_response_headers_policy_config`
- [] `get_streaming_distribution`
- [] `get_streaming_distribution_config`
- [] `list_cache_policies`
- [] `list_cloud_front_origin_access_identities`
- [] `list_conflicting_aliases`
- [] `list_continuous_deployment_policies`
- [X] `list_distributions`
 - `Pagination is not supported yet.`
- [] `list_distributions_by_cache_policy_id`
- [] `list_distributions_by_key_group`
- [] `list_distributions_by_origin_request_policy_id`
- [] `list_distributions_by_realtime_log_config`
- [] `list_distributions_by_response_headers_policy_id`
- [] `list_distributions_by_web_acl_id`
- [] `list_field_level_encryption_configs`
- [] `list_field_level_encryption_profiles`
- [] `list_functions`
- [X] `list_invalidations`
 - `Pagination is not yet implemented`
- [] `list_key_groups`

- [] list_origin_access_controls
- [] list_origin_request_policies
- [] list_public_keys
- [] list_realtime_log_configs
- [] list_response_headers_policies
- [] list_streaming_distributions
- [X] list_tags_for_resource
- [] publish_function
- [] tag_resource
- [] test_function
- [] untag_resource
- [] update_cache_policy
- [] update_cloud_front_origin_access_identity
- [] update_continuous_deployment_policy
- [X] update_distribution

The IfMatch-value is ignored - any value is considered valid. Calling this function without a value is invalid, per AWS' behaviour

- [] update_distribution_with_staging_config
- [] update_field_level_encryption_config
- [] update_field_level_encryption_profile
- [] update_function
- [] update_key_group
- [] update_origin_access_control
- [] update_origin_request_policy
- [] update_public_key
- [] update_realtime_log_config
- [] update_response_headers_policy
- [] update_streaming_distribution

2.8.15 cloudtrail

class moto.cloudtrail.models.**CloudTrailBackend**(*region_name: str, account_id: str*)

Implementation of CloudTrail APIs.

Example usage

```
@mock_cloudtrail
def test_cloudtrail_behaviour:
    boto3.client("cloudtrail")
    ...
```

Implemented features for this service

- [X] add_tags
- [] cancel_query
- [] create_channel
- [] create_event_data_store
- [X] create_trail
- [] delete_channel
- [] delete_event_data_store
- [] delete_resource_policy
- [X] delete_trail
- [] deregister_organization_delegated_admin
- [] describe_query
- [X] describe_trails
- [] get_channel
- [] get_event_data_store
- [X] get_event_selectors
- [] get_import
- [X] get_insight_selectors
- [] get_query_results
- [] get_resource_policy
- [X] get_trail
- [X] get_trail_status
- [] list_channels
- [] list_event_data_stores
- [] list_import_failures
- [] list_imports
- [] list_public_keys
- [] list_queries
- [X] list_tags

 Pagination is not yet implemented

- [X] list_trails
- [] lookup_events
- [X] put_event_selectors
- [X] put_insight_selectors
- [] put_resource_policy
- [] register_organization_delegated_admin

- [X] remove_tags
- [] restore_event_data_store
- [] start_import
- [X] start_logging
- [] start_query
- [] stop_import
- [X] stop_logging
- [] update_channel
- [] update_event_data_store
- [X] update_trail

2.8.16 cloudwatch

Example usage

```
@mock_cloudwatch
def test_cloudwatch_behaviour:
    boto3.client("cloudwatch")
    ...
```

Implemented features for this service

- [X] delete_alarms
- [] delete_anomaly_detector
- [X] delete_dashboards
- [] delete_insight_rules
- [] delete_metric_stream
- [] describe_alarm_history
- [] describe_alarms
- [] describe_alarms_for_metric
- [] describe_anomaly_detectors
- [] describe_insight_rules
- [] disable_alarm_actions
- [] disable_insight_rules
- [] enable_alarm_actions
- [] enable_insight_rules
- [X] get_dashboard
- [] get_insight_rule_report
- [X] get_metric_data
- [X] get_metric_statistics

- [] `get_metric_stream`
- [] `get_metric_widget_image`
- [X] `list_dashboards`
- [] `list_managed_insight_rules`
- [] `list_metric_streams`
- [X] `list_metrics`
- [X] `list_tags_for_resource`
- [] `put_anomaly_detector`
- [] `put_composite_alarm`
- [X] `put_dashboard`
- [] `put_insight_rule`
- [] `put_managed_insight_rules`
- [X] `put_metric_alarm`
- [X] `put_metric_data`
- [] `put_metric_stream`
- [X] `set_alarm_state`
- [] `start_metric_streams`
- [] `stop_metric_streams`
- [X] `tag_resource`
- [X] `untag_resource`

2.8.17 codebuild

Example usage

```
@mock_codebuild
def test_codebuild_behaviour:
    boto3.client("codebuild")
    ...
```

Implemented features for this service

- [] `batch_delete_builds`
- [] `batch_get_build_batches`
- [X] `batch_get_builds`
- [] `batch_get_projects`
- [] `batch_get_report_groups`
- [] `batch_get_reports`
- [X] `create_project`
- [] `create_report_group`

- [] create_webhook
- [] delete_build_batch
- [X] delete_project
- [] delete_report
- [] delete_report_group
- [] delete_resource_policy
- [] delete_source_credentials
- [] delete_webhook
- [] describe_code_coverages
- [] describe_test_cases
- [] get_report_group_trend
- [] get_resource_policy
- [] import_source_credentials
- [] invalidate_project_cache
- [] list_build_batches
- [] list_build_batches_for_project
- [X] list_builds
- [X] list_builds_for_project
- [] list_curated_environment_images
- [X] list_projects
- [] list_report_groups
- [] list_reports
- [] list_reports_for_report_group
- [] list_shared_projects
- [] list_shared_report_groups
- [] list_source_credentials
- [] put_resource_policy
- [] retry_build
- [] retry_build_batch
- [X] start_build
- [] start_build_batch
- [X] stop_build
- [] stop_build_batch
- [] update_project
- [] update_project_visibility
- [] update_report_group

- [] update_webhook

2.8.18 codecommit

Example usage

```
@mock_codecommit
def test_codecommit_behaviour:
    boto3.client("codecommit")
    ...
```

Implemented features for this service

- [] associate_approval_rule_template_with_repository
- [] batch_associate_approval_rule_template_with_repositories
- [] batch_describe_merge_conflicts
- [] batch_disassociate_approval_rule_template_from_repositories
- [] batch_get_commits
- [] batch_get_repositories
- [] create_approval_rule_template
- [] create_branch
- [] create_commit
- [] create_pull_request
- [] create_pull_request_approval_rule
- [X] create_repository
- [] create_unreferenced_merge_commit
- [] delete_approval_rule_template
- [] delete_branch
- [] delete_comment_content
- [] delete_file
- [] delete_pull_request_approval_rule
- [X] delete_repository
- [] describe_merge_conflicts
- [] describe_pull_request_events
- [] disassociate_approval_rule_template_from_repository
- [] evaluate_pull_request_approval_rules
- [] get_approval_rule_template
- [] get_blob
- [] get_branch
- [] get_comment

- [] `get_comment_reactions`
- [] `get_comments_for_compared_commit`
- [] `get_comments_for_pull_request`
- [] `get_commit`
- [] `get_differences`
- [] `get_file`
- [] `get_folder`
- [] `get_merge_commit`
- [] `get_merge_conflicts`
- [] `get_merge_options`
- [] `get_pull_request`
- [] `get_pull_request_approval_states`
- [] `get_pull_request_override_state`
- [X] `get_repository`
- [] `get_repository_triggers`
- [] `list_approval_rule_templates`
- [] `list_associated_approval_rule_templates_for_repository`
- [] `list_branches`
- [] `list_pull_requests`
- [] `list_repositories`
- [] `list_repositories_for_approval_rule_template`
- [] `list_tags_for_resource`
- [] `merge_branches_by_fast_forward`
- [] `merge_branches_by_squash`
- [] `merge_branches_by_three_way`
- [] `merge_pull_request_by_fast_forward`
- [] `merge_pull_request_by_squash`
- [] `merge_pull_request_by_three_way`
- [] `override_pull_request_approval_rules`
- [] `post_comment_for_compared_commit`
- [] `post_comment_for_pull_request`
- [] `post_comment_reply`
- [] `put_comment_reaction`
- [] `put_file`
- [] `put_repository_triggers`
- [] `tag_resource`

- [] test_repository_triggers
- [] untag_resource
- [] update_approval_rule_template_content
- [] update_approval_rule_template_description
- [] update_approval_rule_template_name
- [] update_comment
- [] update_default_branch
- [] update_pull_request_approval_rule_content
- [] update_pull_request_approval_state
- [] update_pull_request_description
- [] update_pull_request_status
- [] update_pull_request_title
- [] update_repository_description
- [] update_repository_name

2.8.19 codepipeline

Example usage

```
@mock_codepipeline
def test_codepipeline_behaviour:
    boto3.client("codepipeline")
    ...
```

Implemented features for this service

- [] acknowledge_job
- [] acknowledge_third_party_job
- [] create_custom_action_type
- [X] create_pipeline
- [] delete_custom_action_type
- [X] delete_pipeline
- [] delete_webhook
- [] deregister_webhook_with_third_party
- [] disable_stage_transition
- [] enable_stage_transition
- [] get_action_type
- [] get_job_details
- [X] get_pipeline
- [] get_pipeline_execution

- [] get_pipeline_state
- [] get_third_party_job_details
- [] list_action_executions
- [] list_action_types
- [] list_pipeline_executions
- [X] list_pipelines
- [X] list_tags_for_resource
- [] list_webhooks
- [] poll_for_jobs
- [] poll_for_third_party_jobs
- [] put_action_revision
- [] put_approval_result
- [] put_job_failure_result
- [] put_job_success_result
- [] put_third_party_job_failure_result
- [] put_third_party_job_success_result
- [] put_webhook
- [] register_webhook_with_third_party
- [] retry_stage_execution
- [] start_pipeline_execution
- [] stop_pipeline_execution
- [X] tag_resource
- [X] untag_resource
- [] update_action_type
- [X] update_pipeline

2.8.20 cognito-identity

Example usage

```
@mock_cognitoidentity
def test_cognitoidentity_behaviour:
    boto3.client("cognito-identity")
    ...
```

Implemented features for this service

- [X] create_identity_pool
- [] delete_identities
- [] delete_identity_pool

- [] describe_identity
- [X] describe_identity_pool
- [X] get_credentials_for_identity
- [X] get_id
- [] get_identity_pool_roles
- [X] get_open_id_token
- [X] get_open_id_token_for_developer_identity
- [] get_principal_tag_attribute_map
- [X] list_identities

The MaxResults-parameter has not yet been implemented

- [] list_identity_pools
- [] list_tags_for_resource
- [] lookup_developer_identity
- [] merge_developer_identities
- [] set_identity_pool_roles
- [] set_principal_tag_attribute_map
- [] tag_resource
- [] unlink_developer_identity
- [] unlink_identity
- [] untag_resource
- [X] update_identity_pool

The AllowClassic-parameter has not yet been implemented

2.8.21 cognito-idp

class moto.cognitoidp.models.CognitoIdpBackend(*region_name: str, account_id: str*)

Moto mocks the JWK uris. If you're using decorators, you can retrieve this information by making a call to <https://cognito-idp.us-west-2.amazonaws.com/someuserpoolid/.well-known/jwks.json>.

Call <http://localhost:5000/userpoolid/.well-known/jwks.json> instead of you're running Moto in Server-Mode or Docker. Because Moto cannot determine this is a CognitoIDP-request based on the URL alone, you have to add an Authorization-header instead: *Authorization: AWS4-HMAC-SHA256 Credential=mock_access_key/20220524/us-east-1/cognito-idp/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=asdf*

In some cases, you need to have reproducible IDs for the user pool. For example, a single initialization before the start of integration tests.

This behavior can be enabled by passing the environment variable: `MOTO_COGNITO_IDP_USER_POOL_ID_STRATEGY=HASH`.

Example usage

```
@mock_cognitoidp
def test_cognitoidp_behaviour:
    boto3.client("cognito-idp")
    ...
```

Implemented features for this service

- [X] add_custom_attributes
- [X] admin_add_user_to_group
- [X] admin_confirm_sign_up
- [X] admin_create_user
- [X] admin_delete_user
- [X] admin_delete_user_attributes
- [] admin_disable_provider_for_user
- [X] admin_disable_user
- [X] admin_enable_user
- [] admin_forget_device
- [] admin_get_device
- [X] admin_get_user
- [X] admin_initiate_auth
- [] admin_link_provider_for_user
- [] admin_list_devices
- [X] admin_list_groups_for_user
- [] admin_list_user_auth_events
- [X] admin_remove_user_from_group
- [X] admin_reset_user_password
- [] admin_respond_to_auth_challenge
- [X] admin_set_user_mfa_preference
- [X] admin_set_user_password
- [] admin_set_user_settings
- [] admin_update_auth_event_feedback
- [] admin_update_device_status
- [X] admin_update_user_attributes
- [X] admin_user_global_sign_out
- [X] associate_software_token
- [X] change_password
- [] confirm_device
- [X] confirm_forgot_password

- [X] confirm_sign_up
- [X] create_group
- [X] create_identity_provider
- [X] create_resource_server
- [] create_user_import_job
- [X] create_user_pool
- [X] create_user_pool_client
- [X] create_user_pool_domain
- [X] delete_group
- [X] delete_identity_provider
- [] delete_resource_server
- [] delete_user
- [] delete_user_attributes
- [X] delete_user_pool
- [X] delete_user_pool_client
- [X] delete_user_pool_domain
- [X] describe_identity_provider
- [] describe_resource_server
- [] describe_risk_configuration
- [] describe_user_import_job
- [X] describe_user_pool
- [X] describe_user_pool_client
- [X] describe_user_pool_domain
- [] forget_device
- [X] forgot_password

The ForgotPassword operation is partially broken in AWS. If the input is 100% correct it works fine.

Otherwise you get semi-random garbage and HTTP 200 OK, for example: - recovery for username which is not registered in any cognito pool - recovery for username belonging to a different user pool than the client id is registered to - phone-based recovery for a user without phone_number / phone_number_verified attributes - same as above, but email / email_verified

- [] get_csv_header
- [] get_device
- [X] get_group
- [] get_identity_provider_by_identifier
- [] get_signing_certificate
- [] get_ui_customization
- [X] get_user

- [] get_user_attribute_verification_code
- [X] get_user_pool_mfa_config
- [X] global_sign_out
- [X] initiate_auth
- [] list_devices
- [X] list_groups
- [X] list_identity_providers
- [] list_resource_servers
- [] list_tags_for_resource
- [] list_user_import_jobs
- [X] list_user_pool_clients
- [X] list_user_pools
- [X] list_users
- [X] list_users_in_group
- [] resend_confirmation_code
- [X] respond_to_auth_challenge
- [] revoke_token
- [] set_risk_configuration
- [] set_ui_customization
- [X] set_user_mfa_preference
- [X] set_user_pool_mfa_config
- [] set_user_settings
- [X] sign_up
- [] start_user_import_job
- [] stop_user_import_job
- [] tag_resource
- [] untag_resource
- [] update_auth_event_feedback
- [] update_device_status
- [X] update_group
- [X] update_identity_provider
- [] update_resource_server
- [X] update_user_attributes

The parameter ClientMetadata has not yet been implemented. No CodeDeliveryDetails are returned.

- [X] update_user_pool
- [X] update_user_pool_client

- [X] update_user_pool_domain
- [X] verify_software_token

The parameter `UserCode` has not yet been implemented

- [] verify_user_attribute

2.8.22 comprehend

class moto.comprehend.models.**ComprehendBackend**(*region_name: str, account_id: str*)
Implementation of Comprehend APIs.

Example usage

```
@mock_comprehend
def test_comprehend_behaviour:
    boto3.client("comprehend")
    ...
```

Implemented features for this service

- [] batch_detect_dominant_language
- [] batch_detect_entities
- [] batch_detect_key_phrases
- [] batch_detect_sentiment
- [] batch_detect_syntax
- [] batch_detect_targeted_sentiment
- [] classify_document
- [] contains_pii_entities
- [] create_document_classifier
- [] create_endpoint
- [X] create_entity_recognizer
The `ClientRequestToken`-parameter is not yet implemented
- [] delete_document_classifier
- [] delete_endpoint
- [X] delete_entity_recognizer
- [] delete_resource_policy
- [] describe_document_classification_job
- [] describe_document_classifier
- [] describe_dominant_language_detection_job
- [] describe_endpoint
- [] describe_entities_detection_job
- [X] describe_entity_recognizer
- [] describe_events_detection_job

- [] describe_key_phrases_detection_job
- [] describe_pii_entities_detection_job
- [] describe_resource_policy
- [] describe_sentiment_detection_job
- [] describe_targeted_sentiment_detection_job
- [] describe_topics_detection_job
- [] detect_dominant_language
- [] detect_entities
- [] detect_key_phrases
- [] detect_pii_entities
- [] detect_sentiment
- [] detect_syntax
- [] detect_targeted_sentiment
- [] import_model
- [] list_document_classification_jobs
- [] list_document_classifier_summaries
- [] list_document_classifiers
- [] list_dominant_language_detection_jobs
- [] list_endpoints
- [] list_entities_detection_jobs
- [] list_entity_recognizer_summaries
- [X] list_entity_recognizers

 Pagination is not yet implemented. The following filters are not yet implemented: Status, SubmitTimeBefore, SubmitTimeAfter

- [] list_events_detection_jobs
- [] list_key_phrases_detection_jobs
- [] list_pii_entities_detection_jobs
- [] list_sentiment_detection_jobs
- [X] list_tags_for_resource
- [] list_targeted_sentiment_detection_jobs
- [] list_topics_detection_jobs
- [] put_resource_policy
- [] start_document_classification_job
- [] start_dominant_language_detection_job
- [] start_entities_detection_job
- [] start_events_detection_job

- [] start_key_phrases_detection_job
- [] start_pii_entities_detection_job
- [] start_sentiment_detection_job
- [] start_targeted_sentiment_detection_job
- [] start_topics_detection_job
- [] stop_dominant_language_detection_job
- [] stop_entities_detection_job
- [] stop_events_detection_job
- [] stop_key_phrases_detection_job
- [] stop_pii_entities_detection_job
- [] stop_sentiment_detection_job
- [] stop_targeted_sentiment_detection_job
- [] stop_training_document_classifier
- [X] stop_training_entity_recognizer
- [X] tag_resource
- [X] untag_resource
- [] update_endpoint

2.8.23 config

Example usage

```
@mock_config
def test_config_behaviour:
    boto3.client("config")
    ...
```

Implemented features for this service

- [X] batch_get_aggregate_resource_config Returns configuration of resource for current regional backend.
 - Item is returned in AWS Config format.
 - As far a moto goes – the only real difference between this function and the *batch_get_resource_config* function is that this will require a Config Aggregator be set up a priori and can search based on resource regions.
 - Note: moto will IGNORE the resource account ID in the search query.
- [X] batch_get_resource_config Returns configuration of resource for the current regional backend.
 - Item is returned in AWS Config format.
 - param resource_keys**
 - param backend_region**
- [X] delete_aggregation_authorization
- [X] delete_config_rule Delete config rule used for evaluating resource compliance.

- delete_configuration_aggregator
- delete_configuration_recorder
- delete_conformance_pack
- delete_delivery_channel
- delete_evaluation_results
- delete_organization_config_rule
- delete_organization_conformance_pack
- delete_pending_aggregation_request
- delete_remediation_configuration
- delete_remediation_exceptions
- delete_resource_config
- delete_retention_configuration
- delete_stored_query
- deliver_config_snapshot
- describe_aggregate_compliance_by_config_rules
- describe_aggregate_compliance_by_conformance_packs
- describe_aggregation_authorizations
- describe_compliance_by_config_rule
- describe_compliance_by_resource
- describe_config_rule_evaluation_status
- describe_config_rules Return details for the given ConfigRule names or for all rules.
- describe_configuration_aggregator_sources_status
- describe_configuration_aggregators
- describe_configuration_recorder_status
- describe_configuration_recorders
- describe_conformance_pack_compliance
- describe_conformance_pack_status
- describe_conformance_packs
- describe_delivery_channel_status
- describe_delivery_channels
- describe_organization_config_rule_statuses
- describe_organization_config_rules
- describe_organization_conformance_pack_statuses
- describe_organization_conformance_packs
- describe_pending_aggregation_requests
- describe_remediation_configurations

- [] describe_remediation_exceptions
- [] describe_remediation_execution_status
- [] describe_retention_configurations
- [] get_aggregate_compliance_details_by_config_rule
- [] get_aggregate_config_rule_compliance_summary
- [] get_aggregate_conformance_pack_compliance_summary
- [] get_aggregate_discovered_resource_counts
- [] get_aggregate_resource_config
- [] get_compliance_details_by_config_rule
- [] get_compliance_details_by_resource
- [] get_compliance_summary_by_config_rule
- [] get_compliance_summary_by_resource_type
- [] get_conformance_pack_compliance_details
- [] get_conformance_pack_compliance_summary
- [] get_custom_rule_policy
- [] get_discovered_resource_counts
- [] get_organization_config_rule_detailed_status
- [X] get_organization_conformance_pack_detailed_status
- [] get_organization_custom_rule_policy
- [X] get_resource_config_history Returns configuration of resource for the current regional backend.

Item returned in AWS Config format.

NOTE: This is –NOT– returning history as it is not supported in moto at this time. (PR's welcome!)

As such, the `later_time`, `earlier_time`, `limit`, and `next_token` are ignored as this will only return 1 item. (If no items, it raises an exception).

- [] get_resource_evaluation_summary
- [] get_stored_query
- [X] list_aggregate_discovered_resources Queries AWS Config listing function that must exist for resource backend.

As far a moto goes – the only real difference between this function and the `list_discovered_resources` function is that this will require a Config Aggregator be set up a priori and can search based on resource regions.

param aggregator_name

param resource_type

param filters

param limit

param next_token

return

- [] list_conformance_pack_compliance_scores
- [X] list_discovered_resources Queries against AWS Config (non-aggregated) listing function.

The listing function must exist for the resource backend.

param resource_type
param backend_region
param ids
param name
param limit
param next_token
return

- [] list_resource_evaluations
- [] list_stored_queries
- [X] list_tags_for_resource Return list of tags for AWS Config resource.
- [X] put_aggregation_authorization
- [X] put_config_rule Add/Update config rule for evaluating resource compliance.

TBD - Only the “accounting” of config rules are handled at the moment. No events are created or triggered. There is no interaction with the config recorder.

- [X] put_configuration_aggregator
- [X] put_configuration_recorder
- [] put_conformance_pack
- [X] put_delivery_channel
- [X] put_evaluations
- [] put_external_evaluation
- [] put_organization_config_rule
- [X] put_organization_conformance_pack
- [] put_remediation_configurations
- [] put_remediation_exceptions
- [] put_resource_config
- [] put_retention_configuration
- [] put_stored_query
- [] select_aggregate_resource_config
- [] select_resource_config
- [] start_config_rules_evaluation
- [X] start_configuration_recorder
- [] start_remediation_execution
- [] start_resource_evaluation

- [X] stop_configuration_recorder
- [X] tag_resource Add tags in config with a matching ARN.
- [X] untag_resource Remove tags in config with a matching ARN.

If the tags in the tag_keys don't match any keys for that ARN, they're just ignored.

2.8.24 databrew

Example usage

```
@mock_databrew
def test_databrew_behaviour:
    boto3.client("databrew")
    ...
```

Implemented features for this service

- [] batch_delete_recipe_version
- [X] create_dataset
- [X] create_profile_job
- [] create_project
- [X] create_recipe
- [X] create_recipe_job
- [X] create_ruleset
- [] create_schedule
- [X] delete_dataset
- [X] delete_job
- [] delete_project
- [X] delete_recipe_version
- [X] delete_ruleset
- [] delete_schedule
- [X] describe_dataset
- [X] describe_job
- [] describe_job_run
- [] describe_project
- [X] describe_recipe
- [X] describe_ruleset
- [] describe_schedule
- [X] list_datasets
- [] list_job_runs
- [X] list_jobs

- [] list_projects
- [X] list_recipe_versions
- [X] list_recipes
- [X] list_rulesets
- [] list_schedules
- [] list_tags_for_resource
- [X] publish_recipe
- [] send_project_session_action
- [] start_job_run
- [] start_project_session
- [] stop_job_run
- [] tag_resource
- [] untag_resource
- [X] update_dataset
- [X] update_profile_job
- [] update_project
- [X] update_recipe
- [X] update_recipe_job
- [X] update_ruleset
- [] update_schedule

2.8.25 datapipeline

Example usage

```
@mock_datapipeline
def test_datapipeline_behaviour:
    boto3.client("datapipeline")
    ...
```

Implemented features for this service

- [X] activate_pipeline
- [] add_tags
- [X] create_pipeline
- [] deactivate_pipeline
- [X] delete_pipeline
- [X] describe_objects
- [X] describe_pipelines
- [] evaluate_expression

- [X] get_pipeline_definition
- [X] list_pipelines
- [] poll_for_task
- [X] put_pipeline_definition
- [] query_objects
- [] remove_tags
- [] report_task_progress
- [] report_task_runner_heartbeat
- [] set_status
- [] set_task_status
- [] validate_pipeline_definition

2.8.26 datasync

Example usage

```
@mock_datasync
def test_datasync_behaviour:
    boto3.client("datasync")
    ...
```

Implemented features for this service

- [X] cancel_task_execution
- [] create_agent
- [] create_location_efs
- [] create_location_fsx_lustre
- [] create_location_fsx_ontap
- [] create_location_fsx_open_zfs
- [] create_location_fsx_windows
- [] create_location_hdfs
- [] create_location_nfs
- [] create_location_object_storage
- [] create_location_s3
- [] create_location_smb
- [X] create_task
- [] delete_agent
- [X] delete_location
- [X] delete_task
- [] describe_agent

- [] describe_location_efs
- [] describe_location_fsx_lustre
- [] describe_location_fsx_ontap
- [] describe_location_fsx_open_zfs
- [] describe_location_fsx_windows
- [] describe_location_hdfs
- [] describe_location_nfs
- [] describe_location_object_storage
- [] describe_location_s3
- [] describe_location_smb
- [] describe_task
- [] describe_task_execution
- [] list_agents
- [] list_locations
- [] list_tags_for_resource
- [] list_task_executions
- [] list_tasks
- [X] start_task_execution
- [] tag_resource
- [] untag_resource
- [] update_agent
- [] update_location_hdfs
- [] update_location_nfs
- [] update_location_object_storage
- [] update_location_smb
- [X] update_task
- [] update_task_execution

2.8.27 dax

Example usage

```
@mock_dax
def test_dax_behaviour:
    boto3.client("dax")
    ...
```

Implemented features for this service

- [X] create_cluster

The following parameters are not yet processed: AvailabilityZones, SubnetGroupNames, SecurityGroups, PreferredMaintenanceWindow, NotificationTopicArn, ParameterGroupName

- [] create_parameter_group
- [] create_subnet_group
- [X] decrease_replication_factor

The AvailabilityZones-parameter is not yet implemented

- [X] delete_cluster
- [] delete_parameter_group
- [] delete_subnet_group
- [X] describe_clusters
- [] describe_default_parameters
- [] describe_events
- [] describe_parameter_groups
- [] describe_parameters
- [] describe_subnet_groups
- [X] increase_replication_factor

The AvailabilityZones-parameter is not yet implemented

- [X] list_tags

Pagination is not yet implemented

- [] reboot_node
- [] tag_resource
- [] untag_resource
- [] update_cluster
- [] update_parameter_group
- [] update_subnet_group

2.8.28 dms

Example usage

```
@mock_dms
def test_dms_behaviour:
    boto3.client("dms")
    ...
```

Implemented features for this service

- [] add_tags_to_resource
- [] apply_pending_maintenance_action
- [] cancel_replication_task_assessment_run
- [] create_endpoint

- [] create_event_subscription
- [] create_fleet_advisor_collector
- [] create_replication_instance
- [] create_replication_subnet_group
- [X] create_replication_task

The following parameters are not yet implemented: CDCStartTime, CDCStartPosition, CDCStopPosition, Tags, TaskData, ResourceIdentifier

- [] delete_certificate
- [] delete_connection
- [] delete_endpoint
- [] delete_event_subscription
- [] delete_fleet_advisor_collector
- [] delete_fleet_advisor_databases
- [] delete_replication_instance
- [] delete_replication_subnet_group
- [X] delete_replication_task
- [] delete_replication_task_assessment_run
- [] describe_account_attributes
- [] describe_applicable_individual_assessments
- [] describe_certificates
- [] describe_connections
- [] describe_endpoint_settings
- [] describe_endpoint_types
- [] describe_endpoints
- [] describe_event_categories
- [] describe_event_subscriptions
- [] describe_events
- [] describe_fleet_advisor_collectors
- [] describe_fleet_advisor_databases
- [] describe_fleet_advisor_Isa_analysis
- [] describe_fleet_advisor_schema_object_summary
- [] describe_fleet_advisor_schemas
- [] describe_orderable_replication_instances
- [] describe_pending_maintenance_actions
- [] describe_refresh_schemas_status
- [] describe_replication_instance_task_logs

- [] describe_replication_instances
- [] describe_replication_subnet_groups
- [] describe_replication_task_assessment_results
- [] describe_replication_task_assessment_runs
- [] describe_replication_task_individual_assessments
- [X] describe_replication_tasks

The parameter WithoutSettings has not yet been implemented

- [] describe_schemas
- [] describe_table_statistics
- [] import_certificate
- [] list_tags_for_resource
- [] modify_endpoint
- [] modify_event_subscription
- [] modify_replication_instance
- [] modify_replication_subnet_group
- [] modify_replication_task
- [] move_replication_task
- [] reboot_replication_instance
- [] refresh_schemas
- [] reload_tables
- [] remove_tags_from_resource
- [] run_fleet_advisor_lsa_analysis
- [X] start_replication_task

The following parameters have not yet been implemented: StartReplicationTaskType, CDCStartTime, CDCStartPosition, CDCStopPosition

- [] start_replication_task_assessment
- [] start_replication_task_assessment_run
- [X] stop_replication_task
- [] test_connection
- [] update_subscriptions_to_event_bridge

2.8.29 ds

class moto.ds.models.DirectoryServiceBackend(*region_name: str, account_id: str*)
Implementation of DirectoryService APIs.

