
Moto Documentation

Release 2.2.16.dev

Steve Pulec

Nov 21, 2021

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	10
2.3	FAQ	13
2.4	Boto vs Boto3	13
2.5	IAM-like Access Control	14
2.6	AWS Config Support	15
2.7	Implemented Services	17
2.8	Contributing	157
2.9	Installation	157
2.10	Architecture	158
2.11	New Features	159
2.12	PR Checklist	160
2.13	FAQ for Developers	161
2.14	Development Tips	163
2.15	Intercepting URL's	163
2.16	Writing tests	164
2.17	Utilities	165
	Index	167

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/spulec/moto.git
cd moto
python setup.py install
```

2.1.2 Moto usage

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

```
import boto3

class MyModel(object):
    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)

```

It is possible to use Moto as a class-decorator. Note that this may behave differently then you might expected - it currently creates a global state on class-level, rather than on method-level.

```
@mock_s3
class TestMockClassLevel(unittest.TestCase):
    def create_my_bucket(self):
        s3 = boto3.resource('s3')
        bucket = s3.Bucket("mybucket")
        bucket.create()

    def test_1_should_create_bucket(self):
        self.create_my_bucket()

        client = boto3.client("s3")
        assert len(client.list_buckets()["Buckets"]) == 1

    def test_2_bucket_still_exists(self):
        client = boto3.client("s3")
        assert len(client.list_buckets()["Buckets"]) == 1
```

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'
```

2. **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.

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'

@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")

    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..
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 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.

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.

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

2.2.1 Examples

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
```

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](#)

2.2.2 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`.

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=true
```

2.2.3 Dashboard

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

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

2.2.4 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")
```

2.2.5 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
```


2.2.6 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>

2.4 Boto vs Boto3

Boto3 is the latest Python SDK, and as such the SDK targeted by Moto. All our *@mock_*-decorators should be usable against any boto3-version.

Still stuck on boto, the former SDK? Moto does have some support, in the form of our deprecated services:

```
from moto import mock_ec2_deprecated
import boto

@mock_ec2_deprecated
def test_something_with_ec2():
    ec2_conn = boto.ec2.connect_to_region('us-east-1')
    ec2_conn.get_only_instances(instance_ids='i-123456')
```

When using both boto2 and boto3, one can do this to avoid confusion:

```
from moto import mock_ec2_deprecated as mock_ec2_b2
from moto import mock_ec2
```

If you want to use Server Mode, the easiest way is to create a boto config file (*~/.boto*) with the following values:

```
[Boto]
is_secure = False
https_validate_certificates = False
proxy_port = 5000
proxy = 127.0.0.1
```

2.5 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.

2.5.1 `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.

2.5.2 `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.
    # 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.

2.6 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.

2.6.1 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.

2.6.2 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)

2.6.3 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](#).

2.7 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():
    ...
```

2.7.1 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
- [] resend_validation_email
- [] update_certificate_options

2.7.2 apigateway

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

- [X] `get_authorizer`
- [X] `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`
- [] `get_gateway_response`
- [] `get_gateway_responses`
- [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`
- [] `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
- [] get_vpc_link
- [] get_vpc_links
- [] import_api_keys
- [] import_documentation_parts
- [] import_rest_api
- [] put_gateway_response
- [] put_integration
- [] put_integration_response
- [] put_method
- [] put_method_response
- [] put_rest_api
- [] tag_resource
- [] test_invoke_authorizer
- [] test_invoke_method
- [] untag_resource
- [] update_account
- [X] update_api_key
- [X] update_authorizer
- [] update_base_path_mapping
- [] update_client_certificate
- [] update_deployment
- [] update_documentation_part
- [] update_documentation_version
- [X] update_domain_name
- [] update_gateway_response
- [] update_integration
- [] update_integration_response
- [X] update_method
- [X] 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

2.7.3 application-autoscaling

Example usage

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

Implemented features for this service

- [X] delete_scaling_policy
- [] 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
- [] describe_scheduled_actions
- [X] put_scaling_policy
- [] put_scheduled_action
- [X] register_scalable_target Registers or updates a scalable target.

2.7.4 athena

Example usage

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

Implemented features for this service

- [] batch_get_named_query
- [] batch_get_query_execution
- [] create_data_catalog
- [X] create_named_query
- [] create_prepared_statement

- [X] create_work_group
- [] delete_data_catalog
- [] delete_named_query
- [] delete_prepared_statement
- [] delete_work_group
- [] get_data_catalog
- [] get_database
- [X] get_named_query
- [] get_prepared_statement
- [] get_query_execution
- [] get_query_results
- [] get_table_metadata
- [X] get_work_group
- [] list_data_catalogs
- [] list_databases
- [] list_engine_versions
- [] list_named_queries
- [] list_prepared_statements
- [] list_query_executions
- [] list_table_metadata
- [] list_tags_for_resource
- [X] list_work_groups
- [X] start_query_execution
- [X] stop_query_execution
- [] tag_resource
- [] untag_resource
- [] update_data_catalog
- [] update_prepared_statement
- [] update_work_group

2.7.5 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
- [] 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
- [] 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
- [] describe_scheduled_actions
- [] describe_tags
- [] describe_termination_policy_types
- [] describe_warm_pool
- [X] detach_instances
- [X] detach_load_balancer_target_groups
- [X] detach_load_balancers
- [] disable_metrics_collection
- [] enable_metrics_collection
- [] enter_standby
- [X] execute_policy
- [] exit_standby
- [] get_predictive_scaling_forecast
- [] put_lifecycle_hook
- [] put_notification_configuration
- [] put_scaling_policy
- [] put_scheduled_update_group_action
- [] put_warm_pool
- [] record_lifecycle_action_heartbeat
- [X] resume_processes
- [X] set_desired_capacity
- [X] set_instance_health
- [X] set_instance_protection
- [] start_instance_refresh
- [X] suspend_processes
- [] terminate_instance_in_auto_scaling_group
- [X] update_auto_scaling_group

2.7.6 batch

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

Create a job queue

param queue_name Queue name

type queue_name str

param priority Queue priority

type priority int

param state Queue state

type state string

param compute_env_order Compute environment list

type compute_env_order list of dict

return Tuple of Name, ARN

rtype tuple of str

- [] create_scheduling_policy
- [X] delete_compute_environment
- [X] delete_job_queue
- [] delete_scheduling_policy
- [X] deregister_job_definition
- [X] describe_compute_environments
- [X] describe_job_definitions
- [X] describe_job_queues
- [X] describe_jobs
- [] describe_scheduling_policies
- [X] list_jobs
- [] list_scheduling_policies
- [] list_tags_for_resource
- [X] register_job_definition
- [X] submit_job

- [] tag_resource
- [X] terminate_job
- [] untag_resource
- [X] update_compute_environment
- [X] update_job_queue

Update a job queue

param queue_name Queue name

type queue_name str

param priority Queue priority

type priority int

param state Queue state

type state string

param compute_env_order Compute environment list

type compute_env_order list of dict

return Tuple of Name, ARN

rtype tuple of str

- [] update_scheduling_policy

2.7.7 cloudformation

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
- [X] create_stack_instances
- [X] create_stack_set
- [] deactivate_type
- [X] delete_change_set
- [X] delete_stack

- [X] delete_stack_instances
- [X] delete_stack_set
- [] deregister_type
- [] describe_account_limits
- [X] describe_change_