Example usage

```
@mock_ds
def test_ds_behaviour:
    boto3.client("ds")
    ...
```

Implemented features for this service

- [] accept_shared_directory
- [] add_ip_routes
- [] add_region
- [X] add_tags_to_resource Add or overwrite one or more tags for specified directory.
- [] cancel_schema_extension
- [X] connect_directory Create a fake AD Connector.
- [X] create_alias Create and assign an alias to a directory.
- [] create_computer
- [] create_conditional_forwarder
- [X] create_directory Create a fake Simple Ad Directory.
- [] create_log_subscription
- [X] create_microsoft_ad Create a fake Microsoft Ad Directory.
- [] create_snapshot
- [] create_trust
- [] delete_conditional_forwarder
- [X] delete_directory Delete directory with the matching ID.
- [] delete_log_subscription
- [] delete_snapshot
- [] delete_trust
- [] deregister_certificate
- [] deregister_event_topic
- [] describe_certificate
- [] describe_client_authentication_settings
- [] describe_conditional_forwarders
- [X] describe_directories Return info on all directories or directories with matching IDs.
- [] describe_domain_controllers
- [] describe_event_topics

- describe_ldaps_settings
- describe_regions
- describe_settings
- describe_shared_directories
- describe_snapshots
- describe_trusts
- describe_update_directory
- disable_client_authentication
- disable_ldaps
- disable_radius
- disable_sso Disable single-sign on for a directory.
- enable_client_authentication
- enable_ldaps
- enable_radius
- enable_sso Enable single-sign on for a directory.
- get_directory_limits Return hard-coded limits for the directories.
- get_snapshot_limits
- list_certificates
- list_ip_routes
- list_log_subscriptions
- list_schema_extensions
- list_tags_for_resource List all tags on a directory.
- register_certificate
- register_event_topic
- reject_shared_directory
- remove_ip_routes
- remove_region
- remove_tags_from_resource Removes tags from a directory.
- reset_user_password
- restore_from_snapshot
- share_directory
- start_schema_extension
- unshare_directory
- update_conditional_forwarder
- update_directory_setup
- update_number_of_domain_controllers

- [] update_radius
- [] update_settings
- [] update_trust
- [] verify_trust

2.8.30 dynamodb

Example usage

```
@mock_dynamodb
def test_dynamodb_behaviour:
    boto3.client("dynamodb")
    ...
```

Implemented features for this service

- [] batch_execute_statement
- [X] batch_get_item
- [X] batch_write_item
- [X] create_backup
- [] create_global_table
- [X] create_table
- [X] delete_backup
- [X] delete_item
- [X] delete_table
- [X] describe_backup
- [X] describe_continuous_backups
- [] describe_contributor_insights
- [X] describe_endpoints
- [] describe_export
- [] describe_global_table
- [] describe_global_table_settings
- [] describe_import
- [] describe_kinesis_streaming_destination
- [] describe_limits
- [X] describe_table
- [] describe_table_replica_auto_scaling
- [X] describe_time_to_live
- [] disable_kinesis_streaming_destination
- [] enable_kinesis_streaming_destination

- execute_statement
- execute_transaction
- export_table_to_point_in_time
- get_item
- import_table
- list_backups
- list_contributor_insights
- list_exports
- list_global_tables
- list_imports
- list_tables
- list_tags_of_resource
- put_item
- query
- restore_table_from_backup
- restore_table_to_point_in_time

Currently this only accepts the source and target table elements, and will copy all items from the source without respect to other arguments.

- scan
- tag_resource
- transact_get_items
- transact_write_items
- untag_resource
- update_continuous_backups
- update_contributor_insights
- update_global_table
- update_global_table_settings
- update_item
- update_table
- update_table_replica_auto_scaling
- update_time_to_live

2.8.31 dynamodbstreams

Example usage

```
@mock_dynamodbstreams
def test_dynamodbstreams_behaviour:
    boto3.client("dynamodbstreams")
    ...
```

Implemented features for this service

- [X] describe_stream
- [X] get_records
- [X] get_shard_iterator
- [X] list_streams

2.8.32 ebs

class moto.ebs.models.**EBSBackend**(region_name: str, account_id: str)
Implementation of EBS APIs.

Example usage

```
@mock_ebs
def test_ebs_behaviour:
    boto3.client("ebs")
    ...
```

Implemented features for this service

- [X] complete_snapshot
- [X] get_snapshot_block
 - The BlockToken-parameter is not yet implemented
- [X] list_changed_blocks
 - The following parameters are not yet implemented: NextToken, MaxResults, StartingBlockIndex
- [X] list_snapshot_blocks
- [X] put_snapshot_block
- [X] start_snapshot

2.8.33 ec2

class moto.ec2.models.**EC2Backend**(*args: Any, **kwargs: Any)
moto includes a limited set of AMIs in *moto/ec2/resources/amis.json*. Additionally, the default AMI's specified by SSM will be provided.

If you require specific AMIs to be available during your tests, you can provide your own AMI definitions by setting the environment variable *MOTO_AMIS_PATH* to point to a JSON file containing definitions of the required AMIs. No other AMI's will be loaded if this environment variable is set.

To create such a file, refer to *scripts/get_amis.py*

Note: You must set `MOTO_AMIS_PATH` before importing moto.

Example usage

```
@mock_ec2
def test_ec2_behaviour:
    boto3.client("ec2")
    ...
```

Implemented features for this service

- [] `accept_address_transfer`
- [] `accept_reserved_instances_exchange_quote`
- [] `accept_transit_gateway_multicast_domain_associations`
- [X] `accept_transit_gateway_peering_attachment`
- [] `accept_transit_gateway_vpc_attachment`
- [] `accept_vpc_endpoint_connections`
- [X] `accept_vpc_peering_connection`
- [] `advertise_byoip_cidr`
- [X] `allocate_address`
- [X] `allocate_hosts`
- [] `allocate_ipam_pool_cidr`
- [] `apply_security_groups_to_client_vpn_target_network`
- [X] `assign_ipv6_addresses`
- [X] `assign_private_ip_addresses`
- [] `assign_private_nat_gateway_address`
- [X] `associate_address`
- [] `associate_client_vpn_target_network`
- [X] `associate_dhcp_options`
- [] `associate_enclave_certificate_iam_role`
- [X] `associate_iam_instance_profile`
- [] `associate_instance_event_window`
- [] `associate_ipam_resource_discovery`
- [] `associate_nat_gateway_address`
- [X] `associate_route_table`
- [X] `associate_subnet_cidr_block`
- [] `associate_transit_gateway_multicast_domain`
- [] `associate_transit_gateway_policy_table`
- [X] `associate_transit_gateway_route_table`

- associate_trunk_interface
- associate_vpc_cidr_block
- attach_classic_link_vpc
- attach_internet_gateway
- attach_network_interface
- attach_verified_access_trust_provider
- attach_volume
- attach_vpn_gateway
- authorize_client_vpn_ingress
- authorize_security_group_egress
- authorize_security_group_ingress
- bundle_instance
- cancel_bundle_task
- cancel_capacity_reservation
- cancel_capacity_reservation_fleets
- cancel_conversion_task
- cancel_export_task
- cancel_image_launch_permission
- cancel_import_task
- cancel_reserved_instances_listing
- cancel_spot_fleet_requests
- cancel_spot_instance_requests
- confirm_product_instance
- copy_fpga_image
- copy_image
- copy_snapshot
- create_capacity_reservation
- create_capacity_reservation_fleet
- create_carrier_gateway
- create_client_vpn_endpoint
- create_client_vpn_route
- create_coip_cidr
- create_coip_pool
- create_customer_gateway
- create_default_subnet
- create_default_vpc

- create_dhcp_options
- create_egress_only_internet_gateway
- create_fleet
- create_flow_logs
- create_fpga_image
- create_image
- create_instance_event_window
- create_instance_export_task
- create_internet_gateway
- create_ipam
- create_ipam_pool
- create_ipam_resource_discovery
- create_ipam_scope
- create_key_pair
- create_launch_template
- create_launch_template_version
- create_local_gateway_route
- create_local_gateway_route_table
- create_local_gateway_route_table_virtual_interface_group_association
- create_local_gateway_route_table_vpc_association
- create_managed_prefix_list
- create_nat_gateway
- create_network_acl
- create_network_acl_entry
- create_network_insights_access_scope
- create_network_insights_path
- create_network_interface
- create_network_interface_permission
- create_placement_group
- create_public_ipv4_pool
- create_replace_root_volume_task
- create_reserved_instances_listing
- create_restore_image_task
- create_route
- create_route_table
- create_security_group

- create_snapshot
- create_snapshots

The CopyTagsFromSource-parameter is not yet implemented.

- create_spot_datafeed_subscription
- create_store_image_task
- create_subnet
- create_subnet_cidr_reservation
- create_tags
- create_traffic_mirror_filter
- create_traffic_mirror_filter_rule
- create_traffic_mirror_session
- create_traffic_mirror_target
- create_transit_gateway
- create_transit_gateway_connect
- create_transit_gateway_connect_peer
- create_transit_gateway_multicast_domain
- create_transit_gateway_peering_attachment
- create_transit_gateway_policy_table
- create_transit_gateway_prefix_list_reference
- create_transit_gateway_route
- create_transit_gateway_route_table
- create_transit_gateway_route_table_announcement
- create_transit_gateway_vpc_attachment
- create_verified_access_endpoint
- create_verified_access_group
- create_verified_access_instance
- create_verified_access_trust_provider
- create_volume
- create_vpc
- create_vpc_endpoint
- create_vpc_endpoint_connection_notification
- create_vpc_endpoint_service_configuration
- create_vpc_peering_connection
- create_vpn_connection
- create_vpn_connection_route
- create_vpn_gateway

- delete_carrier_gateway
- delete_client_vpn_endpoint
- delete_client_vpn_route
- delete_coip_cidr
- delete_coip_pool
- delete_customer_gateway
- delete_dhcp_options
- delete_egress_only_internet_gateway
- delete_fleets
- delete_flow_logs
- delete_fpga_image
- delete_instance_event_window
- delete_internet_gateway
- delete_ipam
- delete_ipam_pool
- delete_ipam_resource_discovery
- delete_ipam_scope
- delete_key_pair
- delete_launch_template
- delete_launch_template_versions
- delete_local_gateway_route
- delete_local_gateway_route_table
- delete_local_gateway_route_table_virtual_interface_group_association
- delete_local_gateway_route_table_vpc_association
- delete_managed_prefix_list
- delete_nat_gateway
- delete_network_acl
- delete_network_acl_entry
- delete_network_insights_access_scope
- delete_network_insights_access_scope_analysis
- delete_network_insights_analysis
- delete_network_insights_path
- delete_network_interface
- delete_network_interface_permission
- delete_placement_group
- delete_public_ipv4_pool

- delete_queued_reserved_instances
- delete_route
- delete_route_table
- delete_security_group
- delete_snapshot
- delete_spot_datafeed_subscription
- delete_subnet
- delete_subnet_cidr_reservation
- delete_tags
- delete_traffic_mirror_filter
- delete_traffic_mirror_filter_rule
- delete_traffic_mirror_session
- delete_traffic_mirror_target
- delete_transit_gateway
- delete_transit_gateway_connect
- delete_transit_gateway_connect_peer
- delete_transit_gateway_multicast_domain
- delete_transit_gateway_peering_attachment
- delete_transit_gateway_policy_table
- delete_transit_gateway_prefix_list_reference
- delete_transit_gateway_route
- delete_transit_gateway_route_table
- delete_transit_gateway_route_table_announcement
- delete_transit_gateway_vpc_attachment
- delete_verified_access_endpoint
- delete_verified_access_group
- delete_verified_access_instance
- delete_verified_access_trust_provider
- delete_volume
- delete_vpc
- delete_vpc_endpoint_connection_notifications
- delete_vpc_endpoint_service_configurations
- delete_vpc_endpoints
- delete_vpc_peering_connection
- delete_vpn_connection
- delete_vpn_connection_route

- delete_vpn_gateway
- deprovision_byoip_cidr
- deprovision_ipam_pool_cidr
- deprovision_public_ipv4_pool_cidr
- deregister_image
- deregister_instance_event_notification_attributes
- deregister_transit_gateway_multicast_group_members
- deregister_transit_gateway_multicast_group_sources
- describe_account_attributes
- describe_address_transfers
- describe_addresses
- describe_addresses_attribute
- describe_aggregate_id_format
- describe_availability_zones
- describe_aws_network_performance_metric_subscriptions
- describe_bundle_tasks
- describe_byoip_cidrs
- describe_capacity_reservation_fleets
- describe_capacity_reservations
- describe_carrier_gateways
- describe_classic_link_instances
- describe_client_vpn_authorization_rules
- describe_client_vpn_connections
- describe_client_vpn_endpoints
- describe_client_vpn_routes
- describe_client_vpn_target_networks
- describe_coip_pools
- describe_conversion_tasks
- describe_customer_gateways
- describe_dhcp_options
- describe_egress_only_internet_gateways

The Filters-argument is not yet supported

- describe_elastic_gpus
- describe_export_image_tasks
- describe_export_tasks
- describe_fast_launch_images

- [] describe_fast_snapshot_restores
- [] describe_fleet_history
- [X] describe_fleet_instances
- [X] describe_fleets
- [X] describe_flow_logs
- [] describe_fpga_image_attribute
- [] describe_fpga_images
- [] describe_host_reservation_offerings
- [] describe_host_reservations
- [X] describe_hosts

 Pagination is not yet implemented

- [X] describe_iam_instance_profile_associations
- [] describe_id_format
- [] describe_identity_id_format
- [] describe_image_attribute
- [X] describe_images
- [] describe_import_image_tasks
- [] describe_import_snapshot_tasks
- [X] describe_instance_attribute
- [X] describe_instance_credit_specifications
- [] describe_instance_event_notification_attributes
- [] describe_instance_event_windows
- [X] describe_instance_status
- [X] describe_instance_type_offerings
- [X] describe_instance_types
- [X] describe_instances
- [X] describe_internet_gateways
- [] describe_ipam_pools
- [] describe_ipam_resource_discoveries
- [] describe_ipam_resource_discovery_associations
- [] describe_ipam_scopes
- [] describe_ipams
- [] describe_ipv6_pools
- [X] describe_key_pairs
- [] describe_launch_template_versions
- [X] describe_launch_templates

- [] describe_local_gateway_route_table_virtual_interface_group_associations
- [] describe_local_gateway_route_table_vpc_associations
- [] describe_local_gateway_route_tables
- [] describe_local_gateway_virtual_interface_groups
- [] describe_local_gateway_virtual_interfaces
- [] describe_local_gateways
- [X] describe_managed_prefix_lists
- [] describe_moving_addresses
- [X] describe_nat_gateways
- [X] describe_network_acls
- [] describe_network_insights_access_scope_analyses
- [] describe_network_insights_access_scopes
- [] describe_network_insights_analyses
- [] describe_network_insights_paths
- [] describe_network_interface_attribute
- [] describe_network_interface_permissions
- [X] describe_network_interfaces
- [] describe_placement_groups
- [] describe_prefix_lists
- [] describe_principal_id_format
- [] describe_public_ipv4_pools
- [X] describe_regions
- [] describe_replace_root_volume_tasks
- [] describe_reserved_instances
- [] describe_reserved_instances_listings
- [] describe_reserved_instances_modifications
- [] describe_reserved_instances_offerings
- [X] describe_route_tables
- [] describe_scheduled_instance_availability
- [] describe_scheduled_instances
- [] describe_security_group_references
- [X] describe_security_group_rules
- [X] describe_security_groups
- [] describe_snapshot_attribute
- [] describe_snapshot_tier_status
- [X] describe_snapshots

- describe_spot_datafeed_subscription
- describe_spot_fleet_instances
- describe_spot_fleet_request_history
- describe_spot_fleet_requests
- describe_spot_instance_requests
- describe_spot_price_history
- describe_stale_security_groups
- describe_store_image_tasks
- describe_subnets
- describe_tags
- describe_traffic_mirror_filters
- describe_traffic_mirror_sessions
- describe_traffic_mirror_targets
- describe_transit_gateway_attachments
- describe_transit_gateway_connect_peers
- describe_transit_gateway_connects
- describe_transit_gateway_multicast_domains
- describe_transit_gateway_peering_attachments
- describe_transit_gateway_policy_tables
- describe_transit_gateway_route_table_announcements
- describe_transit_gateway_route_tables
- describe_transit_gateway_vpc_attachments
- describe_transit_gateways
- describe_trunk_interface_associations
- describe_verified_access_endpoints
- describe_verified_access_groups
- describe_verified_access_instance_logging_configurations
- describe_verified_access_instances
- describe_verified_access_trust_providers
- describe_volume_attribute
- describe_volume_status
- describe_volumes
- describe_volumes_modifications
- describe_vpc_attribute
- describe_vpc_classic_link
- describe_vpc_classic_link_dns_support

- [] describe_vpc_endpoint_connection_notifications
- [] describe_vpc_endpoint_connections
- [X] describe_vpc_endpoint_service_configurations
 - The Filters, MaxResults, NextToken parameters are not yet implemented
- [X] describe_vpc_endpoint_service_permissions
 - The Filters, MaxResults, NextToken parameters are not yet implemented
- [X] describe_vpc_endpoint_services Return info on services to which you can create a VPC endpoint.
 - Currently only the default endpoing services are returned. When create_vpc_endpoint_service_configuration() is implemented, a list of those private endpoints would be kept and when this API is invoked, those private endpoints would be added to the list of default endpoint services.
 - The DryRun parameter is ignored.
- [X] describe_vpc_endpoints
- [X] describe_vpc_peering_connections
- [X] describe_vpcs
- [X] describe_vpn_connections
- [X] describe_vpn_gateways
- [] detach_classic_link_vpc
- [X] detach_internet_gateway
- [X] detach_network_interface
- [] detach_verified_access_trust_provider
- [X] detach_volume
- [X] detach_vpn_gateway
- [] disable_address_transfer
- [] disable_aws_network_performance_metric_subscription
- [X] disable_ebs_encryption_by_default
- [] disable_fast_launch
- [] disable_fast_snapshot_restores
- [] disable_image_deprecation
- [] disable_ipam_organization_admin_account
- [] disable_serial_console_access
- [X] disable_transit_gateway_route_table_propagation
- [] disable_vgw_route_propagation
- [X] disable_vpc_classic_link
- [X] disable_vpc_classic_link_dns_support
- [X] disassociate_address
- [] disassociate_client_vpn_target_network

- disassociate_enclave_certificate_iam_role
- disassociate_iam_instance_profile
- disassociate_instance_event_window
- disassociate_ipam_resource_discovery
- disassociate_nat_gateway_address
- disassociate_route_table
- disassociate_subnet_cidr_block
- disassociate_transit_gateway_multicast_domain
- disassociate_transit_gateway_policy_table
- disassociate_transit_gateway_route_table
- disassociate_trunk_interface
- disassociate_vpc_cidr_block
- enable_address_transfer
- enable_aws_network_performance_metric_subscription
- enable_ebs_encryption_by_default
- enable_fast_launch
- enable_fast_snapshot_restores
- enable_image_deprecation
- enable_ipam_organization_admin_account
- enable_reachability_analyzer_organization_sharing
- enable_serial_console_access
- enable_transit_gateway_route_table_propagation
- enable_vgw_route_propagation
- enable_volume_io
- enable_vpc_classic_link
- enable_vpc_classic_link_dns_support
- export_client_vpn_client_certificate_revocation_list
- export_client_vpn_client_configuration
- export_image
- export_transit_gateway_routes
- get_associated_enclave_certificate_iam_roles
- get_associated_ipv6_pool_cidrs
- get_aws_network_performance_data
- get_capacity_reservation_usage
- get_coip_pool_usage
- get_console_output

- [] `get_console_screenshot`
- [] `get_default_credit_specification`
- [] `get_ebs_default_kms_key_id`
- [X] `get_ebs_encryption_by_default`
- [] `get_flow_logs_integration_template`
- [] `get_groups_for_capacity_reservation`
- [] `get_host_reservation_purchase_preview`
- [] `get_instance_types_from_instance_requirements`
- [] `get_instance_uefi_data`
- [] `get_ipam_address_history`
- [] `get_ipam_discovered_accounts`
- [] `get_ipam_discovered_resource_cidrs`
- [] `get_ipam_pool_allocations`
- [] `get_ipam_pool_cidrs`
- [] `get_ipam_resource_cidrs`
- [] `get_launch_template_data`
- [] `get_managed_prefix_list_associations`
- [X] `get_managed_prefix_list_entries`
- [] `get_network_insights_access_scope_analysis_findings`
- [] `get_network_insights_access_scope_content`
- [X] `get_password_data`
- [] `get_reserved_instances_exchange_quote`
- [] `get_serial_console_access_status`
- [] `get_spot_placement_scores`
- [] `get_subnet_cidr_reservations`
- [] `get_transit_gateway_attachment_propagations`
- [] `get_transit_gateway_multicast_domain_associations`
- [] `get_transit_gateway_policy_table_associations`
- [] `get_transit_gateway_policy_table_entries`
- [] `get_transit_gateway_prefix_list_references`
- [] `get_transit_gateway_route_table_associations`
- [] `get_transit_gateway_route_table_propagations`
- [] `get_verified_access_endpoint_policy`
- [] `get_verified_access_group_policy`
- [] `get_vpn_connection_device_sample_configuration`
- [] `get_vpn_connection_device_types`

- import_client_vpn_client_certificate_revocation_list
- import_image
- import_instance
- import_key_pair
- import_snapshot
- import_volume
- list_images_in_recycle_bin
- list_snapshots_in_recycle_bin
- modify_address_attribute
- modify_availability_zone_group
- modify_capacity_reservation
- modify_capacity_reservation_fleet
- modify_client_vpn_endpoint
- modify_default_credit_specification
- modify_ebs_default_kms_key_id
- modify_fleet
- modify_fpga_image_attribute
- modify_hosts
- modify_id_format
- modify_identity_id_format
- modify_image_attribute
- modify_instance_attribute
- modify_instance_capacity_reservation_attributes
- modify_instance_credit_specification
- modify_instance_event_start_time
- modify_instance_event_window
- modify_instance_maintenance_options
- modify_instance_metadata_options
- modify_instance_placement
- modify_ipam
- modify_ipam_pool
- modify_ipam_resource_cidr
- modify_ipam_resource_discovery
- modify_ipam_scope
- modify_launch_template
- modify_local_gateway_route

- modify_managed_prefix_list
- modify_network_interface_attribute
- modify_private_dns_name_options
- modify_reserved_instances
- modify_security_group_rules
- modify_snapshot_attribute
- modify_snapshot_tier
- modify_spot_fleet_request
- modify_subnet_attribute
- modify_traffic_mirror_filter_network_services
- modify_traffic_mirror_filter_rule
- modify_traffic_mirror_session
- modify_transit_gateway
- modify_transit_gateway_prefix_list_reference
- modify_transit_gateway_vpc_attachment
- modify_verified_access_endpoint
- modify_verified_access_endpoint_policy
- modify_verified_access_group
- modify_verified_access_group_policy
- modify_verified_access_instance
- modify_verified_access_instance_logging_configuration
- modify_verified_access_trust_provider
- modify_volume
- modify_volume_attribute
- modify_vpc_attribute
- modify_vpc_endpoint
- modify_vpc_endpoint_connection_notification
- modify_vpc_endpoint_service_configuration

The following parameters are not yet implemented: RemovePrivateDnsName

- modify_vpc_endpoint_service_payer_responsibility
- modify_vpc_endpoint_service_permissions
- modify_vpc_peering_connection_options
- modify_vpc_tenancy
- modify_vpn_connection
- modify_vpn_connection_options
- modify_vpn_tunnel_certificate

- modify_vpn_tunnel_options
- monitor_instances
- move_address_to_vpc
- move_byoip_cidr_to_ipam
- provision_byoip_cidr
- provision_ipam_pool_cidr
- provision_public_ipv4_pool_cidr
- purchase_host_reservation
- purchase_reserved_instances_offering
- purchase_scheduled_instances
- reboot_instances
- register_image
- register_instance_event_notification_attributes
- register_transit_gateway_multicast_group_members
- register_transit_gateway_multicast_group_sources
- reject_transit_gateway_multicast_domain_associations
- reject_transit_gateway_peering_attachment
- reject_transit_gateway_vpc_attachment
- reject_vpc_endpoint_connections
- reject_vpc_peering_connection
- release_address
- release_hosts
- release_ipam_pool_allocation
- replace_iam_instance_profile_association
- replace_network_acl_association
- replace_network_acl_entry
- replace_route
- replace_route_table_association
- replace_transit_gateway_route
- report_instance_status
- request_spot_fleet
- request_spot_instances
- reset_address_attribute
- reset_ebs_default_kms_key_id
- reset_fpga_image_attribute
- reset_image_attribute

- [] reset_instance_attribute
- [] reset_network_interface_attribute
- [] reset_snapshot_attribute
- [] restore_address_to_classic
- [] restore_image_from_recycle_bin
- [] restore_managed_prefix_list_version
- [] restore_snapshot_from_recycle_bin
- [] restore_snapshot_tier
- [] revoke_client_vpn_ingress
- [X] revoke_security_group_egress
- [X] revoke_security_group_ingress
- [X] run_instances

The Placement-parameter is validated to verify the availability-zone exists for the current region.

The InstanceType-parameter can be validated, to see if it is a known instance-type. Enable this validation by setting the environment variable *MOTO_EC2_ENABLE_INSTANCE_TYPE_VALIDATION=true*

The ImageId-parameter can be validated, to see if it is a known AMI. Enable this validation by setting the environment variable *MOTO_ENABLE_AMI_VALIDATION=true*

The KeyPair-parameter can be validated, to see if it is a known key-pair. Enable this validation by setting the environment variable *MOTO_ENABLE_KEYPAIR_VALIDATION=true*

- [] run_scheduled_instances
- [] search_local_gateway_routes
- [] search_transit_gateway_multicast_groups
- [X] search_transit_gateway_routes

The following filters are currently supported: type, state, route-search.exact-match

- [] send_diagnostic_interrupt
- [X] start_instances
- [] start_network_insights_access_scope_analysis
- [] start_network_insights_analysis
- [] start_vpc_endpoint_service_private_dns_verification
- [X] stop_instances
- [] terminate_client_vpn_connections
- [X] terminate_instances
- [X] unassign_ipv6_addresses
- [X] unassign_private_ip_addresses
- [] unassign_private_nat_gateway_address
- [] unmonitor_instances

- [X] update_security_group_rule_descriptions_egress
- [X] update_security_group_rule_descriptions_ingress
- [] withdraw_byoip_cidr

2.8.34 ec2-instance-connect

Example usage

```
@mock_ec2instanceconnect
def test_ec2instanceconnect_behaviour:
    boto3.client("ec2-instance-connect")
    ...
```

Implemented features for this service

- [] send_serial_console_ssh_public_key
- [X] send_ssh_public_key

2.8.35 ecr

Example usage

```
@mock_ecr
def test_ecr_behaviour:
    boto3.client("ecr")
    ...
```

Implemented features for this service

- [] batch_check_layer_availability
- [X] batch_delete_image
- [X] batch_get_image
 - The parameter AcceptedMediaTypes has not yet been implemented
- [] batch_get_repository_scanning_configuration
- [] complete_layer_upload
- [] create_pull_through_cache_rule
- [X] create_repository
- [X] delete_lifecycle_policy
- [] delete_pull_through_cache_rule
- [X] delete_registry_policy
- [X] delete_repository
- [X] delete_repository_policy
- [] describe_image_replication_status
- [X] describe_image_scan_findings
- [X] describe_images

- [] describe_pull_through_cache_rules
- [X] describe_registry
- [X] describe_repositories
 - maxResults and nextToken not implemented
- [] get_authorization_token
- [] get_download_url_for_layer
- [X] get_lifecycle_policy
- [] get_lifecycle_policy_preview
- [X] get_registry_policy
- [] get_registry_scanning_configuration
- [X] get_repository_policy
- [] initiate_layer_upload
- [X] list_images
 - maxResults and filtering not implemented
- [X] list_tags_for_resource
- [X] put_image
- [X] put_image_scanning_configuration
- [X] put_image_tag_mutability
- [X] put_lifecycle_policy
- [X] put_registry_policy
- [] put_registry_scanning_configuration
- [X] put_replication_configuration
- [X] set_repository_policy
- [X] start_image_scan
- [] start_lifecycle_policy_preview
- [X] tag_resource
- [X] untag_resource
- [] upload_layer_part

2.8.36 ecs

class moto.ecs.models.**EC2ContainerServiceBackend**(*region_name: str, account_id: str*)

ECS resources use the new ARN format by default. Use the following environment variable to revert back to the old/short ARN format: `MOTO_ECS_NEW_ARN=false`

AWS reference: <https://aws.amazon.com/blogs/compute/migrating-your-amazon-ecs-deployment-to-the-new-arn-and-resource-i>

Example usage

```
@mock_ecs
def test_ecs_behaviour:
    boto3.client("ecs")
    ...
```

Implemented features for this service

- [X] create_capacity_provider
- [X] create_cluster
- [X] create_service
- [X] create_task_set
- [X] delete_account_setting
- [X] delete_attributes
- [X] delete_capacity_provider
- [X] delete_cluster
- [X] delete_service
- [X] delete_task_set

The Force-parameter is not yet implemented

- [X] deregister_container_instance
- [X] deregister_task_definition
- [X] describe_capacity_providers
- [X] describe_clusters

Only include=TAGS is currently supported.

- [X] describe_container_instances
- [X] describe_services
- [X] describe_task_definition
- [X] describe_task_sets
- [X] describe_tasks

Only include=TAGS is currently supported.

- [] discover_poll_endpoint
- [] execute_command
- [] get_task_protection
- [X] list_account_settings
- [X] list_attributes

Pagination is not yet implemented

- [X] list_clusters

maxSize and pagination not implemented

- [X] list_container_instances

- [X] list_services
- [] list_services_by_namespace
- [X] list_tags_for_resource Currently implemented only for task definitions and services
- [X] list_task_definition_families

The Status and pagination parameters are not yet implemented

- [X] list_task_definitions
- [X] list_tasks
- [X] put_account_setting
- [] put_account_setting_default
- [X] put_attributes
- [X] put_cluster_capacity_providers
- [X] register_container_instance
- [X] register_task_definition
- [X] run_task
- [X] start_task
- [X] stop_task
- [] submit_attachment_state_changes
- [] submit_container_state_change
- [] submit_task_state_change
- [X] tag_resource
- [X] untag_resource
- [X] update_capacity_provider
- [X] update_cluster

The serviceConnectDefaults-parameter is not yet implemented

- [] update_cluster_settings
- [] update_container_agent
- [X] update_container_instances_state
- [X] update_service
- [X] update_service_primary_task_set Updates task sets be PRIMARY or ACTIVE for given cluster:service task sets
- [] update_task_protection
- [X] update_task_set

2.8.37 efs

class moto.efs.models.EFSBackend(*region_name*, *account_id*)

The backend manager of EFS resources.

This is the state-machine for each region, tracking the file systems, mount targets, and eventually access points that are deployed. Creating, updating, and destroying such resources should always go through this class.

Example usage

```
@mock_efs
def test_efs_behaviour:
    boto3.client("efs")
    ...
```

Implemented features for this service

- [X] create_access_point
- [X] create_file_system Create a new EFS File System Volume.
https://docs.aws.amazon.com/efs/latest/ug/API_CreateFileSystem.html
- [X] create_mount_target Create a new EFS Mount Target for a given File System to a given subnet.
Note that you can only create one mount target for each availability zone (which is implied by the subnet ID).
https://docs.aws.amazon.com/efs/latest/ug/API_CreateMountTarget.html
- [] create_replication_configuration
- [] create_tags
- [X] delete_access_point
- [X] delete_file_system Delete the file system specified by the given file_system_id.
Note that mount targets must be deleted first.
https://docs.aws.amazon.com/efs/latest/ug/API_DeleteFileSystem.html
- [] delete_file_system_policy
- [X] delete_mount_target Delete a mount target specified by the given mount_target_id.
Note that this will also delete a network interface.
https://docs.aws.amazon.com/efs/latest/ug/API_DeleteMountTarget.html
- [] delete_replication_configuration
- [] delete_tags
- [X] describe_access_points
Pagination is not yet implemented
- [] describe_account_preferences
- [X] describe_backup_policy
- [] describe_file_system_policy
- [X] describe_file_systems Describe all the EFS File Systems, or specific File Systems.
https://docs.aws.amazon.com/efs/latest/ug/API_DescribeFileSystems.html

- [X] describe_lifecycle_configuration
- [X] describe_mount_target_security_groups
- [X] describe_mount_targets Describe the mount targets given an access point ID, mount target ID or a file system ID.

https://docs.aws.amazon.com/efs/latest/ug/API_DescribeMountTargets.html

- [] describe_replication_configurations
- [] describe_tags
- [X] list_tags_for_resource
- [X] modify_mount_target_security_groups
- [] put_account_preferences
- [] put_backup_policy
- [] put_file_system_policy
- [X] put_lifecycle_configuration
- [X] tag_resource
- [X] untag_resource
- [] update_file_system

2.8.38 eks

Example usage

```
@mock_eks
def test_eks_behaviour:
    boto3.client("eks")
    ...
```

Implemented features for this service

- [] associate_encryption_config
- [] associate_identity_provider_config
- [] create_addon
- [X] create_cluster
- [X] create_fargate_profile
- [X] create_nodegroup
- [] delete_addon
- [X] delete_cluster
- [X] delete_fargate_profile
- [X] delete_nodegroup
- [] deregister_cluster
- [] describe_addon
- [] describe_addon_configuration

- [] describe_addon_versions
- [X] describe_cluster
- [X] describe_fargate_profile
- [] describe_identity_provider_config
- [X] describe_nodegroup
- [] describe_update
- [] disassociate_identity_provider_config
- [] list_addons
- [X] list_clusters
- [X] list_fargate_profiles
- [] list_identity_provider_configs
- [X] list_nodegroups
- [X] list_tags_for_resource

This function currently will list tags on an EKS cluster only. It does not list tags from a managed node group

- [] list_updates
- [] register_cluster
- [X] tag_resource

This function currently will tag an EKS cluster only. It does not tag a managed node group

- [X] untag_resource

This function currently will remove tags on an EKS cluster only. It does not remove tags from a managed node group

- [] update_addon
- [] update_cluster_config
- [] update_cluster_version
- [] update_nodegroup_config
- [] update_nodegroup_version

2.8.39 elasticache

class moto.elasticache.models.ElasticCacheBackend(*region_name, account_id*)

Implementation of ElastiCache APIs.

Example usage

```
@mock_elasticache
def test_elasticache_behaviour():
    boto3.client("elasticache")
    ...
```

Implemented features for this service

- add_tags_to_resource
- authorize_cache_security_group_ingress
- batch_apply_update_action
- batch_stop_update_action
- complete_migration
- copy_snapshot
- create_cache_cluster
- create_cache_parameter_group
- create_cache_security_group
- create_cache_subnet_group
- create_global_replication_group
- create_replication_group
- create_snapshot
- create_user
- create_user_group
- decrease_node_groups_in_global_replication_group
- decrease_replica_count
- delete_cache_cluster
- delete_cache_parameter_group
- delete_cache_security_group
- delete_cache_subnet_group
- delete_global_replication_group
- delete_replication_group
- delete_snapshot
- delete_user
- delete_user_group
- describe_cache_clusters
- describe_cache_engine_versions
- describe_cache_parameter_groups
- describe_cache_parameters
- describe_cache_security_groups
- describe_cache_subnet_groups
- describe_engine_default_parameters
- describe_events
- describe_global_replication_groups
- describe_replication_groups

- [] describe_reserved_cache_nodes
- [] describe_reserved_cache_nodes_offerings
- [] describe_service_updates
- [] describe_snapshots
- [] describe_update_actions
- [] describe_user_groups
- [X] describe_users

Only the *user_id* parameter is currently supported. Pagination is not yet implemented.

- [] disassociate_global_replication_group
- [] failover_global_replication_group
- [] increase_node_groups_in_global_replication_group
- [] increase_replica_count
- [] list_allowed_node_type_modifications
- [] list_tags_for_resource
- [] modify_cache_cluster
- [] modify_cache_parameter_group
- [] modify_cache_subnet_group
- [] modify_global_replication_group
- [] modify_replication_group
- [] modify_replication_group_shard_configuration
- [] modify_user
- [] modify_user_group
- [] purchase_reserved_cache_nodes_offering
- [] rebalance_slots_in_global_replication_group
- [] reboot_cache_cluster
- [] remove_tags_from_resource
- [] reset_cache_parameter_group
- [] revoke_cache_security_group_ingress
- [] start_migration
- [] test_failover

2.8.40 elasticbeanstalk

Example usage

```
@mock_elasticbeanstalk
def test_elasticbeanstalk_behaviour:
    boto3.client("elasticbeanstalk")
    ...
```

Implemented features for this service

- [] abort_environment_update
- [] apply_environment_managed_action
- [] associate_environment_operations_role
- [] check_dns_availability
- [] compose_environments
- [X] create_application
- [] create_application_version
- [] create_configuration_template
- [X] create_environment
- [] create_platform_version
- [] create_storage_location
- [] delete_application
- [] delete_application_version
- [] delete_configuration_template
- [] delete_environment_configuration
- [] delete_platform_version
- [] describe_account_attributes
- [] describe_application_versions
- [] describe_applications
- [] describe_configuration_options
- [] describe_configuration_settings
- [] describe_environment_health
- [] describe_environment_managed_action_history
- [] describe_environment_managed_actions
- [] describe_environment_resources
- [X] describe_environments
- [] describe_events
- [] describe_instances_health
- [] describe_platform_version

- [] disassociate_environment_operations_role
- [X] list_available_solution_stacks
- [] list_platform_branches
- [] list_platform_versions
- [X] list_tags_for_resource
- [] rebuild_environment
- [] request_environment_info
- [] restart_app_server
- [] retrieve_environment_info
- [] swap_environment_cnames
- [] terminate_environment
- [] update_application
- [] update_application_resource_lifecycle
- [] update_application_version
- [] update_configuration_template
- [] update_environment
- [X] update_tags_for_resource
- [] validate_configuration_settings

2.8.41 elastictranscoder

Example usage

```
@mock_elastictranscoder
def test_elastictranscoder_behaviour:
    boto3.client("elastictranscoder")
    ...
```

Implemented features for this service

- [] cancel_job
- [] create_job
- [X] create_pipeline

The following parameters are not yet implemented: AWSKMSKeyArn, Notifications

- [] create_preset
- [X] delete_pipeline
- [] delete_preset
- [] list_jobs_by_pipeline
- [] list_jobs_by_status
- [X] list_pipelines

- [] list_presets
- [] read_job
- [X] read_pipeline
- [] read_preset
- [] test_role
- [X] update_pipeline

The following parameters are not yet implemented: AWSKMSKeyArn, Notifications, ContentConfig, ThumbnailConfig

- [] update_pipeline_notifications
- [] update_pipeline_status

2.8.42 elb

Example usage

```
@mock_elb
def test_elb_behaviour:
    boto3.client("elb")
    ...
```

Implemented features for this service

- [] add_tags
- [X] apply_security_groups_to_load_balancer
- [X] attach_load_balancer_to_subnets
- [X] configure_health_check
- [X] create_app_cookie_stickiness_policy
- [X] create_lb_cookie_stickiness_policy
- [X] create_load_balancer
- [X] create_load_balancer_listeners
- [] create_load_balancer_policy
- [X] delete_load_balancer
- [X] delete_load_balancer_listeners
- [X] delete_load_balancer_policy
- [] deregister_instances_from_load_balancer
- [] describe_account_limits
- [X] describe_instance_health
- [] describe_load_balancer_attributes
- [X] describe_load_balancer_policies
- [] describe_load_balancer_policy_types
- [X] describe_load_balancers

- [] describe_tags
- [X] detach_load_balancer_from_subnets
- [X] disable_availability_zones_for_load_balancer
- [X] enable_availability_zones_for_load_balancer
- [X] modify_load_balancer_attributes
- [] register_instances_with_load_balancer
- [] remove_tags
- [X] set_load_balancer_listener_ssl_certificate
- [] set_load_balancer_policies_for_backend_server
- [X] set_load_balancer_policies_of_listener

2.8.43 elbv2

Example usage

```
@mock_elbv2
def test_elbv2_behaviour:
    boto3.client("elbv2")
    ...
```

Implemented features for this service

- [X] add_listener_certificates
- [X] add_tags
- [X] create_listener
- [X] create_load_balancer
- [X] create_rule
- [X] create_target_group
- [X] delete_listener
- [X] delete_load_balancer
- [X] delete_rule
- [X] delete_target_group
- [X] deregister_targets
- [] describe_account_limits
- [X] describe_listener_certificates
- [X] describe_listeners
- [X] describe_load_balancer_attributes
- [X] describe_load_balancers
- [X] describe_rules
- [] describe_ssl_policies

- [X] describe_tags
- [] describe_target_group_attributes
- [X] describe_target_groups
- [X] describe_target_health
- [X] modify_listener
- [X] modify_load_balancer_attributes
- [X] modify_rule
- [X] modify_target_group
- [X] modify_target_group_attributes
- [X] register_targets
- [X] remove_listener_certificates
- [X] remove_tags
- [X] set_ip_address_type
- [X] set_rule_priorities
- [X] set_security_groups
- [X] set_subnets

2.8.44 emr

Example usage

```
@mock_emr
def test_emr_behaviour:
    boto3.client("emr")
    ...
```

Implemented features for this service

- [] add_instance_fleet
- [X] add_instance_groups
- [X] add_job_flow_steps
- [X] add_tags
- [] cancel_steps
- [X] create_security_configuration
- [] create_studio
- [] create_studio_session_mapping
- [X] delete_security_configuration
- [] delete_studio
- [] delete_studio_session_mapping
- [X] describe_cluster

- describe_job_flows
- describe_notebook_execution
- describe_release_label
- describe_security_configuration
- describe_step
- describe_studio
- get_auto_termination_policy
- get_block_public_access_configuration
- get_cluster_session_credentials
- get_managed_scaling_policy
- get_studio_session_mapping
- list_bootstrap_actions
- list_clusters
- list_instance_fleets
- list_instance_groups
- list_instances
- list_notebook_executions
- list_release_labels
- list_security_configurations
- list_steps
- list_studio_session_mappings
- list_studios
- modify_cluster
- modify_instance_fleet
- modify_instance_groups
- put_auto_scaling_policy
- put_auto_termination_policy
- put_block_public_access_configuration
- put_managed_scaling_policy
- remove_auto_scaling_policy
- remove_auto_termination_policy
- remove_managed_scaling_policy
- remove_tags
- run_job_flow
- set_termination_protection
- set_visible_to_all_users

- [] start_notebook_execution
- [] stop_notebook_execution
- [X] terminate_job_flows
- [] update_studio
- [] update_studio_session_mapping

2.8.45 emr-containers

class moto.emrcontainers.models.**EMRContainersBackend**(*region_name*, *account_id*)
Implementation of EMRContainers APIs.

Example usage

```
@mock_emrcontainers
def test_emrcontainers_behaviour:
    boto3.client("emr-containers")
    ...
```

Implemented features for this service

- [X] cancel_job_run
- [] create_job_template
- [] create_managed_endpoint
- [X] create_virtual_cluster
- [] delete_job_template
- [] delete_managed_endpoint
- [X] delete_virtual_cluster
- [X] describe_job_run
- [] describe_job_template
- [] describe_managed_endpoint
- [X] describe_virtual_cluster
- [X] list_job_runs
- [] list_job_templates
- [] list_managed_endpoints
- [] list_tags_for_resource
- [X] list_virtual_clusters
- [X] start_job_run
- [] tag_resource
- [] untag_resource

2.8.46 emr-serverless

class moto.emrserverless.models.**EMRServerlessBackend**(*region_name, account_id*)
Implementation of EMRServerless APIs.

Example usage

```
@mock_emrserverless
def test_emrserverless_behaviour:
    boto3.client("emr-serverless")
    ...
```

Implemented features for this service

- [] cancel_job_run
- [X] create_application
- [X] delete_application
- [X] get_application
- [] get_dashboard_for_job_run
- [] get_job_run
- [X] list_applications
- [] list_job_runs
- [] list_tags_for_resource
- [X] start_application
- [] start_job_run
- [X] stop_application
- [] tag_resource
- [] untag_resource
- [X] update_application

2.8.47 es

class moto.es.models.**ElasticsearchServiceBackend**(*region_name, account_id*)
Implementation of ElasticsearchService APIs.

Example usage

```
@mock_es
def test_es_behaviour:
    boto3.client("es")
    ...
```

Implemented features for this service

- [] accept_inbound_cross_cluster_search_connection
- [] add_tags
- [] associate_package

- [] authorize_vpc_endpoint_access
- [] cancel_elasticsearch_service_software_update
- [X] create_elasticsearch_domain
- [] create_outbound_cross_cluster_search_connection
- [] create_package
- [] create_vpc_endpoint
- [X] delete_elasticsearch_domain
- [] delete_elasticsearch_service_role
- [] delete_inbound_cross_cluster_search_connection
- [] delete_outbound_cross_cluster_search_connection
- [] delete_package
- [] delete_vpc_endpoint
- [] describe_domain_auto_tunes
- [] describe_domain_change_progress
- [X] describe_elasticsearch_domain
- [] describe_elasticsearch_domain_config
- [] describe_elasticsearch_domains
- [] describe_elasticsearch_instance_type_limits
- [] describe_inbound_cross_cluster_search_connections
- [] describe_outbound_cross_cluster_search_connections
- [] describe_packages
- [] describe_reserved_elasticsearch_instance_offerings
- [] describe_reserved_elasticsearch_instances
- [] describe_vpc_endpoints
- [] dissociate_package
- [] get_compatible_elasticsearch_versions
- [] get_package_version_history
- [] get_upgrade_history
- [] get_upgrade_status
- [X] list_domain_names

The engine-type parameter is not yet supported. Pagination is not yet implemented.

- [] list_domains_for_package
- [] list_elasticsearch_instance_types
- [] list_elasticsearch_versions
- [] list_packages_for_domain
- [] list_tags

- [] list_vpc_endpoint_access
- [] list_vpc_endpoints
- [] list_vpc_endpoints_for_domain
- [] purchase_reserved_elasticsearch_instance_offering
- [] reject_inbound_cross_cluster_search_connection
- [] remove_tags
- [] revoke_vpc_endpoint_access
- [] start_elasticsearch_service_software_update
- [] update_elasticsearch_domain_config
- [] update_package
- [] update_vpc_endpoint
- [] upgrade_elasticsearch_domain

2.8.48 events

class moto.events.models.**EventsBackend**(*region_name*, *account_id*)

Some Moto services are configured to generate events and send them to EventBridge. See the AWS documentation here: <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-service-event.html>

Events that currently supported

- S3:CreateBucket

Targets that are currently supported

- AWSLambda functions

Please let us know if you want support for an event/target that is not yet listed here.

Example usage

```
@mock_events
def test_events_behaviour:
    boto3.client("events")
    ...
```

Implemented features for this service

- [] activate_event_source
- [X] cancel_replay
- [X] create_api_destination

Creates an API destination, which is an HTTP invocation endpoint configured as a target for events.

Docs: https://docs.aws.amazon.com/eventbridge/latest/APIReference/API_CreateApiDestination.html

Returns: dict

- [X] create_archive
- [X] create_connection

- [] create_endpoint
- [X] create_event_bus
- [] create_partner_event_source
- [] deactivate_event_source
- [] deauthorize_connection
- [X] delete_api_destination

Deletes the specified API destination.

Docs: https://docs.aws.amazon.com/eventbridge/latest/APIReference/API_DeleteApiDestination.html

Args: name: The name of the destination to delete.

Raises: ResourceNotFoundException: When the destination is not present.

Returns: dict

- [X] delete_archive
- [X] delete_connection

Deletes a connection.

Docs: https://docs.aws.amazon.com/eventbridge/latest/APIReference/API_DeleteConnection.html

Args: name: The name of the connection to delete.

Raises: ResourceNotFoundException: When the connection is not present.

Returns: dict

- [] delete_endpoint
- [X] delete_event_bus
- [] delete_partner_event_source
- [X] delete_rule
- [X] describe_api_destination

Retrieves details about an API destination.

Docs: https://docs.aws.amazon.com/eventbridge/latest/APIReference/API_DescribeApiDestination.html

Args: name: The name of the API destination to retrieve.

Returns: dict

- [X] describe_archive
- [X] describe_connection

Retrieves details about a connection.

Docs: https://docs.aws.amazon.com/eventbridge/latest/APIReference/API_DescribeConnection.html

Args: name: The name of the connection to retrieve.

Raises: ResourceNotFoundException: When the connection is not present.

Returns: dict

- [] describe_endpoint
- [X] describe_event_bus
- [] describe_event_source
- [] describe_partner_event_source
- [X] describe_replay
- [X] describe_rule
- [X] disable_rule
- [X] enable_rule
- [X] list_api_destinations
- [X] list_archives
- [X] list_connections
- [] list_endpoints
- [X] list_event_buses
- [] list_event_sources
- [] list_partner_event_source_accounts
- [] list_partner_event_sources
- [X] list_replays
- [X] list_rule_names_by_target
- [X] list_rules
- [X] list_tags_for_resource
- [X] list_targets_by_rule
- [X] put_events

The following targets are supported at the moment:

- CloudWatch Log Group
 - EventBridge Archive
 - SQS Queue + FIFO Queue
 - Cross-region/account EventBus
- [] put_partner_events
 - [X] put_permission
 - [X] put_rule
 - [X] put_targets
 - [X] remove_permission
 - [X] remove_targets
 - [X] start_replay
 - [X] tag_resource
 - [X] test_event_pattern

- [X] `untag_resource`
- [X] `update_api_destination`

Creates an API destination, which is an HTTP invocation endpoint configured as a target for events.

Docs: https://docs.aws.amazon.com/eventbridge/latest/APIReference/API_UpdateApiDestination.html

Returns: dict
- [X] `update_archive`
- [X] `update_connection`
- [] `update_endpoint`

2.8.49 firehose

class `moto.firehose.models.FirehoseBackend`(*region_name*, *account_id*)
Implementation of Firehose APIs.

Example usage

```
@mock_firehose
def test_firehose_behaviour:
    boto3.client("firehose")
    ...
```

Implemented features for this service

- [X] `create_delivery_stream` Create a Kinesis Data Firehose delivery stream.
- [X] `delete_delivery_stream` Delete a delivery stream and its data.
 - AllowForceDelete option is ignored as we only superficially apply state.
- [X] `describe_delivery_stream` Return description of specified delivery stream and its status.
 - Note: the 'limit' and 'exclusive_start_destination_id' parameters are not currently processed/implemented.
- [X] `list_delivery_streams` Return list of delivery streams in alphabetic order of names.
- [X] `list_tags_for_delivery_stream` Return list of tags.
- [X] `put_record` Write a single data record into a Kinesis Data firehose stream.
- [X] `put_record_batch` Write multiple data records into a Kinesis Data firehose stream.
- [] `start_delivery_stream_encryption`
- [] `stop_delivery_stream_encryption`
- [X] `tag_delivery_stream` Add/update tags for specified delivery stream.
- [X] `untag_delivery_stream` Removes tags from specified delivery stream.
- [X] `update_destination` Updates specified destination of specified delivery stream.

2.8.50 forecast

Example usage

```
@mock_forecast
def test_forecast_behaviour:
    boto3.client("forecast")
    ...
```

Implemented features for this service

- [] create_auto_predictor
- [] create_dataset
- [X] create_dataset_group
- [] create_dataset_import_job
- [] create_explainability
- [] create_explainability_export
- [] create_forecast
- [] create_forecast_export_job
- [] create_monitor
- [] create_predictor
- [] create_predictor_backtest_export_job
- [] create_what_if_analysis
- [] create_what_if_forecast
- [] create_what_if_forecast_export
- [] delete_dataset
- [X] delete_dataset_group
- [] delete_dataset_import_job
- [] delete_explainability
- [] delete_explainability_export
- [] delete_forecast
- [] delete_forecast_export_job
- [] delete_monitor
- [] delete_predictor
- [] delete_predictor_backtest_export_job
- [] delete_resource_tree
- [] delete_what_if_analysis
- [] delete_what_if_forecast
- [] delete_what_if_forecast_export
- [] describe_auto_predictor

- describe_dataset
- describe_dataset_group
- describe_dataset_import_job
- describe_explainability
- describe_explainability_export
- describe_forecast
- describe_forecast_export_job
- describe_monitor
- describe_predictor
- describe_predictor_backtest_export_job
- describe_what_if_analysis
- describe_what_if_forecast
- describe_what_if_forecast_export
- get_accuracy_metrics
- list_dataset_groups
- list_dataset_import_jobs
- list_datasets
- list_explainabilities
- list_explainability_exports
- list_forecast_export_jobs
- list_forecasts
- list_monitor_evaluations
- list_monitors
- list_predictor_backtest_export_jobs
- list_predictors
- list_tags_for_resource
- list_what_if_analyses
- list_what_if_forecast_exports
- list_what_if_forecasts
- resume_resource
- stop_resource
- tag_resource
- untag_resource
- update_dataset_group

2.8.51 glacier

Example usage

```
@mock_glacier
def test_glacier_behaviour:
    boto3.client("glacier")
    ...
```

Implemented features for this service

- [] abort_multipart_upload
- [] abort_vault_lock
- [] add_tags_to_vault
- [] complete_multipart_upload
- [] complete_vault_lock
- [X] create_vault
- [] delete_archive
- [X] delete_vault
- [] delete_vault_access_policy
- [] delete_vault_notifications
- [X] describe_job
- [] describe_vault
- [] get_data_retrieval_policy
- [X] get_job_output
- [] get_vault_access_policy
- [] get_vault_lock
- [] get_vault_notifications
- [X] initiate_job
- [] initiate_multipart_upload
- [] initiate_vault_lock
- [X] list_jobs
- [] list_multipart_uploads
- [] list_parts
- [] list_provisioned_capacity
- [] list_tags_for_vault
- [X] list_vaults
- [] purchase_provisioned_capacity
- [] remove_tags_from_vault
- [] set_data_retrieval_policy

- [] set_vault_access_policy
- [] set_vault_notifications
- [X] upload_archive
- [] upload_multipart_part

2.8.52 glue

Example usage

```
@mock_glue
def test_glue_behaviour:
    boto3.client("glue")
    ...
```

Implemented features for this service

- [X] batch_create_partition
- [] batch_delete_connection
- [X] batch_delete_partition
- [X] batch_delete_table
- [] batch_delete_table_version
- [] batch_get_blueprints
- [X] batch_get_crawlers
- [] batch_get_custom_entity_types
- [] batch_get_data_quality_result
- [] batch_get_dev_endpoints
- [] batch_get_jobs
- [X] batch_get_partition
- [] batch_get_triggers
- [] batch_get_workflows
- [] batch_stop_job_run
- [X] batch_update_partition
- [] cancel_data_quality_rule_recommendation_run
- [] cancel_data_quality_ruleset_evaluation_run
- [] cancel_ml_task_run
- [] cancel_statement
- [] check_schema_version_validity
- [] create_blueprint
- [] create_classifier
- [] create_connection

- create_crawler
- create_custom_entity_type
- create_data_quality_ruleset
- create_database
- create_dev_endpoint
- create_job
- create_ml_transform
- create_partition
- create_partition_index
- create_registry
- create_schema

The following parameters/features are not yet implemented: Glue Schema Registry: compatibility checks NONE | BACKWARD | BACKWARD_ALL | FORWARD | FORWARD_ALL | FULL | FULL_ALL and Data format parsing and syntax validation.

- create_script
- create_security_configuration
- create_session
- create_table
- create_trigger
- create_user_defined_function
- create_workflow
- delete_blueprint
- delete_classifier
- delete_column_statistics_for_partition
- delete_column_statistics_for_table
- delete_connection
- delete_crawler
- delete_custom_entity_type
- delete_data_quality_ruleset
- delete_database
- delete_dev_endpoint
- delete_job
- delete_ml_transform
- delete_partition
- delete_partition_index
- delete_registry
- delete_resource_policy

- delete_schema
- delete_schema_versions
- delete_security_configuration
- delete_session
- delete_table
- delete_table_version
- delete_trigger
- delete_user_defined_function
- delete_workflow
- get_blueprint
- get_blueprint_run
- get_blueprint_runs
- get_catalog_import_status
- get_classifier
- get_classifiers
- get_column_statistics_for_partition
- get_column_statistics_for_table
- get_connection
- get_connections
- get_crawler
- get_crawler_metrics
- get_crawlers
- get_custom_entity_type
- get_data_catalog_encryption_settings
- get_data_quality_result
- get_data_quality_rule_recommendation_run
- get_data_quality_ruleset
- get_data_quality_ruleset_evaluation_run
- get_database
- get_databases
- get_dataflow_graph
- get_dev_endpoint
- get_dev_endpoints
- get_job
- get_job_bookmark
- get_job_run

- [] `get_job_runs`
- [] `get_jobs`
- [] `get_mapping`
- [] `get_ml_task_run`
- [] `get_ml_task_runs`
- [] `get_ml_transform`
- [] `get_ml_transforms`
- [] `get_partition`
- [] `get_partition_indexes`
- [X] `get_partitions`

See https://docs.aws.amazon.com/glue/latest/webapi/API_GetPartitions.html for supported expressions.

Expression caveats:

- Column names must consist of UPPERCASE, lowercase, dots and underscores only.
- Nanosecond expressions on timestamp columns are rounded to microseconds.
- Literal dates and timestamps must be valid, i.e. no support for February 31st.
- LIKE expressions are converted to Python regexes, escaping special characters. Only % and _ wildcards are supported, and SQL escaping using [] does not work.

- [] `get_plan`
- [X] `get_registry`
- [] `get_resource_policies`
- [] `get_resource_policy`
- [X] `get_schema`
- [X] `get_schema_by_definition`
- [X] `get_schema_version`
- [] `get_schema_versions_diff`
- [] `get_security_configuration`
- [] `get_security_configurations`
- [] `get_session`
- [] `get_statement`
- [X] `get_table`
- [] `get_table_version`
- [] `get_table_versions`
- [X] `get_tables`
- [X] `get_tags`
- [] `get_trigger`
- [] `get_triggers`

- [] `get_unfiltered_partition_metadata`
- [] `get_unfiltered_partitions_metadata`
- [] `get_unfiltered_table_metadata`
- [] `get_user_defined_function`
- [] `get_user_defined_functions`
- [] `get_workflow`
- [] `get_workflow_run`
- [] `get_workflow_run_properties`
- [] `get_workflow_runs`
- [] `import_catalog_to_glue`
- [] `list_blueprints`
- [X] `list_crawlers`
- [] `list_crawls`
- [] `list_custom_entity_types`
- [] `list_data_quality_results`
- [] `list_data_quality_rule_recommendation_runs`
- [] `list_data_quality_ruleset_evaluation_runs`
- [] `list_data_quality_rulesets`
- [] `list_dev_endpoints`
- [X] `list_jobs`
- [] `list_ml_transforms`
- [X] `list_registries`
- [] `list_schema_versions`
- [] `list_schemas`
- [] `list_sessions`
- [] `list_statements`
- [] `list_triggers`
- [] `list_workflows`
- [] `put_data_catalog_encryption_settings`
- [] `put_resource_policy`
- [X] `put_schema_version_metadata`
- [] `put_workflow_run_properties`
- [] `query_schema_version_metadata`
- [X] `register_schema_version`
- [] `remove_schema_version_metadata`
- [] `reset_job_bookmark`

- [] resume_workflow_run
- [] run_statement
- [] search_tables
- [] start_blueprint_run
- [X] start_crawler
- [] start_crawler_schedule
- [] start_data_quality_rule_recommendation_run
- [] start_data_quality_ruleset_evaluation_run
- [] start_export_labels_task_run
- [] start_import_labels_task_run
- [X] start_job_run
- [] start_ml_evaluation_task_run
- [] start_ml_labeling_set_generation_task_run
- [] start_trigger
- [] start_workflow_run
- [X] stop_crawler
- [] stop_crawler_schedule
- [] stop_session
- [] stop_trigger
- [] stop_workflow_run
- [X] tag_resource
- [X] untag_resource
- [] update_blueprint
- [] update_classifier
- [] update_column_statistics_for_partition
- [] update_column_statistics_for_table
- [] update_connection
- [] update_crawler
- [] update_crawler_schedule
- [] update_data_quality_ruleset
- [X] update_database
- [] update_dev_endpoint
- [] update_job
- [] update_job_from_source_control
- [] update_ml_transform
- [] update_partition

- [] update_registry
- [X] update_schema

The SchemaVersionNumber-argument is not yet implemented

- [] update_source_control_from_job
- [] update_table
- [] update_trigger
- [] update_user_defined_function
- [] update_workflow

2.8.53 greengrass

Example usage

```
@mock_greengrass
def test_greengrass_behaviour:
    boto3.client("greengrass")
    ...
```

Implemented features for this service

- [X] associate_role_to_group
- [] associate_service_role_to_account
- [] create_connector_definition
- [] create_connector_definition_version
- [X] create_core_definition
- [X] create_core_definition_version
- [X] create_deployment
- [X] create_device_definition
- [X] create_device_definition_version
- [X] create_function_definition
- [X] create_function_definition_version
- [X] create_group
- [] create_group_certificate_authority
- [X] create_group_version
- [] create_logger_definition
- [] create_logger_definition_version
- [X] create_resource_definition
- [X] create_resource_definition_version
- [] create_software_update_job
- [X] create_subscription_definition

- [X] create_subscription_definition_version
- [] delete_connector_definition
- [X] delete_core_definition
- [X] delete_device_definition
- [X] delete_function_definition
- [X] delete_group
- [] delete_logger_definition
- [X] delete_resource_definition
- [X] delete_subscription_definition
- [X] disassociate_role_from_group
- [] disassociate_service_role_from_account
- [X] get_associated_role
- [] get_bulk_deployment_status
- [] get_connectivity_info
- [] get_connector_definition
- [] get_connector_definition_version
- [X] get_core_definition
- [X] get_core_definition_version
- [X] get_deployment_status
- [X] get_device_definition
- [X] get_device_definition_version
- [X] get_function_definition
- [X] get_function_definition_version
- [X] get_group
- [] get_group_certificate_authority
- [] get_group_certificate_configuration
- [X] get_group_version
- [] get_logger_definition
- [] get_logger_definition_version
- [X] get_resource_definition
- [X] get_resource_definition_version
- [] get_service_role_for_account
- [X] get_subscription_definition
- [X] get_subscription_definition_version
- [] get_thing_runtime_configuration
- [] list_bulk_deployment_detailed_reports

- list_bulk_deployments
- list_connector_definition_versions
- list_connector_definitions
- list_core_definition_versions
- list_core_definitions
- list_deployments
- list_device_definition_versions
- list_device_definitions
- list_function_definition_versions
- list_function_definitions
- list_group_certificate_authorities
- list_group_versions
- list_groups
- list_logger_definition_versions
- list_logger_definitions
- list_resource_definition_versions
- list_resource_definitions
- list_subscription_definition_versions
- list_subscription_definitions
- list_tags_for_resource
- reset_deployments
- start_bulk_deployment
- stop_bulk_deployment
- tag_resource
- untag_resource
- update_connectivity_info
- update_connector_definition
- update_core_definition
- update_device_definition
- update_function_definition
- update_group
- update_group_certificate_configuration
- update_logger_definition
- update_resource_definition
- update_subscription_definition
- update_thing_runtime_configuration

2.8.54 guardduty

Example usage

```
@mock_guardduty
def test_guardduty_behaviour:
    boto3.client("guardduty")
    ...
```

Implemented features for this service

- [] accept_administrator_invitation
- [] accept_invitation
- [] archive_findings
- [X] create_detector
- [X] create_filter
- [] create_ip_set
- [] create_members
- [] create_publishing_destination
- [] create_sample_findings
- [] create_threat_intel_set
- [] decline_invitations
- [X] delete_detector
- [X] delete_filter
- [] delete_invitations
- [] delete_ip_set
- [] delete_members
- [] delete_publishing_destination
- [] delete_threat_intel_set
- [] describe_malware_scans
- [] describe_organization_configuration
- [] describe_publishing_destination
- [] disable_organization_admin_account
- [] disassociate_from_administrator_account
- [] disassociate_from_master_account
- [] disassociate_members
- [X] enable_organization_admin_account
- [] get_administrator_account
- [X] get_detector
- [X] get_filter

- [] get_findings
- [] get_findings_statistics
- [] get_invitations_count
- [] get_ip_set
- [] get_malware_scan_settings
- [] get_master_account
- [] get_member_detectors
- [] get_members
- [] get_remaining_free_trial_days
- [] get_threat_intel_set
- [] get_usage_statistics
- [] invite_members
- [X] list_detectors

The MaxResults and NextToken-parameter have not yet been implemented.

- [] list_filters
- [] list_findings
- [] list_invitations
- [] list_ip_sets
- [] list_members
- [X] list_organization_admin_accounts

Pagination is not yet implemented

- [] list_publishing_destinations
- [] list_tags_for_resource
- [] list_threat_intel_sets
- [] start_monitoring_members
- [] stop_monitoring_members
- [] tag_resource
- [] unarchive_findings
- [] untag_resource
- [X] update_detector
- [X] update_filter
- [] update_findings_feedback
- [] update_ip_set
- [] update_malware_scan_settings
- [] update_member_detectors
- [] update_organization_configuration

- [] update_publishing_destination
- [] update_threat_intel_set

2.8.55 iam

Example usage

```
@mock_iam
def test_iam_behaviour:
    boto3.client("iam")
    ...
```

Implemented features for this service

- [] add_client_id_to_open_id_connect_provider
- [X] add_role_to_instance_profile
- [X] add_user_to_group
- [X] attach_group_policy
- [X] attach_role_policy
- [X] attach_user_policy
- [] change_password
- [X] create_access_key
- [X] create_account_alias
- [X] create_group
- [X] create_instance_profile
- [X] create_login_profile
- [X] create_open_id_connect_provider
- [X] create_policy
- [X] create_policy_version
- [X] create_role
- [X] create_saml_provider
- [X] create_service_linked_role
- [] create_service_specific_credential
- [X] create_user
- [X] create_virtual_mfa_device
- [X] deactivate_mfa_device Deactivate and detach MFA Device from user if device exists.
- [X] delete_access_key
- [X] delete_account_alias
- [X] delete_account_password_policy
- [X] delete_group

- [X] delete_group_policy
- [X] delete_instance_profile
- [X] delete_login_profile
- [X] delete_open_id_connect_provider
- [X] delete_policy
- [X] delete_policy_version
- [X] delete_role
- [X] delete_role_permissions_boundary
- [X] delete_role_policy
- [X] delete_saml_provider
- [X] delete_server_certificate
- [X] delete_service_linked_role
- [] delete_service_specific_credential
- [X] delete_signing_certificate
- [X] delete_ssh_public_key
- [X] delete_user
- [] delete_user_permissions_boundary
- [X] delete_user_policy
- [X] delete_virtual_mfa_device
- [X] detach_group_policy
- [X] detach_role_policy
- [X] detach_user_policy
- [X] enable_mfa_device Enable MFA Device for user.
- [] generate_credential_report
- [] generate_organizations_access_report
- [] generate_service_last_accessed_details
- [X] get_access_key_last_used
- [X] get_account_authorization_details
- [X] get_account_password_policy
- [X] get_account_summary
- [] get_context_keys_for_custom_policy
- [] get_context_keys_for_principal_policy
- [X] get_credential_report
- [X] get_group
 - Pagination is not yet implemented
- [X] get_group_policy

- [X] `get_instance_profile`
- [X] `get_login_profile`
- [X] `get_open_id_connect_provider`
- [] `get_organizations_access_report`
- [X] `get_policy`
- [X] `get_policy_version`
- [X] `get_role`
- [X] `get_role_policy`
- [X] `get_saml_provider`
- [X] `get_server_certificate`
- [] `get_service_last_accessed_details`
- [] `get_service_last_accessed_details_with_entities`
- [X] `get_service_linked_role_deletion_status`

This method always succeeds for now - we do not yet keep track of deletions

- [X] `get_ssh_public_key`
- [X] `get_user`
- [X] `get_user_policy`
- [X] `list_access_keys`

Pagination is not yet implemented

- [X] `list_account_aliases`
- [X] `list_attached_group_policies`
- [X] `list_attached_role_policies`
- [X] `list_attached_user_policies`
- [] `list_entities_for_policy`
- [X] `list_group_policies`

Pagination is not yet implemented

- [X] `list_groups`
- [] `list_groups_for_user`
- [] `list_instance_profile_tags`
- [] `list_instance_profiles`
- [] `list_instance_profiles_for_role`
- [] `list_mfa_device_tags`
- [X] `list_mfa_devices`
- [X] `list_open_id_connect_provider_tags`
- [X] `list_open_id_connect_providers`
- [X] `list_policies`

- list_policies_granting_service_access
- list_policy_tags
- list_policy_versions
- list_role_policies
- list_role_tags
- list_roles
- list_saml_provider_tags
- list_saml_providers
- list_server_certificate_tags
- list_server_certificates

 Pagination is not yet implemented

- list_service_specific_credentials
- list_signing_certificates
- list_ssh_public_keys
- list_user_policies
- list_user_tags
- list_users
- list_virtual_mfa_devices
- put_group_policy
- put_role_permissions_boundary
- put_role_policy
- put_user_permissions_boundary
- put_user_policy
- remove_client_id_from_open_id_connect_provider
- remove_role_from_instance_profile
- remove_user_from_group
- reset_service_specific_credential
- resync_mfa_device
- set_default_policy_version
- set_security_token_service_preferences
- simulate_custom_policy
- simulate_principal_policy
- tag_instance_profile
- tag_mfa_device
- tag_open_id_connect_provider
- tag_policy

- tag_role
- tag_saml_provider
- tag_server_certificate
- tag_user
- untag_instance_profile
- untag_mfa_device
- untag_open_id_connect_provider
- untag_policy
- untag_role
- untag_saml_provider
- untag_server_certificate
- untag_user
- update_access_key
- update_account_password_policy
- update_assume_role_policy
- update_group
- update_login_profile
- update_open_id_connect_provider_thumbprint
- update_role
- update_role_description
- update_saml_provider
- update_server_certificate
- update_service_specific_credential
- update_signing_certificate
- update_ssh_public_key
- update_user
- upload_server_certificate
- upload_signing_certificate
- upload_ssh_public_key

2.8.56 iot

Example usage

```
@mock_iot
def test_iot_behaviour:
    boto3.client("iot")
    ...
```

Implemented features for this service

- [] accept_certificate_transfer
- [] add_thing_to_billing_group
- [X] add_thing_to_thing_group
- [] associate_targets_with_job
- [X] attach_policy
- [X] attach_principal_policy
- [] attach_security_profile
- [X] attach_thing_principal
- [] cancel_audit_mitigation_actions_task
- [] cancel_audit_task
- [] cancel_certificate_transfer
- [] cancel_detect_mitigation_actions_task
- [X] cancel_job
- [X] cancel_job_execution

The parameters ExpectedVersion and StatusDetails are not yet implemented

- [] clear_default_authorizer
- [] confirm_topic_rule_destination
- [] create_audit_suppression
- [] create_authorizer
- [] create_billing_group
- [X] create_certificate_from_csr
- [] create_custom_metric
- [] create_dimension
- [X] create_domain_configuration

The ValidationCertificateArn-parameter is not yet implemented

- [] create_dynamic_thing_group
- [] create_fleet_metric
- [X] create_job
- [] create_job_template

- create_keys_and_certificate
- create_mitigation_action
- create_ota_update
- create_policy
- create_policy_version
- create_provisioning_claim
- create_provisioning_template
- create_provisioning_template_version
- create_role_alias
- create_scheduled_audit
- create_security_profile
- create_stream
- create_thing
- create_thing_group
- create_thing_type
- create_topic_rule
- create_topic_rule_destination
- delete_account_audit_configuration
- delete_audit_suppression
- delete_authorizer
- delete_billing_group
- delete_ca_certificate
- delete_certificate
- delete_custom_metric
- delete_dimension
- delete_domain_configuration
- delete_dynamic_thing_group
- delete_fleet_metric
- delete_job
- delete_job_execution
- delete_job_template
- delete_mitigation_action
- delete_ota_update
- delete_policy
- delete_policy_version
- delete_provisioning_template

- [] delete_provisioning_template_version
- [] delete_registration_code
- [] delete_role_alias
- [] delete_scheduled_audit
- [] delete_security_profile
- [] delete_stream
- [X] delete_thing

The ExpectedVersion-parameter is not yet implemented

- [X] delete_thing_group

The ExpectedVersion-parameter is not yet implemented

- [X] delete_thing_type
- [X] delete_topic_rule
- [] delete_topic_rule_destination
- [] delete_v2_logging_level
- [X] deprecate_thing_type
- [] describe_account_audit_configuration
- [] describe_audit_finding
- [] describe_audit_mitigation_actions_task
- [] describe_audit_suppression
- [] describe_audit_task
- [] describe_authorizer
- [] describe_billing_group
- [X] describe_ca_certificate
- [X] describe_certificate
- [] describe_custom_metric
- [] describe_default_authorizer
- [] describe_detect_mitigation_actions_task
- [] describe_dimension
- [X] describe_domain_configuration
- [X] describe_endpoint
- [] describe_event_configurations
- [] describe_fleet_metric
- [] describe_index
- [X] describe_job
- [X] describe_job_execution
- [] describe_job_template

- [] describe_managed_job_template
- [] describe_mitigation_action
- [] describe_provisioning_template
- [] describe_provisioning_template_version
- [] describe_role_alias
- [] describe_scheduled_audit
- [] describe_security_profile
- [] describe_stream
- [X] describe_thing
- [X] describe_thing_group
- [] describe_thing_registration_task
- [X] describe_thing_type
- [X] detach_policy
- [X] detach_principal_policy
- [] detach_security_profile
- [X] detach_thing_principal
- [X] disable_topic_rule
- [X] enable_topic_rule
- [] get_behavior_model_training_summaries
- [] get_buckets_aggregation
- [] get_cardinality
- [] get_effective_policies
- [] get_indexing_configuration
- [X] get_job_document
- [] get_logging_options
- [] get_ota_update
- [] get_percentiles
- [X] get_policy
- [X] get_policy_version
- [X] get_registration_code
- [] get_statistics
- [X] get_topic_rule
- [] get_topic_rule_destination
- [] get_v2_logging_options
- [] list_active_violations
- [X] list_attached_policies

- [] list_audit_findings
- [] list_audit_mitigation_actions_executions
- [] list_audit_mitigation_actions_tasks
- [] list_audit_suppressions
- [] list_audit_tasks
- [] list_authorizers
- [] list_billing_groups
- [] list_ca_certificates
- [X] list_certificates

Pagination is not yet implemented

- [X] list_certificates_by_ca
- [] list_custom_metrics
- [] list_detect_mitigation_actions_executions
- [] list_detect_mitigation_actions_tasks
- [] list_dimensions
- [X] list_domain_configurations
- [] list_fleet_metrics
- [] list_indices
- [X] list_job_executions_for_job
- [X] list_job_executions_for_thing
- [] list_job_templates
- [X] list_jobs

The following parameter are not yet implemented: Status, TargetSelection, ThingGroupName, Thing-
GroupId

- [] list_managed_job_templates
- [] list_metric_values
- [] list_mitigation_actions
- [] list_ota_updates
- [] list_outgoing_certificates
- [X] list_policies
- [X] list_policy_principals
- [X] list_policy_versions
- [X] list_principal_policies
- [X] list_principal_things
- [] list_provisioning_template_versions

- [] list_provisioning_templates
- [] list_related_resources_for_audit_finding
- [] list_role_aliases
- [] list_scheduled_audits
- [] list_security_profiles
- [] list_security_profiles_for_target
- [] list_streams
- [] list_tags_for_resource
- [X] list_targets_for_policy
- [] list_targets_for_security_profile
- [X] list_thing_groups
- [X] list_thing_groups_for_thing

 Pagination is not yet implemented

- [X] list_thing_principals
- [] list_thing_registration_task_reports
- [] list_thing_registration_tasks
- [X] list_thing_types
- [X] list_things
- [] list_things_in_billing_group
- [X] list_things_in_thing_group

 Pagination and the recursive-parameter is not yet implemented

- [] list_topic_rule_destinations
- [X] list_topic_rules
- [] list_v2_logging_levels
- [] list_violation_events
- [] put_verification_state_on_violation
- [X] register_ca_certificate

 The VerificationCertificate-parameter is not yet implemented

- [X] register_certificate
- [X] register_certificate_without_ca
- [] register_thing
- [] reject_certificate_transfer
- [] remove_thing_from_billing_group
- [X] remove_thing_from_thing_group
- [X] replace_topic_rule
- [X] search_index

Pagination is not yet implemented. Only basic search queries are supported for now.

- [] set_default_authorizer
- [X] set_default_policy_version
- [] set_logging_options
- [] set_v2_logging_level
- [] set_v2_logging_options
- [] start_audit_mitigation_actions_task
- [] start_detect_mitigation_actions_task
- [] start_on_demand_audit_task
- [] start_thing_registration_task
- [] stop_thing_registration_task
- [] tag_resource
- [] test_authorization
- [] test_invoke_authorizer
- [] transfer_certificate
- [] untag_resource
- [] update_account_audit_configuration
- [] update_audit_suppression
- [] update_authorizer
- [] update_billing_group
- [X] update_ca_certificate

The newAutoRegistrationStatus and removeAutoRegistration-parameters are not yet implemented

- [X] update_certificate
- [] update_custom_metric
- [] update_dimension
- [X] update_domain_configuration
- [] update_dynamic_thing_group
- [] update_event_configurations
- [] update_fleet_metric
- [] update_indexing_configuration
- [] update_job
- [] update_mitigation_action
- [] update_provisioning_template
- [] update_role_alias
- [] update_scheduled_audit
- [] update_security_profile

- [] update_stream
- [X] update_thing
 - The ExpectedVersion-parameter is not yet implemented
- [X] update_thing_group
- [X] update_thing_groups_for_thing
- [] update_topic_rule_destination
- [] validate_security_profile_behaviors

2.8.57 iot-data

Example usage

```
@mock_iotdata
def test_iotdata_behaviour:
    boto3.client("iot-data")
    ...
```

Implemented features for this service

- [X] delete_thing_shadow
- [] get_retained_message
- [X] get_thing_shadow
- [] list_named_shadows_for_thing
- [] list_retained_messages
- [X] publish
- [X] update_thing_shadow
 - spec of payload:**
 - need node *state*
 - state node must be an Object
 - State contains an invalid node: ‘foo’

2.8.58 kinesis

Example usage

```
@mock_kinesis
def test_kinesis_behaviour:
    boto3.client("kinesis")
    ...
```

Implemented features for this service

- [X] add_tags_to_stream
- [X] create_stream
- [X] decrease_stream_retention_period

- [X] delete_stream
- [X] deregister_stream_consumer
- [] describe_limits
- [X] describe_stream
- [X] describe_stream_consumer
- [X] describe_stream_summary
- [X] disable_enhanced_monitoring
- [X] enable_enhanced_monitoring
- [X] get_records
- [X] get_shard_iterator
- [X] increase_stream_retention_period
- [X] list_shards
- [X] list_stream_consumers

 Pagination is not yet implemented

- [X] list_streams
- [X] list_tags_for_stream
- [X] merge_shards
- [X] put_record
- [X] put_records
- [X] register_stream_consumer
- [X] remove_tags_from_stream
- [X] split_shard
- [X] start_stream_encryption
- [X] stop_stream_encryption
- [] subscribe_to_shard
- [X] update_shard_count
- [X] update_stream_mode

2.8.59 kinesis-video-archived-media

Example usage

```
@mock_kinesisvideoarchivedmedia
def test_kinesisvideoarchivedmedia_behaviour:
    boto3.client("kinesis-video-archived-media")
    ...
```

Implemented features for this service

- [X] get_clip

- [X] `get_dash_streaming_session_url`
- [X] `get_hls_streaming_session_url`
- [] `get_images`
- [] `get_media_for_fragment_list`
- [] `list_fragments`

2.8.60 kinesisvideo

Example usage

```
@mock_kinesisvideo
def test_kinesisvideo_behaviour:
    boto3.client("kinesisvideo")
    ...
```

Implemented features for this service

- [] `create_signaling_channel`
- [X] `create_stream`
- [] `delete_signaling_channel`
- [X] `delete_stream`

The `CurrentVersion`-parameter is not yet implemented

- [] `describe_edge_configuration`
- [] `describe_image_generation_configuration`
- [] `describe_mapped_resource_configuration`
- [] `describe_media_storage_configuration`
- [] `describe_notification_configuration`
- [] `describe_signaling_channel`
- [X] `describe_stream`
- [X] `get_data_endpoint`
- [] `get_signaling_channel_endpoint`
- [] `list_signaling_channels`
- [X] `list_streams`

Pagination and the `StreamNameCondition`-parameter are not yet implemented

- [] `list_tags_for_resource`
- [] `list_tags_for_stream`
- [] `start_edge_configuration_update`
- [] `tag_resource`
- [] `tag_stream`
- [] `untag_resource`

- [] untag_stream
- [] update_data_retention
- [] update_image_generation_configuration
- [] update_media_storage_configuration
- [] update_notification_configuration
- [] update_signaling_channel
- [] update_stream

2.8.61 kms

Example usage

```
@mock_kms
def test_kms_behaviour:
    boto3.client("kms")
    ...
```

Implemented features for this service

- [X] cancel_key_deletion
- [] connect_custom_key_store
- [] create_alias
- [] create_custom_key_store
- [X] create_grant
- [X] create_key

The provided Policy currently does not need to be valid. If it is valid, Moto will perform authorization checks on key-related operations, just like AWS does.

These authorization checks are quite basic for now. Moto will only throw an AccessDeniedException if the following

- The principal is set to “*”
 - The resource is set to “*”
 - The Action matches *describe_key*
- [X] decrypt
 - [X] delete_alias Delete the alias.
 - [] delete_custom_key_store
 - [] delete_imported_key_material
 - [] describe_custom_key_stores
 - [X] describe_key
 - [X] disable_key
 - [X] disable_key_rotation
 - [] disconnect_custom_key_store

- [X] enable_key
- [X] enable_key_rotation
- [X] encrypt
- [X] generate_data_key
- [] generate_data_key_pair
- [] generate_data_key_pair_without_plaintext
- [] generate_data_key_without_plaintext
- [] generate_mac
- [] generate_random
- [X] get_key_policy
- [X] get_key_rotation_status
- [] get_parameters_for_import
- [] get_public_key
- [] import_key_material
- [] list_aliases
- [X] list_grants
- [] list_key_policies
- [X] list_keys
- [X] list_resource_tags
- [X] list_retirable_grants
- [X] put_key_policy
- [X] re_encrypt
- [X] replicate_key
- [X] retire_grant
- [X] revoke_grant
- [X] schedule_key_deletion
- [X] sign Sign message using generated private key.
 - signing_algorithm is ignored and hardcoded to RSASSA_PSS_SHA_256
 - grant_tokens are not implemented
- [X] tag_resource
- [X] untag_resource
- [] update_alias
- [] update_custom_key_store
- [X] update_key_description
- [] update_primary_region
- [X] verify Verify message using public key from generated private key.

- signing_algorithm is ignored and hardcoded to RSASSA_PSS_SHA_256
- grant_tokens are not implemented
- [] verify_mac

2.8.62 lambda

class moto.awslambda.models.LambdaBackend(region_name: str, account_id: str)

Implementation of the AWS Lambda endpoint. Invoking functions is supported - they will run inside a Docker container, emulating the real AWS behaviour as closely as possible.

It is possible to connect from AWS Lambdas to other services, as long as you are running Moto in ServerMode. The Lambda has access to environment variables `MOTO_HOST` and `MOTO_PORT`, which can be used to build the url that MotoServer runs on:

```
def lambda_handler(event, context):
    host = os.environ.get("MOTO_HOST")
    port = os.environ.get("MOTO_PORT")
    url = host + ":" + port
    ec2 = boto3.client('ec2', region_name='us-west-2', endpoint_url=url)

    # Or even simpler:
    full_url = os.environ.get("MOTO_HTTP_ENDPOINT")
    ec2 = boto3.client("ec2", region_name="eu-west-1", endpoint_url=full_url)

    ec2.do_whatever_inside_the_existing_moto_server()
```

Moto will run on port 5000 by default. This can be overwritten by setting an environment variable when starting Moto:

```
# This env var will be propagated to the Docker container running the Lambda
↳ functions
MOTO_PORT=5000 moto_server
```

The Docker container uses the default network mode, *bridge*. The following environment variables are available for fine-grained control over the Docker connection options:

```
# Provide the name of a custom network to connect to
MOTO_DOCKER_NETWORK_NAME=mycustomnetwork moto_server

# Override the network mode
# For example, network_mode=host would use the network of the host machine
# Note that this option will be ignored if MOTO_DOCKER_NETWORK_NAME is also set
MOTO_DOCKER_NETWORK_MODE=host moto_server
```

The Docker images used by Moto are taken from the following repositories:

- `mlupin/docker-lambda` (for recent versions)
- `lambci/lambda` (for older/outdated versions)

Use the following environment variable to configure Moto to look for images in an additional repository:

```
MOTO_DOCKER_LAMBDA_IMAGE=mlupin/docker-lambda
```

Note: When using the decorators, a Docker container cannot reach Moto, as the Docker-container loses all mock-context. Any boto3-involutions used within your Lambda will try to connect to AWS.

Example usage

```
@mock_lambda
def test_lambda_behaviour:
    boto3.client("lambda")
    ...
```

Implemented features for this service

- [] add_layer_version_permission
- [X] add_permission
- [X] create_alias
- [] create_code_signing_config
- [X] create_event_source_mapping
- [X] create_function

The Code.ImageUri is not validated by default. Set environment variable MOTO_LAMBDA_STUB_ECR=false if you want to validate the image exists in our mocked ECR.

- [X] create_function_url_config
The Qualifier-parameter is not yet implemented. Function URLs are not yet mocked, so invoking them will fail
- [X] delete_alias
- [] delete_code_signing_config
- [X] delete_event_source_mapping
- [X] delete_function
- [] delete_function_code_signing_config
- [X] delete_function_concurrency
- [] delete_function_event_invoke_config
- [X] delete_function_url_config

The Qualifier-parameter is not yet implemented

- [X] delete_layer_version
- [] delete_provisioned_concurrency_config
- [] get_account_settings
- [X] get_alias
- [X] get_code_signing_config
- [X] get_event_source_mapping
- [X] get_function
- [] get_function_code_signing_config

- [X] `get_function_concurrency`
 - [] `get_function_configuration`
 - [] `get_function_event_invoke_config`
 - [X] `get_function_url_config`
- The Qualifier-parameter is not yet implemented

- [X] `get_layer_version`
- [] `get_layer_version_by_arn`
- [] `get_layer_version_policy`
- [X] `get_policy`
- [] `get_provisioned_concurrency_config`
- [] `get_runtime_management_config`
- [X] `invoke`

Invoking a Function with `PackageType=Image` is not yet supported.

- [] `invoke_async`
- [] `list_aliases`
- [] `list_code_signing_configs`
- [X] `list_event_source_mappings`
- [] `list_function_event_invoke_configs`
- [] `list_function_url_configs`
- [X] `list_functions`
- [] `list_functions_by_code_signing_config`
- [] `list_layer_versions`
- [X] `list_layers`
- [] `list_provisioned_concurrency_configs`
- [X] `list_tags`
- [X] `list_versions_by_function`
- [X] `publish_layer_version`
- [] `publish_version`
- [] `put_function_code_signing_config`
- [X] `put_function_concurrency`
- [] `put_function_event_invoke_config`
- [] `put_provisioned_concurrency_config`
- [] `put_runtime_management_config`
- [] `remove_layer_version_permission`
- [X] `remove_permission`
- [X] `tag_resource`

- [X] untag_resource
- [X] update_alias

The RevisionId parameter is not yet implemented

- [] update_code_signing_config
- [X] update_event_source_mapping
- [X] update_function_code
- [X] update_function_configuration
- [] update_function_event_invoke_config
- [X] update_function_url_config

The Qualifier-parameter is not yet implemented

2.8.63 logs

Example usage

```
@mock_logs
def test_logs_behaviour:
    boto3.client("logs")
    ...
```

Implemented features for this service

- [] associate_kms_key
- [] cancel_export_task
- [X] create_export_task
- [X] create_log_group
- [X] create_log_stream
- [] delete_data_protection_policy
- [] delete_destination
- [X] delete_log_group
- [X] delete_log_stream
- [X] delete_metric_filter
- [] delete_query_definition
- [X] delete_resource_policy Remove resource policy with a policy name matching given name.
- [X] delete_retention_policy
- [X] delete_subscription_filter
- [] describe_destinations
- [] describe_export_tasks
- [X] describe_log_groups
- [X] describe_log_streams

- [X] describe_metric_filters
- [] describe_queries
- [] describe_query_definitions
- [X] describe_resource_policies Return list of resource policies.

The next_token and limit arguments are ignored. The maximum number of resource policies per region is a small number (less than 50), so pagination isn't needed.
- [X] describe_subscription_filters
- [] disassociate_kms_key
- [X] filter_log_events

The following filter patterns are currently supported: Single Terms, Multiple Terms, Exact Phrases. If the pattern is not supported, all events are returned.
- [] get_data_protection_policy
- [X] get_log_events
- [] get_log_group_fields
- [] get_log_record
- [] get_query_results
- [] list_tags_for_resource
- [X] list_tags_log_group
- [] put_data_protection_policy
- [] put_destination
- [] put_destination_policy
- [X] put_log_events

The SequenceToken-parameter is not yet implemented
- [X] put_metric_filter
- [] put_query_definition
- [X] put_resource_policy Creates/updates resource policy and return policy object
- [X] put_retention_policy
- [X] put_subscription_filter
- [X] start_query
- [] stop_query
- [X] tag_log_group
- [] tag_resource
- [] test_metric_filter
- [X] untag_log_group
- [] untag_resource

2.8.64 managedblockchain

Example usage

```
@mock_managedblockchain
def test_managedblockchain_behaviour:
    boto3.client("managedblockchain")
    ...
```

Implemented features for this service

- [] create_accessor
- [X] create_member
- [X] create_network
- [X] create_node
- [X] create_proposal
- [] delete_accessor
- [X] delete_member
- [X] delete_node
- [] get_accessor
- [X] get_member
- [X] get_network
- [X] get_node
- [X] get_proposal
- [] list_accessors
- [X] list_invitations
- [X] list_members
- [X] list_networks
- [X] list_nodes
- [X] list_proposal_votes
- [X] list_proposals
- [] list_tags_for_resource
- [X] reject_invitation
- [] tag_resource
- [] untag_resource
- [X] update_member
- [X] update_node
- [X] vote_on_proposal

2.8.65 mediconnect

Example usage

```
@mock_mediconnect
def test_mediconnect_behaviour:
    boto3.client("mediconnect")
    ...
```

Implemented features for this service

- [] add_flow_media_streams
- [X] add_flow_outputs
- [X] add_flow_sources
- [X] add_flow_vpc_interfaces
- [X] create_flow
- [X] delete_flow
- [X] describe_flow
- [] describe_offering
- [] describe_reservation
- [X] grant_flow_entitlements
- [] list_entitlements
- [X] list_flows
- [] list_offerings
- [] list_reservations
- [X] list_tags_for_resource
- [] purchase_offering
- [] remove_flow_media_stream
- [X] remove_flow_output
- [] remove_flow_source
- [X] remove_flow_vpc_interface
- [X] revoke_flow_entitlement
- [X] start_flow
- [X] stop_flow
- [X] tag_resource
- [] untag_resource
- [] update_flow
- [X] update_flow_entitlement
- [] update_flow_media_stream
- [X] update_flow_output

- [X] update_flow_source

2.8.66 medialive

Example usage

```
@mock_medialive
def test_medialive_behaviour:
    boto3.client("medialive")
    ...
```

Implemented features for this service

- [] accept_input_device_transfer
- [] batch_delete
- [] batch_start
- [] batch_stop
- [] batch_update_schedule
- [] cancel_input_device_transfer
- [] claim_device
- [X] create_channel

The RequestID and Reserved parameters are not yet implemented

- [X] create_input

The VPC and RequestId parameters are not yet implemented

- [] create_input_security_group
- [] create_multiplex
- [] create_multiplex_program
- [] create_partner_input
- [] create_tags
- [X] delete_channel
- [X] delete_input
- [] delete_input_security_group
- [] delete_multiplex
- [] delete_multiplex_program
- [] delete_reservation
- [] delete_schedule
- [] delete_tags
- [X] describe_channel
- [X] describe_input
- [] describe_input_device

- [] describe_input_device_thumbnail
- [] describe_input_security_group
- [] describe_multiplex
- [] describe_multiplex_program
- [] describe_offering
- [] describe_reservation
- [] describe_schedule
- [X] list_channels
- [] list_input_device_transfers
- [] list_input_devices
- [] list_input_security_groups
- [X] list_inputs
- [] list_multiplex_programs
- [] list_multiplexes
- [] list_offerings
- [] list_reservations
- [] list_tags_for_resource
- [] purchase_offering
- [] reboot_input_device
- [] reject_input_device_transfer
- [X] start_channel
- [] start_input_device_maintenance_window
- [] start_multiplex
- [X] stop_channel
- [] stop_multiplex
- [] transfer_input_device
- [X] update_channel
- [] update_channel_class
- [X] update_input
- [] update_input_device
- [] update_input_security_group
- [] update_multiplex
- [] update_multiplex_program
- [] update_reservation

2.8.67 mediapackage

Example usage

```
@mock_mediapackage
def test_mediapackage_behaviour:
    boto3.client("mediapackage")
    ...
```

Implemented features for this service

- [] configure_logs
- [X] create_channel
- [] create_harvest_job
- [X] create_origin_endpoint
- [X] delete_channel
- [X] delete_origin_endpoint
- [X] describe_channel
- [] describe_harvest_job
- [X] describe_origin_endpoint
- [X] list_channels
- [] list_harvest_jobs
- [X] list_origin_endpoints
- [] list_tags_for_resource
- [] rotate_channel_credentials
- [] rotate_ingest_endpoint_credentials
- [] tag_resource
- [] untag_resource
- [] update_channel
- [X] update_origin_endpoint

2.8.68 mediastore

Example usage

```
@mock_mediastore
def test_mediastore_behaviour:
    boto3.client("mediastore")
    ...
```

Implemented features for this service

- [X] create_container
- [X] delete_container

- [] delete_container_policy
- [] delete_cors_policy
- [] delete_lifecycle_policy
- [] delete_metric_policy
- [X] describe_container
- [X] get_container_policy
- [] get_cors_policy
- [X] get_lifecycle_policy
- [X] get_metric_policy
- [X] list_containers

 Pagination is not yet implemented

- [X] list_tags_for_resource
- [X] put_container_policy
- [] put_cors_policy
- [X] put_lifecycle_policy
- [X] put_metric_policy
- [] start_access_logging
- [] stop_access_logging
- [] tag_resource
- [] untag_resource

2.8.69 mediastore-data

Example usage

```
@mock_mediastoredata
def test_mediastoredata_behaviour:
    boto3.client("mediastore-data")
    ...
```

Implemented features for this service

- [X] delete_object
- [] describe_object
- [X] get_object

 The Range-parameter is not yet supported.

- [X] list_items

 The Path- and MaxResults-parameters are not yet supported.

- [X] put_object

 The following parameters are not yet implemented: ContentType, CacheControl, UploadAvailability

2.8.70 meteringmarketplace

Example usage

```
@mock_meteringmarketplace
def test_meteringmarketplace_behaviour:
    boto3.client("meteringmarketplace")
    ...
```

Implemented features for this service

- [X] batch_meter_usage
- [] meter_usage
- [] register_usage
- [] resolve_customer

2.8.71 mq

class `moto.mq.models.MQBackend`(*region_name*, *account_id*)

No EC2 integration exists yet - subnet ID's and security group values are not validated. Default values may not exist.

Example usage

```
@mock_mq
def test_mq_behaviour:
    boto3.client("mq")
    ...
```

Implemented features for this service

- [X] create_broker
- [X] create_configuration
- [X] create_tags
- [X] create_user
- [X] delete_broker
- [X] delete_tags
- [X] delete_user
- [X] describe_broker
- [] describe_broker_engine_types
- [] describe_broker_instance_options
- [X] describe_configuration
- [X] describe_configuration_revision
- [X] describe_user
- [X] list_brokers

Pagination is not yet implemented

- [] list_configuration_revisions

- [X] list_configurations

 Pagination has not yet been implemented.

- [X] list_tags

- [X] list_users

- [X] reboot_broker

- [X] update_broker

- [X] update_configuration

 No validation occurs on the provided XML. The authenticationStrategy may be changed depending on the provided configuration.

- [X] update_user

2.8.72 opsworks

Example usage

```
@mock_opsworks
def test_opsworks_behaviour:
    boto3.client("opsworks")
    ...
```

Implemented features for this service

- [] assign_instance

- [] assign_volume

- [] associate_elastic_ip

- [] attach_elastic_load_balancer

- [] clone_stack

- [X] create_app

- [] create_deployment

- [X] create_instance

- [X] create_layer

- [X] create_stack

- [] create_user_profile

- [] delete_app

- [] delete_instance

- [] delete_layer

- [] delete_stack

- [] delete_user_profile

- [] deregister_ecs_cluster

- [] deregister_elastic_ip

- deregister_instance
- deregister_rds_db_instance
- deregister_volume
- describe_agent_versions
- describe_apps
- describe_commands
- describe_deployments
- describe_ecs_clusters
- describe_elastic_ips
- describe_elastic_load_balancers
- describe_instances
- describe_layers
- describe_load_based_auto_scaling
- describe_my_user_profile
- describe_operating_systems
- describe_permissions
- describe_raid_arrays
- describe_rds_db_instances
- describe_service_errors
- describe_stack_provisioning_parameters
- describe_stack_summary
- describe_stacks
- describe_time_based_auto_scaling
- describe_user_profiles
- describe_volumes
- detach_elastic_load_balancer
- disassociate_elastic_ip
- get_hostname_suggestion
- grant_access
- list_tags
- reboot_instance
- register_ecs_cluster
- register_elastic_ip
- register_instance
- register_rds_db_instance
- register_volume

- [] set_load_based_auto_scaling
- [] set_permission
- [] set_time_based_auto_scaling
- [X] start_instance
- [] start_stack
- [] stop_instance
- [] stop_stack
- [] tag_resource
- [] unassign_instance
- [] unassign_volume
- [] untag_resource
- [] update_app
- [] update_elastic_ip
- [] update_instance
- [] update_layer
- [] update_my_user_profile
- [] update_rds_db_instance
- [] update_stack
- [] update_user_profile
- [] update_volume

2.8.73 organizations

Example usage

```
@mock_organizations
def test_organizations_behaviour:
    boto3.client("organizations")
    ...
```

Implemented features for this service

- [] accept_handshake
- [X] attach_policy
- [] cancel_handshake
- [X] close_account
- [X] create_account
- [] create_gov_cloud_account
- [X] create_organization
- [X] create_organizational_unit

- create_policy
- decline_handshake
- delete_organization
- delete_organizational_unit
- delete_policy
- delete_resource_policy
- deregister_delegated_administrator
- describe_account
- describe_create_account_status
- describe_effective_policy
- describe_handshake
- describe_organization
- describe_organizational_unit
- describe_policy
- describe_resource_policy
- detach_policy
- disable_aws_service_access
- disable_policy_type
- enable_all_features
- enable_aws_service_access
- enable_policy_type
- invite_account_to_organization
- leave_organization
- list_accounts
- list_accounts_for_parent
- list_aws_service_access_for_organization
- list_children
- list_create_account_status
- list_delegated_administrators
- list_delegated_services_for_account
- list_handshakes_for_account
- list_handshakes_for_organization
- list_organizational_units_for_parent
- list_parents
- list_policies
- list_policies_for_target

- [X] list_roots
- [X] list_tags_for_resource
- [X] list_targets_for_policy
- [X] move_account
- [] put_resource_policy
- [X] register_delegated_administrator
- [X] remove_account_from_organization
- [X] tag_resource
- [X] untag_resource
- [X] update_organizational_unit
- [X] update_policy

2.8.74 Patching other Services

Since moto does not support every AWS service available there is a way to patch boto3 calls until they are supported.

To do so, you need to mock the `botocore.client.BaseClient._make_api_call` function using `mock.patch`:

```
import boto3
import botocore
from unittest.mock import patch

# Original botocore _make_api_call function
orig = botocore.client.BaseClient._make_api_call

# Mocked botocore _make_api_call function
def mock_make_api_call(self, operation_name, kwarg):
    # For example for the Access Analyzer service
    # As you can see the operation_name has the list_analyzers snake_case form but
    # we are using the ListAnalyzers form.
    # Rationale -> https://github.com/boto/botocore/blob/develop/botocore/client.py
    ↪ #L810:L816
    if operation_name == 'ListAnalyzers':
        return { "analyzers":
            [
                {
                    "arn": "ARN",
                    "name": "Test Analyzer" ,
                    "status": "Enabled",
                    "findings": 0,
                    "tags": "",
                    "type": "ACCOUNT",
                    "region": "eu-west-1"
                }
            ]
        }
    # If we don't want to patch the API call
    return orig(self, operation_name, kwarg)
```

(continues on next page)

(continued from previous page)

```
def test_list_findings():
    client = boto3.client("accessanalyzer")

    with patch('botocore.client.BaseClient._make_api_call', new=mock_make_api_call):
        analyzers_list = client.list_analyzers()
        assert len(analyzers_list["analyzers"]) == 1
        # include your assertions here
```

Note that this does not use Moto, to keep it simple, but if you use any moto-decorators in addition to the patch, the call to `orig(self, operation_name, kwarg)` will be intercepted by Moto.

2.8.75 personalize

class `moto.personalize.models.PersonalizeBackend`(*region_name, account_id*)
Implementation of Personalize APIs.

Example usage

```
@mock_personalize
def test_personalize_behaviour:
    boto3.client("personalize")
    ...
```

Implemented features for this service

- [] `create_batch_inference_job`
- [] `create_batch_segment_job`
- [] `create_campaign`
- [] `create_dataset`
- [] `create_dataset_export_job`
- [] `create_dataset_group`
- [] `create_dataset_import_job`
- [] `create_event_tracker`
- [] `create_filter`
- [] `create_metric_attribution`
- [] `create_recommender`
- [X] `create_schema`
- [] `create_solution`
- [] `create_solution_version`
- [] `delete_campaign`
- [] `delete_dataset`
- [] `delete_dataset_group`
- [] `delete_event_tracker`
- [] `delete_filter`

- delete_metric_attribution
- delete_recommender
- delete_schema
- delete_solution
- describe_algorithm
- describe_batch_inference_job
- describe_batch_segment_job
- describe_campaign
- describe_dataset
- describe_dataset_export_job
- describe_dataset_group
- describe_dataset_import_job
- describe_event_tracker
- describe_feature_transformation
- describe_filter
- describe_metric_attribution
- describe_recipe
- describe_recommender
- describe_schema
- describe_solution
- describe_solution_version
- get_solution_metrics
- list_batch_inference_jobs
- list_batch_segment_jobs
- list_campaigns
- list_dataset_export_jobs
- list_dataset_groups
- list_dataset_import_jobs
- list_datasets
- list_event_trackers
- list_filters
- list_metric_attribution_metrics
- list_metric_attributions
- list_recipes
- list_recommenders
- list_schemas

Pagination is not yet implemented

- [] list_solution_versions
- [] list_solutions
- [] list_tags_for_resource
- [] start_recommender
- [] stop_recommender
- [] stop_solution_version_creation
- [] tag_resource
- [] untag_resource
- [] update_campaign
- [] update_metric_attribution
- [] update_recommender

2.8.76 pinpoint

class moto.pinpoint.models.**PinpointBackend**(*region_name*, *account_id*)
Implementation of Pinpoint APIs.

Example usage

```
@mock_pinpoint
def test_pinpoint_behaviour:
    boto3.client("pinpoint")
    ...
```

Implemented features for this service

- [X] create_app
- [] create_campaign
- [] create_email_template
- [] create_export_job
- [] create_import_job
- [] create_in_app_template
- [] create_journey
- [] create_push_template
- [] create_recommender_configuration
- [] create_segment
- [] create_sms_template
- [] create_voice_template
- [] delete_adm_channel
- [] delete_apns_channel
- [] delete_apns_sandbox_channel

- [] delete_apns_voip_channel
- [] delete_apns_voip_sandbox_channel
- [X] delete_app
- [] delete_baidu_channel
- [] delete_campaign
- [] delete_email_channel
- [] delete_email_template
- [] delete_endpoint
- [X] delete_event_stream
- [] delete_gcm_channel
- [] delete_in_app_template
- [] delete_journey
- [] delete_push_template
- [] delete_recommender_configuration
- [] delete_segment
- [] delete_sms_channel
- [] delete_sms_template
- [] delete_user_endpoints
- [] delete_voice_channel
- [] delete_voice_template
- [] get_adm_channel
- [] get_apns_channel
- [] get_apns_sandbox_channel
- [] get_apns_voip_channel
- [] get_apns_voip_sandbox_channel
- [X] get_app
- [] get_application_date_range_kpi
- [X] get_application_settings
- [X] get_apps

 Pagination is not yet implemented

- [] get_baidu_channel
- [] get_campaign
- [] get_campaign_activities
- [] get_campaign_date_range_kpi
- [] get_campaign_version
- [] get_campaign_versions

- [] `get_campaigns`
- [] `get_channels`
- [] `get_email_channel`
- [] `get_email_template`
- [] `get_endpoint`
- [X] `get_event_stream`
- [] `get_export_job`
- [] `get_export_jobs`
- [] `get_gcm_channel`
- [] `get_import_job`
- [] `get_import_jobs`
- [] `get_in_app_messages`
- [] `get_in_app_template`
- [] `get_journey`
- [] `get_journey_date_range_kpi`
- [] `get_journey_execution_activity_metrics`
- [] `get_journey_execution_metrics`
- [] `get_push_template`
- [] `get_recommender_configuration`
- [] `get_recommender_configurations`
- [] `get_segment`
- [] `get_segment_export_jobs`
- [] `get_segment_import_jobs`
- [] `get_segment_version`
- [] `get_segment_versions`
- [] `get_segments`
- [] `get_sms_channel`
- [] `get_sms_template`
- [] `get_user_endpoints`
- [] `get_voice_channel`
- [] `get_voice_template`
- [] `list_journeys`
- [X] `list_tags_for_resource`
- [] `list_template_versions`
- [] `list_templates`
- [] `phone_number_validate`

- put_event_stream
- put_events
- remove_attributes
- send_messages
- send_otp_message
- send_users_messages
- tag_resource
- untag_resource
- update_adm_channel
- update_apns_channel
- update_apns_sandbox_channel
- update_apns_voip_channel
- update_apns_voip_sandbox_channel
- update_application_settings
- update_baidu_channel
- update_campaign
- update_email_channel
- update_email_template
- update_endpoint
- update_endpoints_batch
- update_gcm_channel
- update_in_app_template
- update_journey
- update_journey_state
- update_push_template
- update_recommender_configuration
- update_segment
- update_sms_channel
- update_sms_template
- update_template_active_version
- update_voice_channel
- update_voice_template
- verify_otp_message

2.8.77 polly

Example usage

```
@mock_polly
def test_polly_behaviour:
    boto3.client("polly")
    ...
```

Implemented features for this service

- [X] delete_lexicon
- [X] describe_voices
 - Pagination is not yet implemented
- [X] get_lexicon
- [] get_speech_synthesis_task
- [X] list_lexicons
 - Pagination is not yet implemented
- [] list_speech_synthesis_tasks
- [X] put_lexicon
- [] start_speech_synthesis_task
- [] synthesize_speech

2.8.78 quicksight

class moto.quicksight.models.**QuickSightBackend**(*region_name, account_id*)
Implementation of QuickSight APIs.

Example usage

```
@mock_quicksight
def test_quicksight_behaviour:
    boto3.client("quicksight")
    ...
```

Implemented features for this service

- [] cancel_ingestion
- [] create_account_customization
- [] create_account_subscription
- [] create_analysis
- [] create_dashboard
- [X] create_data_set
- [] create_data_source
- [] create_folder
- [] create_folder_membership

- create_group
- create_group_membership
- create_iam_policy_assignment
- create_ingestion
- create_namespace
- create_template
- create_template_alias
- create_theme
- create_theme_alias
- delete_account_customization
- delete_account_subscription
- delete_analysis
- delete_dashboard
- delete_data_set
- delete_data_source
- delete_folder
- delete_folder_membership
- delete_group
- delete_group_membership
- delete_iam_policy_assignment
- delete_namespace
- delete_template
- delete_template_alias
- delete_theme
- delete_theme_alias
- delete_user
- delete_user_by_principal_id
- describe_account_customization
- describe_account_settings
- describe_account_subscription
- describe_analysis
- describe_analysis_definition
- describe_analysis_permissions
- describe_dashboard
- describe_dashboard_definition
- describe_dashboard_permissions

- [] describe_data_set
- [] describe_data_set_permissions
- [] describe_data_source
- [] describe_data_source_permissions
- [] describe_folder
- [] describe_folder_permissions
- [] describe_folder_resolved_permissions
- [X] describe_group
- [X] describe_group_membership
- [] describe_iam_policy_assignment
- [] describe_ingestion
- [] describe_ip_restriction
- [] describe_namespace
- [] describe_template
- [] describe_template_alias
- [] describe_template_definition
- [] describe_template_permissions
- [] describe_theme
- [] describe_theme_alias
- [] describe_theme_permissions
- [X] describe_user
- [] generate_embed_url_for_anonymous_user
- [] generate_embed_url_for_registered_user
- [] get_dashboard_embed_url
- [] get_session_embed_url
- [] list_analyses
- [] list_dashboard_versions
- [] list_dashboards
- [] list_data_sets
- [] list_data_sources
- [] list_folder_members
- [] list_folders
- [X] list_group_memberships
 - The NextToken and MaxResults parameters are not yet implemented
- [X] list_groups
 - The NextToken and MaxResults parameters are not yet implemented

- [] list_iam_policy_assignments
- [] list_iam_policy_assignments_for_user
- [] list_ingestions
- [] list_namespaces
- [] list_tags_for_resource
- [] list_template_aliases
- [] list_template_versions
- [] list_templates
- [] list_theme_aliases
- [] list_theme_versions
- [] list_themes
- [] list_user_groups
- [X] list_users

The NextToken and MaxResults parameters are not yet implemented

- [X] register_user

The following parameters are not yet implemented: IamArn, SessionName, CustomsPermissionsName, ExternalLoginFederationProviderType, CustomFederationProviderUrl, ExternalLoginId

- [] restore_analysis
- [] search_analyses
- [] search_dashboards
- [] search_data_sets
- [] search_data_sources
- [] search_folders
- [] search_groups
- [] tag_resource
- [] untag_resource
- [] update_account_customization
- [] update_account_settings
- [] update_analysis
- [] update_analysis_permissions
- [] update_dashboard
- [] update_dashboard_permissions
- [] update_dashboard_published_version
- [] update_data_set
- [] update_data_set_permissions
- [] update_data_source

- update_data_source_permissions
- update_folder
- update_folder_permissions
- update_group
- update_iam_policy_assignment
- update_ip_restriction
- update_public_sharing_settings
- update_template
- update_template_alias
- update_template_permissions
- update_theme
- update_theme_alias
- update_theme_permissions
- update_user

2.8.79 ram

Example usage

```
@mock_ram
def test_ram_behaviour:
    boto3.client("ram")
    ...
```

Implemented features for this service

- accept_resource_share_invitation
- associate_resource_share
- associate_resource_share_permission
- create_resource_share
- delete_resource_share
- disassociate_resource_share
- disassociate_resource_share_permission
- enable_sharing_with_aws_organization
- get_permission
- get_resource_policies
- get_resource_share_associations
- get_resource_share_invitations
- get_resource_shares
- list_pending_invitation_resources

- [] list_permission_versions
- [] list_permissions
- [] list_principals
- [] list_resource_share_permissions
- [] list_resource_types
- [] list_resources
- [] promote_resource_share_created_from_policy
- [] reject_resource_share_invitation
- [] tag_resource
- [] untag_resource
- [X] update_resource_share

2.8.80 rds

Example usage

```
@mock_rds
def test_rds_behaviour:
    boto3.client("rds")
    ...
```

Implemented features for this service

- [] add_role_to_db_cluster
- [] add_role_to_db_instance
- [] add_source_identifier_to_subscription
- [X] add_tags_to_resource
- [] apply_pending_maintenance_action
- [] authorize_db_security_group_ingress
- [] backtrack_db_cluster
- [X] cancel_export_task
- [] copy_db_cluster_parameter_group
- [] copy_db_cluster_snapshot
- [] copy_db_parameter_group
- [] copy_db_snapshot
- [] copy_option_group
- [] create_blue_green_deployment
- [] create_custom_db_engine_version
- [X] create_db_cluster
- [] create_db_cluster_endpoint

- create_db_cluster_parameter_group
- create_db_cluster_snapshot
- create_db_instance
- create_db_instance_read_replica
- create_db_parameter_group
- create_db_proxy
- create_db_proxy_endpoint
- create_db_security_group
- create_db_snapshot
- create_db_subnet_group
- create_event_subscription
- create_global_cluster
- create_option_group
- delete_blue_green_deployment
- delete_custom_db_engine_version
- delete_db_cluster
- delete_db_cluster_endpoint
- delete_db_cluster_parameter_group
- delete_db_cluster_snapshot
- delete_db_instance
- delete_db_instance_automated_backup
- delete_db_parameter_group
- delete_db_proxy
- delete_db_proxy_endpoint
- delete_db_security_group
- delete_db_snapshot
- delete_db_subnet_group
- delete_event_subscription
- delete_global_cluster
- delete_option_group
- deregister_db_proxy_targets
- describe_account_attributes
- describe_blue_green_deployments
- describe_certificates
- describe_db_cluster_backtracks
- describe_db_cluster_endpoints

- describe_db_cluster_parameter_groups
- describe_db_cluster_parameters
- describe_db_cluster_snapshot_attributes
- describe_db_cluster_snapshots
- describe_db_clusters
- describe_db_engine_versions
- describe_db_instance_automated_backups
- describe_db_instances
- describe_db_log_files
- describe_db_parameter_groups
- describe_db_parameters
- describe_db_proxies
- describe_db_proxy_endpoints
- describe_db_proxy_target_groups
- describe_db_proxy_targets
- describe_db_security_groups
- describe_db_snapshot_attributes
- describe_db_snapshots
- describe_db_subnet_groups
- describe_engine_default_cluster_parameters
- describe_engine_default_parameters
- describe_event_categories
- describe_event_subscriptions
- describe_events
- describe_export_tasks
- describe_global_clusters
- describe_option_group_options
- describe_option_groups
- describe_orderable_db_instance_options
- describe_pending_maintenance_actions
- describe_reserved_db_instances
- describe_reserved_db_instances_offerings
- describe_source_regions
- describe_valid_db_instance_modifications
- download_db_log_file_portion
- failover_db_cluster

- failover_global_cluster
- list_tags_for_resource
- modify_activity_stream
- modify_certificates
- modify_current_db_cluster_capacity
- modify_custom_db_engine_version
- modify_db_cluster
- modify_db_cluster_endpoint
- modify_db_cluster_parameter_group
- modify_db_cluster_snapshot_attribute
- modify_db_instance
- modify_db_parameter_group
- modify_db_proxy
- modify_db_proxy_endpoint
- modify_db_proxy_target_group
- modify_db_snapshot
- modify_db_snapshot_attribute
- modify_db_subnet_group
- modify_event_subscription
- modify_global_cluster
- modify_option_group
- promote_read_replica
- promote_read_replica_db_cluster
- purchase_reserved_db_instances_offering
- reboot_db_cluster
- reboot_db_instance
- register_db_proxy_targets
- remove_from_global_cluster
- remove_role_from_db_cluster
- remove_role_from_db_instance
- remove_source_identifier_from_subscription
- remove_tags_from_resource
- reset_db_cluster_parameter_group
- reset_db_parameter_group
- restore_db_cluster_from_s3
- restore_db_cluster_from_snapshot

- [] restore_db_cluster_to_point_in_time
- [X] restore_db_instance_from_db_snapshot
- [] restore_db_instance_from_s3
- [] restore_db_instance_to_point_in_time
- [] revoke_db_security_group_ingress
- [] start_activity_stream
- [X] start_db_cluster
- [X] start_db_instance
- [] start_db_instance_automated_backups_replication
- [X] start_export_task
- [] stop_activity_stream
- [X] stop_db_cluster
- [X] stop_db_instance
- [] stop_db_instance_automated_backups_replication
- [] switchover_blue_green_deployment
- [] switchover_read_replica

2.8.81 redshift

Example usage

```
@mock_redshift
def test_redshift_behaviour:
    boto3.client("redshift")
    ...
```

Implemented features for this service

- [] accept_reserved_node_exchange
- [] add_partner
- [] associate_data_share_consumer
- [X] authorize_cluster_security_group_ingress
- [] authorize_data_share
- [] authorize_endpoint_access
- [] authorize_snapshot_access
- [] batch_delete_cluster_snapshots
- [] batch_modify_cluster_snapshots
- [] cancel_resize
- [] copy_cluster_snapshot
- [] create_authentication_profile

- create_cluster
- create_cluster_parameter_group
- create_cluster_security_group
- create_cluster_snapshot
- create_cluster_subnet_group
- create_endpoint_access
- create_event_subscription
- create_hsm_client_certificate
- create_hsm_configuration
- create_scheduled_action
- create_snapshot_copy_grant
- create_snapshot_schedule
- create_tags
- create_usage_limit
- deauthorize_data_share
- delete_authentication_profile
- delete_cluster
- delete_cluster_parameter_group
- delete_cluster_security_group
- delete_cluster_snapshot
- delete_cluster_subnet_group
- delete_endpoint_access
- delete_event_subscription
- delete_hsm_client_certificate
- delete_hsm_configuration
- delete_partner
- delete_scheduled_action
- delete_snapshot_copy_grant
- delete_snapshot_schedule
- delete_tags
- delete_usage_limit
- describe_account_attributes
- describe_authentication_profiles
- describe_cluster_db_revisions
- describe_cluster_parameter_groups
- describe_cluster_parameters

- describe_cluster_security_groups
- describe_cluster_snapshots
- describe_cluster_subnet_groups
- describe_cluster_tracks
- describe_cluster_versions
- describe_clusters
- describe_data_shares
- describe_data_shares_for_consumer
- describe_data_shares_for_producer
- describe_default_cluster_parameters
- describe_endpoint_access
- describe_endpoint_authorization
- describe_event_categories
- describe_event_subscriptions
- describe_events
- describe_hsm_client_certificates
- describe_hsm_configurations
- describe_logging_status
- describe_node_configuration_options
- describe_orderable_cluster_options
- describe_partners
- describe_reserved_node_exchange_status
- describe_reserved_node_offerings
- describe_reserved_nodes
- describe_resize
- describe_scheduled_actions
- describe_snapshot_copy_grants
- describe_snapshot_schedules
- describe_storage
- describe_table_restore_status
- describe_tags
- describe_usage_limits
- disable_logging
- disable_snapshot_copy
- disassociate_data_share_consumer
- enable_logging

- enable_snapshot_copy
- get_cluster_credentials
- get_cluster_credentials_with_iam
- get_reserved_node_exchange_configuration_options
- get_reserved_node_exchange_offerings
- modify_aqua_configuration
- modify_authentication_profile
- modify_cluster
- modify_cluster_db_revision
- modify_cluster_iam_roles
- modify_cluster_maintenance
- modify_cluster_parameter_group
- modify_cluster_snapshot
- modify_cluster_snapshot_schedule
- modify_cluster_subnet_group
- modify_endpoint_access
- modify_event_subscription
- modify_scheduled_action
- modify_snapshot_copy_retention_period
- modify_snapshot_schedule
- modify_usage_limit
- pause_cluster
- purchase_reserved_node_offering
- reboot_cluster
- reject_data_share
- reset_cluster_parameter_group
- resize_cluster
- restore_from_cluster_snapshot
- restore_table_from_cluster_snapshot
- resume_cluster
- revoke_cluster_security_group_ingress
- revoke_endpoint_access
- revoke_snapshot_access
- rotate_encryption_key
- update_partner_status

2.8.82 redshift-data

Example usage

```
@mock_redshiftdata
def test_redshiftdata_behaviour:
    boto3.client("redshift-data")
    ...
```

Implemented features for this service

- [] batch_execute_statement
- [X] cancel_statement
- [X] describe_statement
- [] describe_table
- [X] execute_statement
 - Runs an SQL statement Validation of parameters is very limited because there is no redshift integration
- [X] get_statement_result
 - Return static statement result StatementResult is the result of the SQL query “sql” passed as parameter when calling “execute_statement” As such, it cannot be mocked
- [] list_databases
- [] list_schemas
- [] list_statements
- [] list_tables

2.8.83 rekognition

class moto.rekognition.models.**RekognitionBackend**(*region_name: str, account_id: str*)
Implementation of Rekognition APIs.

Example usage

```
@mock_rekognition
def test_rekognition_behaviour:
    boto3.client("rekognition")
    ...
```

Implemented features for this service

- [] compare_faces
- [] copy_project_version
- [] create_collection
- [] create_dataset
- [] create_project
- [] create_project_version
- [] create_stream_processor

- [] delete_collection
- [] delete_dataset
- [] delete_faces
- [] delete_project
- [] delete_project_policy
- [] delete_project_version
- [] delete_stream_processor
- [] describe_collection
- [] describe_dataset
- [] describe_project_versions
- [] describe_projects
- [] describe_stream_processor
- [] detect_custom_labels
- [] detect_faces
- [] detect_labels
- [] detect_moderation_labels
- [] detect_protective_equipment
- [] detect_text
- [] distribute_dataset_entries
- [] get_celebrity_info
- [] get_celebrity_recognition
- [] get_content_moderation
- [] get_face_detection
- [X] get_face_search

This returns hardcoded values and none of the parameters are taken into account.

- [] get_label_detection
- [] get_person_tracking
- [] get_segment_detection
- [X] get_text_detection

This returns hardcoded values and none of the parameters are taken into account.

- [] index_faces
- [] list_collections
- [] list_dataset_entries
- [] list_dataset_labels
- [] list_faces
- [] list_project_policies

- [] list_stream_processors
- [] list_tags_for_resource
- [] put_project_policy
- [] recognize_celebrities
- [] search_faces
- [] search_faces_by_image
- [] start_celebrity_recognition
- [] start_content_moderation
- [] start_face_detection
- [X] start_face_search
- [] start_label_detection
- [] start_person_tracking
- [] start_project_version
- [] start_segment_detection
- [] start_stream_processor
- [X] start_text_detection
- [] stop_project_version
- [] stop_stream_processor
- [] tag_resource
- [] untag_resource
- [] update_dataset_entries
- [] update_stream_processor

2.8.84 resource-groups

Example usage

```
@mock_resourcegroups
def test_resourcegroups_behaviour:
    boto3.client("resource-groups")
    ...
```

Implemented features for this service

- [X] create_group
- [X] delete_group
- [] get_account_settings
- [X] get_group
- [X] get_group_configuration
- [] get_group_query

- [X] get_tags
- [] group_resources
- [] list_group_resources
- [X] list_groups
 - Pagination or the Filters-parameter is not yet implemented
- [X] put_group_configuration
- [] search_resources
- [X] tag
- [] ungroup_resources
- [X] untag
- [] update_account_settings
- [X] update_group
- [X] update_group_query

2.8.85 resourcegroupstaggingapi

Example usage

```
@mock_resourcegroupstaggingapi
def test_resourcegroupstaggingapi_behaviour:
    boto3.client("resourcegroupstaggingapi")
    ...
```

Implemented features for this service

- [] describe_report_creation
- [] get_compliance_summary
- [X] get_resources
- [X] get_tag_keys
- [X] get_tag_values
- [] start_report_creation
- [] tag_resources
- [] untag_resources

2.8.86 route53

Example usage

```
@mock_route53
def test_route53_behaviour:
    boto3.client("route53")
    ...
```

Implemented features for this service

- [] activate_key_signing_key
- [X] associate_vpc_with_hosted_zone
- [] change_cidr_collection
- [X] change_resource_record_sets
- [X] change_tags_for_resource
- [] create_cidr_collection
- [X] create_health_check
- [X] create_hosted_zone
- [] create_key_signing_key
- [X] create_query_logging_config Process the create_query_logging_config request.
- [X] create_reusable_delegation_set
- [] create_traffic_policy
- [] create_traffic_policy_instance
- [] create_traffic_policy_version
- [] create_vpc_association_authorization
- [] deactivate_key_signing_key
- [] delete_cidr_collection
- [X] delete_health_check
- [X] delete_hosted_zone
- [] delete_key_signing_key
- [X] delete_query_logging_config Delete query logging config, if it exists.
- [X] delete_reusable_delegation_set
- [] delete_traffic_policy
- [] delete_traffic_policy_instance
- [] delete_vpc_association_authorization
- [] disable_hosted_zone_dnssec
- [X] disassociate_vpc_from_hosted_zone
- [] enable_hosted_zone_dnssec
- [] get_account_limit

- [] get_change
- [] get_checker_ip_ranges
- [X] get_dnssec
- [] get_geo_location
- [X] get_health_check
- [] get_health_check_count
- [] get_health_check_last_failure_reason
- [] get_health_check_status
- [X] get_hosted_zone
- [X] get_hosted_zone_count
- [] get_hosted_zone_limit
- [X] get_query_logging_config Return query logging config, if it exists.
- [X] get_reusable_delegation_set
- [] get_reusable_delegation_set_limit
- [] get_traffic_policy
- [] get_traffic_policy_instance
- [] get_traffic_policy_instance_count
- [] list_cidr_blocks
- [] list_cidr_collections
- [] list_cidr_locations
- [] list_geo_locations
- [X] list_health_checks
- [X] list_hosted_zones
- [X] list_hosted_zones_by_name
- [X] list_hosted_zones_by_vpc
 - Pagination is not yet implemented
- [X] list_query_logging_configs Return a list of query logging configs.
- [X] list_resource_record_sets
 - The StartRecordIdentifier-parameter is not yet implemented
- [X] list_reusable_delegation_sets
 - Pagination is not yet implemented
- [X] list_tags_for_resource
- [] list_tags_for_resources
- [] list_traffic_policies
- [] list_traffic_policy_instances
- [] list_traffic_policy_instances_by_hosted_zone

- [] list_traffic_policy_instances_by_policy
- [] list_traffic_policy_versions
- [] list_vpc_association_authorizations
- [] test_dns_answer
- [X] update_health_check
- [X] update_hosted_zone_comment
- [] update_traffic_policy_comment
- [] update_traffic_policy_instance

2.8.87 route53resolver

class moto.route53resolver.models.**Route53ResolverBackend**(*region_name*, *account_id*)
Implementation of Route53Resolver APIs.

Example usage

```
@mock_route53resolver
def test_route53resolver_behaviour:
    boto3.client("route53resolver")
    ...
```

Implemented features for this service

- [] associate_firewall_rule_group
- [X] associate_resolver_endpoint_ip_address
- [] associate_resolver_query_log_config
- [X] associate_resolver_rule
- [] create_firewall_domain_list
- [] create_firewall_rule
- [] create_firewall_rule_group
- [X] create_resolver_endpoint

Return description for a newly created resolver endpoint.

NOTE: IPv6 IPs are currently not being filtered when calculating the create_resolver_endpoint() IpAddresses.

- [] create_resolver_query_log_config
- [X] create_resolver_rule Return description for a newly created resolver rule.
- [] delete_firewall_domain_list
- [] delete_firewall_rule
- [] delete_firewall_rule_group
- [X] delete_resolver_endpoint
- [] delete_resolver_query_log_config
- [X] delete_resolver_rule

- [] disassociate_firewall_rule_group
- [X] disassociate_resolver_endpoint_ip_address
- [] disassociate_resolver_query_log_config
- [X] disassociate_resolver_rule
- [] get_firewall_config
- [] get_firewall_domain_list
- [] get_firewall_rule_group
- [] get_firewall_rule_group_association
- [] get_firewall_rule_group_policy
- [] get_resolver_config
- [] get_resolver_dnssec_config
- [X] get_resolver_endpoint
- [] get_resolver_query_log_config
- [] get_resolver_query_log_config_association
- [] get_resolver_query_log_config_policy
- [X] get_resolver_rule Return info for specified resolver rule.
- [X] get_resolver_rule_association
- [] get_resolver_rule_policy
- [] import_firewall_domains
- [] list_firewall_configs
- [] list_firewall_domain_lists
- [] list_firewall_domains
- [] list_firewall_rule_group_associations
- [] list_firewall_rule_groups
- [] list_firewall_rules
- [] list_resolver_configs
- [] list_resolver_dnssec_configs
- [X] list_resolver_endpoint_ip_addresses
- [X] list_resolver_endpoints
- [] list_resolver_query_log_config_associations
- [] list_resolver_query_log_configs
- [X] list_resolver_rule_associations
- [X] list_resolver_rules
- [X] list_tags_for_resource
- [] put_firewall_rule_group_policy
- [] put_resolver_query_log_config_policy

- [] put_resolver_rule_policy
- [X] tag_resource
- [X] untag_resource
- [] update_firewall_config
- [] update_firewall_domains
- [] update_firewall_rule
- [] update_firewall_rule_group_association
- [] update_resolver_config
- [] update_resolver_dnssec_config
- [X] update_resolver_endpoint
- [] update_resolver_rule

2.8.88 s3

class moto.s3.models.S3Backend(*region_name, account_id*)

Moto implementation for S3.

Custom S3 endpoints are supported, if you are using a S3-compatible storage solution like Ceph. Example usage:

```
os.environ["MOTO_S3_CUSTOM_ENDPOINTS"] = "http://custom.internal.endpoint,http://
↳ custom.other.endpoint"
@mock_s3
def test_my_custom_endpoint():
    boto3.client("s3", endpoint_url="http://custom.internal.endpoint")
    ...
```

Note that this only works if the environment variable is set **before** the mock is initialized.

Example usage

```
@mock_s3
def test_s3_behaviour():
    boto3.client("s3")
    ...
```

Implemented features for this service

- [X] abort_multipart_upload
- [X] complete_multipart_upload
- [X] copy_object
- [X] create_bucket
- [X] create_multipart_upload
- [X] delete_bucket
- [] delete_bucket_analytics_configuration
- [X] delete_bucket_cors
- [X] delete_bucket_encryption

- [] delete_bucket_intelligent_tiering_configuration
- [] delete_bucket_inventory_configuration
- [X] delete_bucket_lifecycle
- [] delete_bucket_metrics_configuration
- [X] delete_bucket_ownership_controls
- [X] delete_bucket_policy
- [X] delete_bucket_replication
- [X] delete_bucket_tagging
- [X] delete_bucket_website
- [X] delete_object
- [X] delete_object_tagging
- [X] delete_objects
- [X] delete_public_access_block
- [] get_bucket_accelerate_configuration
- [X] get_bucket_acl
- [] get_bucket_analytics_configuration
- [X] get_bucket_cors
- [X] get_bucket_encryption
- [] get_bucket_intelligent_tiering_configuration
- [] get_bucket_inventory_configuration
- [X] get_bucket_lifecycle
- [] get_bucket_lifecycle_configuration
- [X] get_bucket_location
- [X] get_bucket_logging
- [] get_bucket_metrics_configuration
- [] get_bucket_notification
- [X] get_bucket_notification_configuration
- [X] get_bucket_ownership_controls
- [X] get_bucket_policy
- [] get_bucket_policy_status
- [X] get_bucket_replication
- [] get_bucket_request_payment
- [X] get_bucket_tagging
- [X] get_bucket_versioning
- [] get_bucket_website
- [X] get_object

- [X] `get_object_acl`
- [] `get_object_attributes`
- [X] `get_object_legal_hold`
- [X] `get_object_lock_configuration`
- [] `get_object_retention`
- [X] `get_object_tagging`
- [] `get_object_torrent`
- [X] `get_public_access_block`
- [X] `head_bucket`
- [X] `head_object`
- [] `list_bucket_analytics_configurations`
- [] `list_bucket_intelligent_tiering_configurations`
- [] `list_bucket_inventory_configurations`
- [] `list_bucket_metrics_configurations`
- [X] `list_buckets`
- [] `list_multipart_uploads`
- [X] `list_object_versions`
- [X] `list_objects`
- [X] `list_objects_v2`
- [X] `list_parts`
- [X] `put_bucket_accelerate_configuration`
- [X] `put_bucket_acl`
- [] `put_bucket_analytics_configuration`
- [X] `put_bucket_cors`
- [X] `put_bucket_encryption`
- [] `put_bucket_intelligent_tiering_configuration`
- [] `put_bucket_inventory_configuration`
- [X] `put_bucket_lifecycle`
- [] `put_bucket_lifecycle_configuration`
- [X] `put_bucket_logging`
- [] `put_bucket_metrics_configuration`
- [] `put_bucket_notification`
- [X] `put_bucket_notification_configuration`

The configuration can be persisted, but at the moment we only send notifications to the following targets:

- AWSLambda

- SQS

For the following events:

- 's3:ObjectCreated:Copy'
- 's3:ObjectCreated:Put'

- [X] put_bucket_ownership_controls
- [X] put_bucket_policy

Basic policy enforcement is in place.

Restrictions:

- Only statements with principal=* are taken into account
- Conditions are not taken into account

- [X] put_bucket_replication
- [] put_bucket_request_payment
- [X] put_bucket_tagging
- [X] put_bucket_versioning
- [] put_bucket_website
- [X] put_object
- [X] put_object_acl
- [X] put_object_legal_hold
- [X] put_object_lock_configuration
- [X] put_object_retention
- [] put_object_tagging
- [] put_public_access_block
- [] restore_object
- [] select_object_content
- [X] upload_part
- [] upload_part_copy
- [] write_get_object_response

2.8.89 s3control

Example usage

```
@mock_s3control
def test_s3control_behaviour:
    boto3.client("s3control")
    ...
```

Implemented features for this service

- [X] create_access_point

- [] create_access_point_for_object_lambda
- [] create_bucket
- [] create_job
- [] create_multi_region_access_point
- [X] delete_access_point
- [] delete_access_point_for_object_lambda
- [X] delete_access_point_policy
- [] delete_access_point_policy_for_object_lambda
- [] delete_bucket
- [] delete_bucket_lifecycle_configuration
- [] delete_bucket_policy
- [] delete_bucket_tagging
- [] delete_job_tagging
- [] delete_multi_region_access_point
- [X] delete_public_access_block
- [] delete_storage_lens_configuration
- [] delete_storage_lens_configuration_tagging
- [] describe_job
- [] describe_multi_region_access_point_operation
- [X] get_access_point
- [] get_access_point_configuration_for_object_lambda
- [] get_access_point_for_object_lambda
- [X] get_access_point_policy
- [] get_access_point_policy_for_object_lambda
- [X] get_access_point_policy_status

We assume the policy status is always public

- [] get_access_point_policy_status_for_object_lambda
- [] get_bucket
- [] get_bucket_lifecycle_configuration
- [] get_bucket_policy
- [] get_bucket_tagging
- [] get_bucket_versioning
- [] get_job_tagging
- [] get_multi_region_access_point
- [] get_multi_region_access_point_policy
- [] get_multi_region_access_point_policy_status

- [] get_multi_region_access_point_routes
- [X] get_public_access_block
- [] get_storage_lens_configuration
- [] get_storage_lens_configuration_tagging
- [] list_access_points
- [] list_access_points_for_object_lambda
- [] list_jobs
- [] list_multi_region_access_points
- [] list_regional_buckets
- [] list_storage_lens_configurations
- [] put_access_point_configuration_for_object_lambda
- [] put_access_point_policy
- [] put_access_point_policy_for_object_lambda
- [] put_bucket_lifecycle_configuration
- [] put_bucket_policy
- [] put_bucket_tagging
- [] put_bucket_versioning
- [] put_job_tagging
- [] put_multi_region_access_point_policy
- [X] put_public_access_block
- [] put_storage_lens_configuration
- [] put_storage_lens_configuration_tagging
- [] submit_multi_region_access_point_routes
- [] update_job_priority
- [] update_job_status

2.8.90 sagemaker

Example usage

```
@mock_sagemaker
def test_sagemaker_behaviour:
    boto3.client("sagemaker")
    ...
```

Implemented features for this service

- [] add_association
- [X] add_tags
- [X] associate_trial_component

- [] batch_describe_model_package
- [] create_action
- [] create_algorithm
- [] create_app
- [] create_app_image_config
- [] create_artifact
- [] create_auto_ml_job
- [] create_code_repository
- [] create_compilation_job
- [] create_context
- [] create_data_quality_job_definition
- [] create_device_fleet
- [] create_domain
- [] create_edge_deployment_plan
- [] create_edge_deployment_stage
- [] create_edge_packaging_job
- [X] create_endpoint
- [X] create_endpoint_config
- [X] create_experiment
- [] create_feature_group
- [] create_flow_definition
- [] create_hub
- [] create_human_task_ui
- [] create_hyper_parameter_tuning_job
- [] create_image
- [] create_image_version
- [] create_inference_experiment
- [] create_inference_recommendations_job
- [] create_labeling_job
- [X] create_model
- [] create_model_bias_job_definition
- [] create_model_card
- [] create_model_card_export_job
- [] create_model_explainability_job_definition
- [] create_model_package
- [] create_model_package_group

- create_model_quality_job_definition
- create_monitoring_schedule
- create_notebook_instance
- create_notebook_instance_lifecycle_config
- create_pipeline
- create_presigned_domain_url
- create_presigned_notebook_instance_url
- create_processing_job
- create_project
- create_space
- create_studio_lifecycle_config
- create_training_job
- create_transform_job
- create_trial
- create_trial_component
- create_user_profile
- create_workforce
- create_workteam
- delete_action
- delete_algorithm
- delete_app
- delete_app_image_config
- delete_artifact
- delete_association
- delete_code_repository
- delete_context
- delete_data_quality_job_definition
- delete_device_fleet
- delete_domain
- delete_edge_deployment_plan
- delete_edge_deployment_stage
- delete_endpoint
- delete_endpoint_config
- delete_experiment
- delete_feature_group
- delete_flow_definition

- delete_hub
- delete_hub_content
- delete_human_task_ui
- delete_image
- delete_image_version
- delete_inference_experiment
- delete_model
- delete_model_bias_job_definition
- delete_model_card
- delete_model_explainability_job_definition
- delete_model_package
- delete_model_package_group
- delete_model_package_group_policy
- delete_model_quality_job_definition
- delete_monitoring_schedule
- delete_notebook_instance
- delete_notebook_instance_lifecycle_config
- delete_pipeline
- delete_project
- delete_space
- delete_studio_lifecycle_config
- delete_tags
- delete_trial
- delete_trial_component
- delete_user_profile
- delete_workforce
- delete_workteam
- deregister_devices
- describe_action
- describe_algorithm
- describe_app
- describe_app_image_config
- describe_artifact
- describe_auto_ml_job
- describe_code_repository
- describe_compilation_job

- describe_context
- describe_data_quality_job_definition
- describe_device
- describe_device_fleet
- describe_domain
- describe_edge_deployment_plan
- describe_edge_packaging_job
- describe_endpoint
- describe_endpoint_config
- describe_experiment
- describe_feature_group
- describe_feature_metadata
- describe_flow_definition
- describe_hub
- describe_hub_content
- describe_human_task_ui
- describe_hyper_parameter_tuning_job
- describe_image
- describe_image_version
- describe_inference_experiment
- describe_inference_recommendations_job
- describe_labeling_job
- describe_lineage_group
- describe_model
- describe_model_bias_job_definition
- describe_model_card
- describe_model_card_export_job
- describe_model_explainability_job_definition
- describe_model_package
- describe_model_package_group
- describe_model_quality_job_definition
- describe_monitoring_schedule
- describe_notebook_instance
- describe_notebook_instance_lifecycle_config
- describe_pipeline
- describe_pipeline_definition_for_execution

- describe_pipeline_execution
- describe_processing_job
- describe_project
- describe_space
- describe_studio_lifecycle_config
- describe_subscribed_workteam
- describe_training_job
- describe_transform_job
- describe_trial
- describe_trial_component
- describe_user_profile
- describe_workforce
- describe_workteam
- disable_sagemaker_servicecatalog_portfolio
- disassociate_trial_component
- enable_sagemaker_servicecatalog_portfolio
- get_device_fleet_report
- get_lineage_group_policy
- get_model_package_group_policy
- get_sagemaker_servicecatalog_portfolio_status
- get_search_suggestions
- import_hub_content
- list_actions
- list_algorithms
- list_aliases
- list_app_image_configs
- list_apps
- list_artifacts
- list_associations
- list_auto_ml_jobs
- list_candidates_for_auto_ml_job
- list_code_repositories
- list_compilation_jobs
- list_contexts
- list_data_quality_job_definitions
- list_device_fleets

- [] list_devices
- [] list_domains
- [] list_edge_deployment_plans
- [] list_edge_packaging_jobs
- [] list_endpoint_configs
- [] list_endpoints
- [X] list_experiments
- [] list_feature_groups
- [] list_flow_definitions
- [] list_hub_content_versions
- [] list_hub_contents
- [] list_hubs
- [] list_human_task_uis
- [] list_hyper_parameter_tuning_jobs
- [] list_image_versions
- [] list_images
- [] list_inference_experiments
- [] list_inference_recommendations_job_steps
- [] list_inference_recommendations_jobs
- [] list_labeling_jobs
- [] list_labeling_jobs_for_workteam
- [] list_lineage_groups
- [] list_model_bias_job_definitions
- [] list_model_card_export_jobs
- [] list_model_card_versions
- [] list_model_cards
- [] list_model_explainability_job_definitions
- [] list_model_metadata
- [] list_model_package_groups
- [] list_model_packages
- [] list_model_quality_job_definitions
- [X] list_models
- [] list_monitoring_alert_history
- [] list_monitoring_alerts
- [] list_monitoring_executions
- [] list_monitoring_schedules

- [] list_notebook_instance_lifecycle_configs
- [] list_notebook_instances
- [] list_pipeline_execution_steps
- [X] list_pipeline_executions
- [X] list_pipeline_parameters_for_execution
- [X] list_pipelines
- [X] list_processing_jobs
- [] list_projects
- [] list_spaces
- [] list_stage_devices
- [] list_studio_lifecycle_configs
- [] list_subscribed_workteams
- [X] list_tags
- [X] list_training_jobs
- [] list_training_jobs_for_hyper_parameter_tuning_job
- [] list_transform_jobs
- [X] list_trial_components
- [X] list_trials
- [] list_user_profiles
- [] list_workforces
- [] list_workteams
- [] put_model_package_group_policy
- [] query_lineage
- [] register_devices
- [] render_ui_template
- [] retry_pipeline_execution
- [X] search
- [] send_pipeline_execution_step_failure
- [] send_pipeline_execution_step_success
- [] start_edge_deployment_stage
- [] start_inference_experiment
- [] start_monitoring_schedule
- [X] start_notebook_instance
- [X] start_pipeline_execution
- [] stop_auto_ml_job
- [] stop_compilation_job

- stop_edge_deployment_stage
- stop_edge_packaging_job
- stop_hyper_parameter_tuning_job
- stop_inference_experiment
- stop_inference_recommendations_job
- stop_labeling_job
- stop_monitoring_schedule
- stop_notebook_instance
- stop_pipeline_execution
- stop_processing_job
- stop_training_job
- stop_transform_job
- update_action
- update_app_image_config
- update_artifact
- update_code_repository
- update_context
- update_device_fleet
- update_devices
- update_domain
- update_endpoint
- update_endpoint_weights_and_capacities
- update_experiment
- update_feature_group
- update_feature_metadata
- update_hub
- update_image
- update_image_version
- update_inference_experiment
- update_model_card
- update_model_package
- update_monitoring_alert
- update_monitoring_schedule
- update_notebook_instance
- update_notebook_instance_lifecycle_config
- update_pipeline

- [] update_pipeline_execution
- [] update_project
- [] update_space
- [] update_training_job
- [] update_trial
- [] update_trial_component
- [] update_user_profile
- [] update_workforce
- [] update_workteam

2.8.91 sdb

Example usage

```
@mock_sdb
def test_sdb_behaviour:
    boto3.client("sdb")
    ...
```

Implemented features for this service

- [] batch_delete_attributes
- [] batch_put_attributes
- [X] create_domain
- [] delete_attributes
- [X] delete_domain
- [] domain_metadata
- [X] get_attributes
 - Behaviour for the consistent_read-attribute is not yet implemented
- [X] list_domains
 - The *max_number_of_domains* and *next_token* parameter have not been implemented yet - we simply return all domains.
- [X] put_attributes
 - Behaviour for the expected-attribute is not yet implemented.
- [] select

2.8.92 secretsmanager

Example usage

```
@mock_secretsmanager
def test_secretsmanager_behaviour:
    boto3.client("secretsmanager")
    ...
```

Implemented features for this service

- [X] cancel_rotate_secret
- [X] create_secret
- [] delete_resource_policy
- [X] delete_secret
- [X] describe_secret
- [X] get_random_password
- [X] get_resource_policy
- [X] get_secret_value
- [X] list_secret_version_ids
- [X] list_secrets
- [] put_resource_policy
- [X] put_secret_value
- [] remove_regions_from_replication
- [] replicate_secret_to_regions
- [X] restore_secret
- [X] rotate_secret
- [] stop_replication_to_replica
- [X] tag_resource
- [X] untag_resource
- [X] update_secret
- [X] update_secret_version_stage
- [] validate_resource_policy

2.8.93 service-quotas

class moto.servicequotas.models.**ServiceQuotasBackend**(*region_name: str, account_id: str*)
Implementation of ServiceQuotas APIs.

Example usage

```
@mock_servicequotas
def test_servicequotas_behaviour:
    boto3.client("service-quotas")
    ...
```

Implemented features for this service

- [] associate_service_quota_template
- [] delete_service_quota_increase_request_from_template
- [] disassociate_service_quota_template
- [] get_association_for_service_quota_template
- [] get_aws_default_service_quota
- [] get_requested_service_quota_change
- [X] get_service_quota
- [] get_service_quota_increase_request_from_template
- [X] list_aws_default_service_quotas

The ServiceCodes that are currently implemented are: vpc Pagination is not yet implemented.

- [] list_requested_service_quota_change_history
- [] list_requested_service_quota_change_history_by_quota
- [] list_service_quota_increase_requests_in_template
- [] list_service_quotas
- [] list_services
- [] list_tags_for_resource
- [] put_service_quota_increase_request_into_template
- [] request_service_quota_increase
- [] tag_resource
- [] untag_resource

2.8.94 servicediscovery

class moto.servicediscovery.models.**ServiceDiscoveryBackend**(*region_name, account_id*)
Implementation of ServiceDiscovery APIs.

Example usage

```
@mock_servicediscovery
def test_servicediscovery_behaviour:
    boto3.client("servicediscovery")
    ...
```

Implemented features for this service

- [X] create_http_namespace
- [X] create_private_dns_namespace
- [X] create_public_dns_namespace
- [X] create_service
- [X] delete_namespace
- [X] delete_service
- [] deregister_instance
- [] discover_instances
- [] get_instance
- [] get_instances_health_status
- [X] get_namespace
- [X] get_operation
- [X] get_service
- [] list_instances
- [X] list_namespaces
 - Pagination or the Filters-parameter is not yet implemented
- [X] list_operations
 - Pagination or the Filters-argument is not yet implemented
- [X] list_services
 - Pagination or the Filters-argument is not yet implemented
- [X] list_tags_for_resource
- [] register_instance
- [X] tag_resource
- [X] untag_resource
- [] update_http_namespace
- [] update_instance_custom_health_status
- [] update_private_dns_namespace

- [] update_public_dns_namespace
- [X] update_service

2.8.95 ses

class moto.ses.models.SESBackend(region_name, account_id)

Responsible for mocking calls to SES.

Sent messages are persisted in the backend. If you need to verify that a message was sent successfully, you can use the internal API to check:

```
from moto.core import DEFAULT_ACCOUNT_ID
from moto.ses import ses_backends
ses_backend = ses_backends[DEFAULT_ACCOUNT_ID][region]
messages = ses_backend.sent_messages # sent_messages is a List of Message objects
```

Note that, as this is an internal API, the exact format may differ per versions.

Example usage

```
@mock_ses
def test_ses_behaviour:
    boto3.client("ses")
    ...
```

Implemented features for this service

- [] clone_receipt_rule_set
- [X] create_configuration_set
- [X] create_configuration_set_event_destination
- [] create_configuration_set_tracking_options
- [] create_custom_verification_email_template
- [] create_receipt_filter
- [X] create_receipt_rule
- [X] create_receipt_rule_set
- [] create_template
- [] delete_configuration_set
- [] delete_configuration_set_event_destination
- [] delete_configuration_set_tracking_options
- [] delete_custom_verification_email_template
- [X] delete_identity
- [] delete_identity_policy
- [] delete_receipt_filter
- [] delete_receipt_rule
- [] delete_receipt_rule_set

- [] delete_template
- [] delete_verified_email_address
- [] describe_active_receipt_rule_set
- [X] describe_configuration_set
- [X] describe_receipt_rule
- [X] describe_receipt_rule_set
- [] get_account_sending_enabled
- [] get_custom_verification_email_template
- [] get_identity_dkim_attributes
- [X] get_identity_mail_from_domain_attributes
- [X] get_identity_notification_attributes
- [] get_identity_policies
- [X] get_identity_verification_attributes
- [X] get_send_quota
- [X] get_send_statistics
- [X] get_template
- [] list_configuration_sets
- [] list_custom_verification_email_templates
- [X] list_identities
- [] list_identity_policies
- [] list_receipt_filters
- [] list_receipt_rule_sets
- [X] list_templates
- [X] list_verified_email_addresses
- [] put_configuration_set_delivery_options
- [] put_identity_policy
- [] reorder_receipt_rule_set
- [] send_bounce
- [X] send_bulk_templated_email
- [] send_custom_verification_email
- [X] send_email
- [X] send_raw_email
- [X] send_templated_email
- [] set_active_receipt_rule_set
- [] set_identity_dkim_enabled
- [X] set_identity_feedback_forwarding_enabled

- [] set_identity_headers_in_notifications_enabled
- [X] set_identity_mail_from_domain
- [X] set_identity_notification_topic
- [] set_receipt_rule_position
- [] test_render_template
- [] update_account_sending_enabled
- [] update_configuration_set_event_destination
- [] update_configuration_set_reputation_metrics_enabled
- [] update_configuration_set_sending_enabled
- [] update_configuration_set_tracking_options
- [] update_custom_verification_email_template
- [X] update_receipt_rule
- [X] update_template
- [] verify_domain_dkim
- [] verify_domain_identity
- [X] verify_email_address
- [X] verify_email_identity

2.8.96 signer

class moto.signer.models.**SignerBackend**(*region_name, account_id*)
Implementation of signer APIs.

Example usage

```
@mock_signer
def test_signer_behaviour:
    boto3.client("signer")
    ...
```

Implemented features for this service

- [] add_profile_permission
- [X] cancel_signing_profile
- [] describe_signing_job
- [] get_signing_platform
- [X] get_signing_profile
- [] list_profile_permissions
- [] list_signing_jobs
- [X] list_signing_platforms

 Pagination is not yet implemented. The parameters category, partner, target are not yet implemented

- [] list_signing_profiles

- [] list_tags_for_resource
- [X] put_signing_profile

The following parameters are not yet implemented: SigningMaterial, Overrides, SigningParameters

- [] remove_profile_permission
- [] revoke_signature
- [] revoke_signing_profile
- [] start_signing_job
- [] tag_resource
- [] untag_resource

2.8.97 sns

class moto.sns.models.SNSBackend(*region_name, account_id*)

Responsible for mocking calls to SNS. Integration with SQS/HTTP/etc is supported.

Messages published to a topic are persisted in the backend. If you need to verify that a message was published successfully, you can use the internal API to check the message was published successfully:

```
from moto.core import DEFAULT_ACCOUNT_ID
from moto.sns import sns_backends
sns_backend = sns_backends[DEFAULT_ACCOUNT_ID]["us-east-1"] # Use the appropriate_
↪account/region
all_send_notifications = sns_backend.topics[topic_arn].sent_notifications
```

Note that, as this is an internal API, the exact format may differ per versions.

Example usage

```
@mock_sns
def test_sns_behaviour:
    boto3.client("sns")
    ...
```

Implemented features for this service

- [X] add_permission
- [] check_if_phone_number_is_opted_out
- [] confirm_subscription
- [X] create_platform_application
- [X] create_platform_endpoint
- [] create_sms_sandbox_phone_number
- [X] create_topic
- [X] delete_endpoint
- [X] delete_platform_application
- [] delete_sms_sandbox_phone_number
- [X] delete_topic

- [] get_data_protection_policy
- [] get_endpoint_attributes
- [] get_platform_application_attributes
- [] get_sms_attributes
- [] get_sms_sandbox_account_status
- [X] get_subscription_attributes
- [] get_topic_attributes
- [X] list_endpoints_by_platform_application
- [] list_origination_numbers
- [] list_phone_numbers_opted_out
- [X] list_platform_applications
- [] list_sms_sandbox_phone_numbers
- [X] list_subscriptions
- [] list_subscriptions_by_topic
- [X] list_tags_for_resource
- [X] list_topics
- [] opt_in_phone_number
- [X] publish
- [X] publish_batch

The MessageStructure and MessageDeduplicationId-parameters have not yet been implemented.

- [] put_data_protection_policy
- [X] remove_permission
- [X] set_endpoint_attributes
- [] set_platform_application_attributes
- [] set_sms_attributes
- [X] set_subscription_attributes
- [] set_topic_attributes
- [X] subscribe
- [X] tag_resource
- [X] unsubscribe
- [X] untag_resource
- [] verify_sms_sandbox_phone_number

2.8.98 sqs

Example usage

```
@mock_sqs
def test_sqs_behaviour:
    boto3.client("sqs")
    ...
```

Implemented features for this service

- [X] add_permission
- [X] change_message_visibility
- [] change_message_visibility_batch
- [X] create_queue
- [X] delete_message
- [] delete_message_batch
- [X] delete_queue
- [X] get_queue_attributes
- [X] get_queue_url
- [X] list_dead_letter_source_queues
- [X] list_queue_tags
- [X] list_queues
- [X] purge_queue
- [X] receive_message
- [X] remove_permission
- [X] send_message
- [X] send_message_batch
- [X] set_queue_attributes
- [X] tag_queue
- [X] untag_queue

2.8.99 ssm

class moto.ssm.models.SimpleSystemManagerBackend(*region_name*, *account_id*)

Moto supports the following default parameters out of the box:

- /aws/service/global-infrastructure/regions
- /aws/service/global-infrastructure/services

Note that these are hardcoded, so they may be out of date for new services/regions.

Integration with SecretsManager is also supported.

Example usage

```
@mock_ssm
def test_ssm_behaviour:
    boto3.client("ssm")
    ...
```

Implemented features for this service

- [X] add_tags_to_resource
- [] associate_ops_item_related_item
- [] cancel_command
- [] cancel_maintenance_window_execution
- [] create_activation
- [] create_association
- [] create_association_batch
- [X] create_document
- [X] create_maintenance_window

Creates a maintenance window. No error handling or input validation has been implemented yet.

- [] create_ops_item
 - [] create_ops_metadata
 - [] create_patch_baseline
 - [] create_resource_data_sync
 - [] delete_activation
 - [] delete_association
 - [X] delete_document
 - [] delete_inventory
 - [X] delete_maintenance_window
- Assumes the provided WindowId exists. No error handling has been implemented yet.
- [] delete_ops_metadata
 - [X] delete_parameter
 - [X] delete_parameters
 - [] delete_patch_baseline
 - [] delete_resource_data_sync
 - [] delete_resource_policy
 - [] deregister_managed_instance
 - [] deregister_patch_baseline_for_patch_group
 - [] deregister_target_from_maintenance_window
 - [] deregister_task_from_maintenance_window
 - [] describe_activations

- [] describe_association
- [] describe_association_execution_targets
- [] describe_association_executions
- [] describe_automation_executions
- [] describe_automation_step_executions
- [] describe_available_patches
- [X] describe_document
- [X] describe_document_permission

Parameters max_results, permission_type, and next_token not yet implemented

- [] describe_effective_instance_associations
- [] describe_effective_patches_for_patch_baseline
- [] describe_instance_associations_status
- [] describe_instance_information
- [] describe_instance_patch_states
- [] describe_instance_patch_states_for_patch_group
- [] describe_instance_patches
- [] describe_inventory_deletions
- [] describe_maintenance_window_execution_task_invocations
- [] describe_maintenance_window_execution_tasks
- [] describe_maintenance_window_executions
- [] describe_maintenance_window_schedule
- [] describe_maintenance_window_targets
- [] describe_maintenance_window_tasks
- [X] describe_maintenance_windows

Returns all windows. No pagination has been implemented yet. Only filtering for Name is supported.
The NextExecutionTime-field is not returned.

- [] describe_maintenance_windows_for_target
- [] describe_ops_items
- [X] describe_parameters
- [] describe_patch_baselines
- [] describe_patch_group_state
- [] describe_patch_groups
- [] describe_patch_properties
- [] describe_sessions
- [] disassociate_ops_item_related_item
- [] get_automation_execution

- [] `get_calendar_state`
- [X] `get_command_invocation`
https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_GetCommandInvocation.html
- [] `get_connection_status`
- [] `get_default_patch_baseline`
- [] `get_deployable_patch_snapshot_for_instance`
- [X] `get_document`
- [] `get_inventory`
- [] `get_inventory_schema`
- [X] `get_maintenance_window`
 The window is assumed to exist - no error handling has been implemented yet. The `NextExecutionTime`-field is not returned.
- [] `get_maintenance_window_execution`
- [] `get_maintenance_window_execution_task`
- [] `get_maintenance_window_execution_task_invocation`
- [] `get_maintenance_window_task`
- [] `get_ops_item`
- [] `get_ops_metadata`
- [] `get_ops_summary`
- [X] `get_parameter`
- [X] `get_parameter_history`
- [X] `get_parameters`
- [X] `get_parameters_by_path` Implement the `get-parameters-by-path-API` in the backend.
- [] `get_patch_baseline`
- [] `get_patch_baseline_for_patch_group`
- [] `get_resource_policies`
- [] `get_service_setting`
- [X] `label_parameter_version`
- [] `list_association_versions`
- [] `list_associations`
- [] `list_command_invocations`
- [X] `list_commands`
https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_ListCommands.html
- [] `list_compliance_items`
- [] `list_compliance_summaries`
- [] `list_document_metadata_history`

- list_document_versions
- list_documents
- list_inventory_entries
- list_ops_item_events
- list_ops_item_related_items
- list_ops_metadata
- list_resource_compliance_summaries
- list_resource_data_sync
- list_tags_for_resource
- modify_document_permission
- put_compliance_items
- put_inventory
- put_parameter
- put_resource_policy
- register_default_patch_baseline
- register_patch_baseline_for_patch_group
- register_target_with_maintenance_window
- register_task_with_maintenance_window
- remove_tags_from_resource
- reset_service_setting
- resume_session
- send_automation_signal
- send_command
- start_associations_once
- start_automation_execution
- start_change_request_execution
- start_session
- stop_automation_execution
- terminate_session
- unlabel_parameter_version
- update_association
- update_association_status
- update_document
- update_document_default_version
- update_document_metadata
- update_maintenance_window

- [] update_maintenance_window_target
- [] update_maintenance_window_task
- [] update_managed_instance_role
- [] update_ops_item
- [] update_ops_metadata
- [] update_patch_baseline
- [] update_resource_data_sync
- [] update_service_setting

2.8.100 sso-admin

class moto.ssoadmin.models.SSOAdminBackend(*region_name, account_id*)
Implementation of SSOAdmin APIs.

Example usage

```
@mock_ssoadmin
def test_ssoadmin_behaviour:
    boto3.client("sso-admin")
    ...
```

Implemented features for this service

- [] attach_customer_managed_policy_reference_to_permission_set
- [] attach_managed_policy_to_permission_set
- [X] create_account_assignment
- [] create_instance_access_control_attribute_configuration
- [X] create_permission_set
- [X] delete_account_assignment
- [] delete_inline_policy_from_permission_set
- [] delete_instance_access_control_attribute_configuration
- [X] delete_permission_set
- [] delete_permissions_boundary_from_permission_set
- [] describe_account_assignment_creation_status
- [] describe_account_assignment_deletion_status
- [] describe_instance_access_control_attribute_configuration
- [X] describe_permission_set
- [] describe_permission_set_provisioning_status
- [] detach_customer_managed_policy_reference_from_permission_set
- [] detach_managed_policy_from_permission_set
- [] get_inline_policy_for_permission_set
- [] get_permissions_boundary_for_permission_set

- [] list_account_assignment_creation_status
- [] list_account_assignment_deletion_status
- [X] list_account_assignments
 - Pagination has not yet been implemented
- [] list_accounts_for_provisioned_permission_set
- [] list_customer_managed_policy_references_in_permission_set
- [] list_instances
- [] list_managed_policies_in_permission_set
- [] list_permission_set_provisioning_status
- [X] list_permission_sets
- [] list_permission_sets_provisioned_to_account
- [] list_tags_for_resource
- [] provision_permission_set
- [] put_inline_policy_to_permission_set
- [] put_permissions_boundary_to_permission_set
- [] tag_resource
- [] untag_resource
- [] update_instance_access_control_attribute_configuration
- [X] update_permission_set

2.8.101 stepfunctions

Example usage

```
@mock_stepfunctions
def test_stepfunctions_behaviour():
    boto3.client("stepfunctions")
    ...
```

Implemented features for this service

- [] create_activity
- [X] create_state_machine
- [] delete_activity
- [X] delete_state_machine
- [] describe_activity
- [X] describe_execution
- [] describe_map_run
- [X] describe_state_machine
- [] describe_state_machine_for_execution

- [] `get_activity_task`
- [X] `get_execution_history`
- [] `list_activities`
- [X] `list_executions`
- [] `list_map_runs`
- [X] `list_state_machines`
- [X] `list_tags_for_resource`
- [] `send_task_failure`
- [] `send_task_heartbeat`
- [] `send_task_success`
- [X] `start_execution`
- [] `start_sync_execution`
- [X] `stop_execution`
- [X] `tag_resource`
- [X] `untag_resource`
- [] `update_map_run`
- [X] `update_state_machine`

2.8.102 sts

Example usage

```
@mock_sts
def test_sts_behaviour:
    boto3.client("sts")
    ...
```

Implemented features for this service

- [X] `assume_role`
 - Assume an IAM Role. Note that the role does not need to exist. The ARN can point to another account, providing an opportunity to switch accounts.
- [X] `assume_role_with_saml`
- [X] `assume_role_with_web_identity`
- [] `decode_authorization_message`
- [] `get_access_key_info`
- [X] `get_caller_identity`
- [X] `get_federation_token`
- [X] `get_session_token`

2.8.103 support

Example usage

```
@mock_support
def test_support_behaviour:
    boto3.client("support")
    ...
```

Implemented features for this service

- [] add_attachments_to_set
- [] add_communication_to_case
- [X] create_case

The IssueType-parameter is not yet implemented

- [] describe_attachment
- [X] describe_cases

The following parameters have not yet been implemented: DisplayID, AfterTime, BeforeTime, MaxResults, Language

- [] describe_communications
- [] describe_services
- [] describe_severity_levels
- [] describe_trusted_advisor_check_refresh_statuses
- [] describe_trusted_advisor_check_result
- [] describe_trusted_advisor_check_summaries
- [X] describe_trusted_advisor_checks

The Language-parameter is not yet implemented

- [X] refresh_trusted_advisor_check
- [X] resolve_case

2.8.104 swf

Example usage

```
@mock_swf
def test_swf_behaviour:
    boto3.client("swf")
    ...
```

Implemented features for this service

- [] count_closed_workflow_executions
- [] count_open_workflow_executions
- [X] count_pending_activity_tasks
- [X] count_pending_decision_tasks

- [] deprecate_activity_type
- [X] deprecate_domain
- [] deprecate_workflow_type
- [] describe_activity_type
- [X] describe_domain
- [X] describe_workflow_execution
- [] describe_workflow_type
- [] get_workflow_execution_history
- [] list_activity_types
- [X] list_closed_workflow_executions
- [X] list_domains
- [X] list_open_workflow_executions
- [] list_tags_for_resource
- [] list_workflow_types
- [X] poll_for_activity_task
- [X] poll_for_decision_task
- [X] record_activity_task_heartbeat
- [] register_activity_type
- [X] register_domain
- [] register_workflow_type
- [] request_cancel_workflow_execution
- [] respond_activity_task_canceled
- [X] respond_activity_task_completed
- [X] respond_activity_task_failed
- [X] respond_decision_task_completed
- [X] signal_workflow_execution
- [X] start_workflow_execution
- [] tag_resource
- [X] terminate_workflow_execution
- [] undeprecate_activity_type
- [X] undeprecate_domain
- [] undeprecate_workflow_type
- [] untag_resource

2.8.105 textract

class moto.textract.models.**TextextractBackend**(*region_name, account_id*)
Implementation of Textract APIs.

Example usage

```
@mock_textract
def test_textract_behaviour:
    boto3.client("textract")
    ...
```

Implemented features for this service

- [] analyze_document
- [] analyze_expense
- [] analyze_id
- [] detect_document_text
- [] get_document_analysis
- [X] get_document_text_detection

 Pagination has not yet been implemented

- [] get_expense_analysis
- [] get_lending_analysis
- [] get_lending_analysis_summary
- [] start_document_analysis
- [X] start_document_text_detection

 The following parameters have not yet been implemented: ClientRequestToken, JobTag, Notification-Channel, OutputConfig, KmsKeyID

- [] start_expense_analysis
- [] start_lending_analysis

2.8.106 timestream-write

class moto.timestreamwrite.models.**TimestreamWriteBackend**(*region_name, account_id*)
When using the decorators, you can use the following internal API to verify records have arrived:

```
from moto.core import DEFAULT_ACCOUNT_ID as ACCOUNT_ID
from moto.timestreamwrite.models import timestreamwrite_backends

backend = timestreamwrite_backends[ACCOUNT_ID]["us-east-1"]
records = backend.databases["mydatabase"].tables["mytable"].records
```

Example usage

```
@mock_timestreamwrite
def test_timestreamwrite_behaviour:
    boto3.client("timestream-write")
    ...
```

Implemented features for this service

- [X] create_database
- [X] create_table
- [X] delete_database
- [X] delete_table
- [X] describe_database
- [X] describe_endpoints
- [X] describe_table
- [X] list_databases
- [X] list_tables
- [X] list_tags_for_resource
- [X] tag_resource
- [X] untag_resource
- [X] update_database
- [X] update_table
- [X] write_records

2.8.107 transcribe

Example usage

```
@mock_transcribe
def test_transcribe_behaviour:
    boto3.client("transcribe")
    ...
```

Implemented features for this service

- [] create_call_analytics_category
- [] create_language_model
- [X] create_medical_vocabulary
- [X] create_vocabulary
- [] create_vocabulary_filter
- [] delete_call_analytics_category
- [] delete_call_analytics_job
- [] delete_language_model
- [X] delete_medical_transcription_job

- delete_medical_vocabulary
- delete_transcription_job
- delete_vocabulary
- delete_vocabulary_filter
- describe_language_model
- get_call_analytics_category
- get_call_analytics_job
- get_medical_transcription_job
- get_medical_vocabulary
- get_transcription_job
- get_vocabulary
- get_vocabulary_filter
- list_call_analytics_categories
- list_call_analytics_jobs
- list_language_models
- list_medical_transcription_jobs
- list_medical_vocabularies
- list_tags_for_resource
- list_transcription_jobs
- list_vocabularies
- list_vocabulary_filters
- start_call_analytics_job
- start_medical_transcription_job
- start_transcription_job
- tag_resource
- untag_resource
- update_call_analytics_category
- update_medical_vocabulary
- update_vocabulary
- update_vocabulary_filter

2.8.108 wafv2

`class moto.wafv2.models.WAFV2Backend(region_name, account_id)`
https://docs.aws.amazon.com/waf/latest/APIReference/API_Operations_AWS_WAFV2.html

Example usage

```
@mock_wafv2
def test_wafv2_behaviour:
    boto3.client("wafv2")
    ...
```

Implemented features for this service

- [X] `associate_web_acl`

Only APIGateway Stages can be associated at the moment.

- [] `check_capacity`
- [] `create_ip_set`
- [] `create_regex_pattern_set`
- [] `create_rule_group`
- [X] `create_web_acl`

The following parameters are not yet implemented: CustomResponseBodies, CaptchaConfig

- [] `delete_firewall_manager_rule_groups`
- [] `delete_ip_set`
- [] `delete_logging_configuration`
- [] `delete_permission_policy`
- [] `delete_regex_pattern_set`
- [] `delete_rule_group`
- [X] `delete_web_acl`

The LockToken-parameter is not yet implemented

- [] `describe_managed_rule_group`
- [X] `disassociate_web_acl`
- [] `generate_mobile_sdk_release_url`
- [] `get_ip_set`
- [] `get_logging_configuration`
- [] `get_managed_rule_set`
- [] `get_mobile_sdk_release`
- [] `get_permission_policy`
- [] `get_rate_based_statement_managed_keys`
- [] `get_regex_pattern_set`
- [] `get_rule_group`

- [] `get_sampled_requests`
- [X] `get_web_acl`
- [X] `get_web_acl_for_resource`
- [] `list_available_managed_rule_group_versions`
- [] `list_available_managed_rule_groups`
- [] `list_ip_sets`
- [] `list_logging_configurations`
- [] `list_managed_rule_sets`
- [] `list_mobile_sdk_releases`
- [] `list_regex_pattern_sets`
- [] `list_resources_for_web_acl`
- [X] `list_rule_groups`
- [X] `list_tags_for_resource`

 Pagination is not yet implemented

- [X] `list_web_acls`
- [] `put_logging_configuration`
- [] `put_managed_rule_set_versions`
- [] `put_permission_policy`
- [X] `tag_resource`
- [X] `untag_resource`
- [] `update_ip_set`
- [] `update_managed_rule_set_version_expiry_date`
- [] `update_regex_pattern_set`
- [] `update_rule_group`
- [X] `update_web_acl`

 The following parameters are not yet implemented: `LockToken`, `CustomResponseBodies`, `Captcha-Config`

2.9 Contributing

Want to contribute to Moto? Yes please! The aim is to mock as many services and features as possible. With AWS releasing new services almost daily, that is for sure a moving target! So any help is welcome.

Check our [Development Installation](#) guide on how to install Moto locally. The [Architecture](#) page can be useful or interesting if you want to dive deeper in Moto's guts, but feel free to skip this page if you're not interested.

Want to add some new features, or improving the behaviour of existing ones? [New Features](#) is for you.

We also have a collection of [Development Tips](#) to give some guidance.

Finally, the [FAQ](#) page may have some questions to your burning answers that popped up while reading this saga.

2.10 Development Installation

This is a guide how to install Moto for contributors.

The following software is assumed to be present:

- Python 3.x
- Docker
- Git

2.10.1 Checking out the code

Contributing to Moto involves forking the project. GitHub has a handy guide explaining how to do this: <https://docs.github.com/en/get-started/quickstart/contributing-to-projects>

2.10.2 Installing Moto locally

It is recommended to work from some kind of virtual environment, i.e. *virtualenv*, to prevent cross-contamination with other projects. From within such a virtualenv, run the following command to install all required dependencies:

```
make init
```

With all dependencies installed, run the following command to run all the tests and verify your environment is ready:

```
make test
```

Note that this may take awhile - there are many services, and each service will have a boatload of tests.

To verify all tests pass for a specific service, for example for *s3*, run these commands manually:

```
flake8 moto/s3
black --check moto/s3 tests/test_s3
pylint tests/test_s3
pytest -sv tests/test_s3
```

If black fails, you can run the following command to automatically format the offending files:

```
make format
```

If any of these steps fail, please see our [FAQ](#) or open an issue on Github.

2.11 Architecture

If you're interested in the inner workings of Moto, or are trying to hunt down some tricky bug, the below sections will help you learn more about how Moto works.

2.11.1 Decorator Architecture

When using decorators, Moto works by intercepting the HTTP request that is sent out by boto3. This has multiple benefits:

- boto3 keeps the responsibility of any initial parameter validation
- boto3 keeps the responsibility of any post-processing of the response
- Other SDK's can also be used against Moto, as all SDK's use the HTTP API to talk to AWS in the end.

Botocore utilizes an event-based architecture. Events such as *creating-client-class* and *before-send* are emitted for all boto3-requests.

When the decorator starts, Moto registers a hook into the *before-send*-event that allows us to intercept the HTTP-request that was about to be sent. For every intercepted request, Moto figures out which service/feature is called based on the HTTP request prepared by *boto3*, and calls our own stub instead.

2.11.2 Determining which service/feature is called

There are multiple ways for Moto to determine which request was called. For each request we need to know two things:

1. Which service is this request for?
2. Which feature is called?

When using one or more decorators, Moto will load all urls from *{service}/urls.py:url_bases*. Incoming requests are matched against those to figure out which service the request has to go to. After that, we try to find right feature by looking at:

1. The *Action*-parameter in the querystring or body, or
2. The *x-amz-target*-header, or
3. The full URI. Boto3 has a model for each service that maps the HTTP method and URI to a feature. Note that this only works if the *Responses*-class has an attribute *SERVICE_NAME*, as Moto needs to know which boto3-client has this model.

When using Moto in ServerMode, all incoming requests will be made to *http://localhost*, or wherever the MotoServer is hosted, so we can no longer use the request URI to figure out which service to talk to. In order to still know which service the request was intended for, Moto will check:

1. The authorization header first (*HTTP_AUTHORIZATION*)
2. The target header next (*HTTP_X_AMZ_TARGET*)
3. Or the path header (*PATH_INFO*)
4. If all else fails, we assume S3 as the default

Now that we have determined the service, we can rebuild the original host this request was sent to. With the combination of the restored host and path we can match against the *url_bases* and *url_paths* in *{service}/urls.py* to determine which Moto-method is responsible for handling the incoming request.

2.11.3 File Architecture

To keep a logical separation between services, each one is located into a separate folder. Each service follows the same file structure.

The below table explains the purpose of each file:

File	Responsibility
<code>__init__.py</code>	Initializes the decorator to be used by developers
<code>urls.py</code>	List of the URL's that should be intercepted
<code>re-sponses.py</code>	Requests are redirected here first. Responsible for extracting parameters and determining the response format
<code>models.py</code>	Responsible for the data storage and logic required.
<code>exceptions.py</code>	Not required - this would contain any custom exceptions, if your code throws any

2.12 New Features

Moto has a script that can automatically provide the scaffolding for a new service, and for adding new features to an existing service. This script does all the heavy lifting of generating template code, by looking up the API specification of a given *boto3* method and adding the necessary code to mock it.

Please try it out by running:

```
export AWS_DEFAULT_REGION="us-east-2"
python scripts/scaffold.py
```

The script uses the *click*-module to assists with autocompletion.

- Use Tab to auto-complete the first suggest service, or
- Use the up and down-arrows on the keyboard to select something from the dropdown
- Press enter to continue

An example interaction:

```
$ python scripts/scaffold.py
Select service: codedeploy

==Current Implementation Status==
[ ] add_tags_to_on_premises_instances
...
[ ] create_deployment
...
[ ] update_deployment_group
=====
Select Operation: create_deployment

    Initializing service    codedeploy
    creating                moto/codedeploy
    creating                moto/codedeploy/models.py
    creating                moto/codedeploy/exceptions.py
```

(continues on next page)

(continued from previous page)

```
creating      moto/codedeploy/__init__.py
creating      moto/codedeploy/responses.py
creating      moto/codedeploy/urls.py
creating      tests/test_codedeploy
creating      tests/test_codedeploy/test_server.py
creating      tests/test_codedeploy/test_codedeploy.py
inserting code moto/codedeploy/responses.py
inserting code moto/codedeploy/models.py
```

Remaining steps after development is complete:

- Run `scripts/implementation_coverage.py`,
- Run `scripts/update_backend_index.py`.

Note: The implementation coverage script is used to automatically update the full list of supported services.

Warning: In order to speed up the performance of MotoServer, all AWS URL's that need intercepting are indexed. When adding/replacing any URLs in `{service}/urls.py`, please run `python scripts/update_backend_index.py` to update this index.

2.13 PR Checklist

Ready to open a new PR? Great. Please have a quick look at this checklist to make sure that you're ready:

- Feature is added
- The linter is happy
- Tests are added
- All tests pass, existing and new

Can't quite get it working? Create the PR anyway! Seeing the code can help others figure out what's wrong.

Halfway through and too busy to finish? Create the PR anyway! Others can use your work and build on it to finish the feature.

Note: You can indicate a PR that's not quite ready with the `needs-help`-label.

Are you not sure on what you want to implement, or want some advice on how to approach things? Feel free to open a new issue on Github: <https://github.com/getmoto/moto/issues>

2.14 FAQ for Developers

2.14.1 When running the linter...

Why does black give different results?

Different versions of black produce different results. The CI system uses the version set in *requirements-dev.txt*. To ensure that our CI passes, please format the code using the same version.

2.14.2 When running a test...

Why does it take ages to run a single test?

There are a few reasons why this could happen. If the test uses Docker, it could take a while to:

- Download the appropriate Docker image
- Run the image
- Wait for the logs to appear

Why are my tests failing in ServerMode?

- Make sure that the correct url paths are present in *urls.py*
- Make sure that you've run *scripts/update_backend_index.py* afterwards, to let MotoServer know the urls have changed.

2.14.3 What ...

Does ServerMode refer to?

ServerMode refers to Moto running as a stand-alone server. This can be useful to:

- Test non-Python SDK's
- Have a semi-permanent, local AWS-like server running that multiple applications can talk to

Types of tests are there?

There are three types of tests:

1. decorator tests
2. ServerMode tests
3. server tests (located in *test_server.py*)

The decorator tests refer to the normal unit tests that are run against an in-memory Moto instance.

The ServerMode tests refer to the same set of tests - but run against an external MotoServer instance. When running tests in ServerMode, each *boto3-client* and *boto3-resource* are intercepted, and enriched with the `endpoint_url="http://localhost:5000"` argument. This allows us to run the same test twice, and verify that Moto behaves the same when using decorators, and in ServerMode.

The last ‘server’ tests are low-level tests that can be used to verify that Moto behaves exactly like the AWS HTTP API. Each test will spin up the MotoServer in memory, and run HTTP requests directly against that server. This allows the developer to test things like HTTP headers, the exact response/request format, etc.

Alternatives are there?

The best alternative would be [LocalStack](#).

LocalStack is Moto’s bigger brother with more advanced features, such as EC2 VM’s that you can SSH into and Dockerized RDS-installations that you can connect to.

2.14.4 Why am I getting errors...

That my new module could not be found when using the scaffold-script?

```
File "scripts/scaffold.py", line 499, in insert_codes
    insert_code_to_class(responses_path, BaseResponse, func_in_responses)
File "scripts/scaffold.py", line 424, in insert_code_to_class
    mod = importlib.import_module(mod_path)
File "/usr/lib/python3.8/importlib/__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)

ModuleNotFoundError: No module named 'moto.kafka'
```

This will happen if you’ve ran `pip install .` prior to running `scripts/scaffold.py`.

Instead, install Moto as an editable module instead:

```
pip uninstall moto
pip install -e .
```

Related to Docker when running tests?

AWSLambda and Batch services use Docker to execute the code provided to the system, which means that Docker needs to be installed on your system in order for these tests to run.

Installing Moto using ZSH on MacOS?

When using `pip install` on ZSH, you might see the following: `zsh: no matches found`. This is because ZSH requires the full module to be in quotes.

```
pip install "moto[ssm]"
```

2.15 Development Tips

Below you can find some tips that might help during development.

2.15.1 Naming Conventions

Let's say you want to implement the *import_certificate* feature for the ACM service.

Implementing the feature itself can be done by creating a method called *import_certificate* in *moto/acm/responses.py*. It's considered good practice to deal with input/output formatting and validation in *responses.py*, and create a method *import_certificate* in *moto/acm/models.py* that handles the actual import logic.

When writing tests, you'll want to add a new method called *def test_import_certificate* to *tests/test_acm/test_acm.py*. Additional tests should also have names indicate of what's happening, i.e. *def test_import_certificate_fails_without_name*, *def test_import_existing_certificate*, etc.

Partial Implementations

If a service is only partially implemented, a warning can be used to inform the user:

```
import warnings
warnings.warn("The Filters-parameter is not yet implemented for client.method()")
```

2.16 Intercepting URLs

2.16.1 Determining which URLs to intercept

In order for Moto to know which requests to intercept, Moto needs to know which URLs to intercept. But how do we know which URL's should be intercepted? There are a few ways of doing it:

- For an existing service, copy/paste the url-path for an existing feature and cross your fingers and toes
- Use the service model that is used by botocore: <https://github.com/boto/botocore/tree/develop/botocore/data> Look for the *requestUri*-field in the *services.json* file.
- Make a call to AWS itself, and intercept the request using a proxy. This gives you all information you could need, including the URL, parameters, request and response format.

Intercepting AWS requests

Download and install a proxy such [MITMProxy](#).

With the proxy running, the easiest way of proxying requests to AWS is probably via the CLI.

```
export HTTP_PROXY=http://localhost:8080
export HTTPS_PROXY=http://localhost:8080
aws ses describe-rule-set --no-verify-ssl
```

```
from botocore.config import Config
proxy_config = Config(proxies={'http': 'localhost:8080', 'https': 'localhost:8080'})
boto3.client("ses", config=proxy_config, use_ssl=False, verify=False)
```

2.17 Writing tests

One test should only verify a single feature/method. I.e., one test for `create_resource()`, another for `update_resource()`, etc.

2.17.1 Negative tests

When writing negative tests, try to use the following format:

```
with pytest.raises(botocore.exceptions.ClientError) as exc:
    client.failing_call(..)
err = exc.value.response["Error"]
# These assertions use the 'sure' library, but any assertion style is accepted
err["Code"].should.equal(..)
err["Message"].should.equal(..)
```

This is the best way to ensure that exceptions are dealt with correctly by Moto.

2.17.2 ServerMode tests

Our CI runs all tests twice - one normal run, and one run in ServerMode. In ServerMode, Moto is started as a stand-alone Flask server, and all tests are run against this Flask-instance.

To verify whether your tests pass in ServerMode, you can run the following commands:

```
moto_server
TEST_SERVER_MODE=true pytest -sv tests/test_service/..
```

2.17.3 Parallel tests

To speed up our CI, the tests for the `awslambda`, `batch`, `ec2` and `sqs` services will run in parallel. This means the following:

- Make sure you use unique names for functions/queues/etc
- Calls to `describe_reservations()/list_queues()/etc` might return resources from other tests

2.17.4 Terraform tests

To verify that Moto behaves correctly, we run a subset of Terraform's tests against the MotoServer to ensure it behaves the same as AWS does.

These tests will be run automatically for every PR, so you should not need to make any changes here.

A list of which tests currently pass against Moto can be found in `tests/terraformtests/terraform-tests.success.txt`.

Use the following commands to see the full list of available tests:

```
cd tests/terraformtests/terraform-provider-aws
# Choose the correct service in the next command - this example will list all tests for
↳ the ELB-service
go test ./internal/service/elb/ -v -list TestAcc
```


In order to check whether MotoServer behaves correctly against a specific test, you can use the following commands:

```
# Ensure you are back in the root-directory
# Start the MotoServer on port 4566
moto_server -p 4566
# Run the new tests
make terraformtests SERVICE_NAME=elb TEST_NAMES=NewTestName
```

2.18 Utilities

2.18.1 Tagging Service

A dedicated TaggingService exists in *moto.utilities*, to help with storing/retrieving tags for resources.

Not all services use it yet, but contributors are encouraged to use the TaggingService for all new features.

2.18.2 Paginator

When requesting a list of resources, almost all AWS services use pagination to stagger the response. Moto provides a utility class to automatically paginate a response, without having to manually write the logic.

There are three components that make this work:

1. The Responses-method will call the backend with a `max_result/next_token` parameter
2. The backend-method has a plain method that's decorated with `@paginate`
3. A configuration model is supplied to the decorator that contains all the details.

See the below example how it works in practice:

```
class MyResponse(BaseResponse):

    # The Response-class looks like any other - read the input parameters, and call the
    ↪backend to retrieve the resources
    def list_resources():
        max_results = 100
        next_token = self._get_param("NextToken")
        # Note that we're getting both the results and the next_token
        # The decorator in the backend returns this tuple
        paged_results, next_token = self.backend.list_resources(
            max_results=max_results, next_token=next_token
        )
        ...

from moto.utilities.paginator import paginate
class MyBackend(BaseBackend):

    # The model that contains the configuration required for the paginator
    PAGINATION_MODEL = {
        # key = name of the method in the backend
        "list_resources": {
            #
```

(continues on next page)

(continued from previous page)

```

        # name of the kwarg that contains the next token, which should be passed to.
↳the backend
        # backend.list_resources(next_token=..)
        "input_token": "next_token",
        #
        # name of the kwarg that contains the max number of results, which should be.
↳passed to the backend
        "limit_key": "max_results",
        #
        # The default limit of the above parameter is not provided
        "limit_default": 100,
        #
        # One or more attributes that guarantee uniqueness for a given resource.
        # For most resources it will just be an ID, or ARN, which is always unique.
        # An encoded version of these attributes is used as the NextToken.
        "unique_attribute": "arn",
        # Provide a list if only a combination of attributes is guaranteed to be.
↳unique
        "unique_attribute": ["start_date", "execution_arn"],
        #
        # By default, an exception will be thrown if the user-provided next_token is.
↳invalid
        "fail_on_invalid_token": True # Default value - no need to specify this
        # This can be customized to:
        # - silently fail, and just return an empty list
        "fail_on_invalid_token": False,
        # - throw a custom exception, by providing an exception class
        # The paginator will `raise CustomException()` or `raise.
↳CustomException(invalid_token)`
        "fail_on_invalid_token": CustomException
    },
    # another method that will use different configuration options
    "list_other_things": {
        ...
    },
}

# The decorator with the pagination logic
@paginate(pagination_model=PAGINATION_MODEL)
# Note that this method does not have the 'next_token'/'max_results'-arguments
def list_resources(self):
    # Note that we simply return all resources - the decorator takes care of all.
↳pagination magic
    return self.full_list_of_resources

@paginate(pagination_model=PAGINATION_MODEL)
# If we do need the 'next_token'/'max_results'-arguments, just add them to the function
# The decorator will only pass them along if required
def list_other_things(self, max_results=None):
    if max_results == "42":
        # Custom validation magic
        pass

```

(continues on next page)

(continued from previous page)

```
return self.full_list_of_resources
```

2.19 State Transition Management

When developing a model where the resource is not available immediately, such as EC2 instances, a configuration option is available to specify whether you want mocked resources to be available immediately (to speed up unit testing), or whether you want an artificial delay to more closely mimick AWS' behaviour where resources are only available/ready after some time.

See the user-documentation here: [State Transitions](#)

In order for a new model to support this behaviour out of the box, it needs to be configured and registered with the State Manager. The following steps need to be taken for this to be effective:

- Extend the new model with the ManagedState-class
- Call the ManagedState-constructor with information on which state transitions are supported
- Decide when to advance the status
- Register the model with the StateManager

An example model could look like this:

```
from moto.moto_api._internal.managed_state_model import ManagedState

class NewModel(ManagedState):
    def __init__(self):
        ManagedState.__init__(self,
                               # A unique name should be chosen to uniquely identify this_
                               ↪model
                               # Any name is acceptable - a typical format would be
                               ↪'API:type'
                               # Examples: 'S3::bucket', 'APIGateway::Method',
                               ↪'DynamoDB::Table'
                               model_name="new::model",
                               # List all the possible status-transitions here
                               transitions=[("initializing", "starting"),
                                           ("starting", "ready")])

    def to_json(self):
        # ManagedState gives us a 'status'-attribute out of the box
        # On the first iteration, this will be set to the first status of the first_
        ↪transition
        return {
            "name": ...,
            "status": self.status,
            ...
        }

from moto.moto_api import state_manager

class Backend():
```

(continues on next page)

```
def __init__():
    # This is how we register the model, and specify the default transition-behaviour
    # Typically this is done when constructing the Backend-class
    state_manager.register_default_transition(
        # This name should be the same as the name used in NewModel
        model_name="new:model",
        # Any transition-config is possible - this is a good default option though
        transition={"progression": "immediate"},
    )

def list_resources():
    for ec2_instance in all_resources:
        # For users who configured models of this type to transition manually, this
        ↪ is where we advance the status
        # Say the transition is registered like so: {"progression": "manual", "times
        ↪ ": 3}
        #
        # The user calls 'list_resources' 3 times, the advance-method is called 3
        ↪ times, and the state manager advances the state after the 3rd time.
        # This all happens out of the box - just make sure that the `advance()`-
        ↪ method is invoked when appropriate
        #
        # If the transition is set to progress immediately, this method does exactly
        ↪ nothing.
        #
        # If the user decides to change the progression to be time-based, where the
        ↪ status changed every y seconds, this method does exactly nothing.
        # It will has to be called though, for people who do have the manual
        ↪ progression configured
        model.advance()
    return all_models

def describe_resource():
    resource = ...
    # Depending on the API, there may be different ways for the user to retrieve the
    ↪ same information
    # Make sure that each way (describe, list, get_, ) calls the advance()-method,
    ↪ and the resource can actually progress to the next state
    resource.advance()
    return resource
```

A

ACMPCABackend (class in *moto.acmpca.models*), 30
 APIGatewayBackend (class in *moto.apigateway.models*),
 32
 ApiGatewayV2Backend (class in
moto.apigatewayv2.models), 36
 AppSyncBackend (class in *moto.appsync.models*), 39

B

BatchBackend (class in *moto.batch.models*), 45
 BudgetsBackend (class in *moto.budgets.models*), 46

C

CloudFormationBackend (class in
moto.cloudformation.models), 49
 CloudTrailBackend (class in *moto.cloudtrail.models*),
 54
 CognitoIdpBackend (class in *moto.cognitoidp.models*),
 63
 ComprehendBackend (class in
moto.comprehend.models), 67
 CostExplorerBackend (class in *moto.ce.models*), 47

D

DirectoryServiceBackend (class in *moto.ds.models*),
 80

E

EBSBackend (class in *moto.ebs.models*), 84
 EC2Backend (class in *moto.ec2.models*), 84
 EC2ContainerServiceBackend (class in
moto.ecs.models), 103
 EFSBackend (class in *moto.efs.models*), 106
 ElastiCacheBackend (class
moto.elasticache.models), 108
 ElasticsearchServiceBackend (class in
moto.es.models), 118
 EMRContainersBackend (class
moto.emrcontainers.models), 117
 EMRServerlessBackend (class
moto.emrserverless.models), 118
 EventsBackend (class in *moto.events.models*), 120

F

FirehoseBackend (class in *moto.firehose.models*), 123

L

LambdaBackend (class in *moto.awslambda.models*), 155

M

MQBackend (class in *moto.mq.models*), 166

P

PersonalizeBackend (class in
moto.personalize.models), 172
 PinpointBackend (class in *moto.pinpoint.models*), 174
 PrometheusServiceBackend (class in
moto.amp.models), 31

Q

QuickSightBackend (class in *moto.quicksight.models*),
 178

R

RekognitionBackend (class in
moto.rekognition.models), 191
 Route53ResolverBackend (class in
moto.route53resolver.models), 197

S

S3Backend (class in *moto.s3.models*), 199
 ServiceDiscoveryBackend (class in
moto.servicediscovery.models), 216
 ServiceQuotasBackend (class in
moto.servicequotas.models), 215
 SESBackend (class in *moto.ses.models*), 217
 SignerBackend (class in *moto.signer.models*), 219
 SimpleSystemManagerBackend (class in
moto.ssm.models), 222
 SNSBackend (class in *moto.sns.models*), 220
 SSOAdminBackend (class in *moto.ssoadmin.models*), 227

T

TextractBackend (class in *moto.textract.models*), 232

TimestreamWriteBackend (class in *moto.timestreamwrite.models*), 232

W

WAFV2Backend (class in *moto.wafv2.models*), 235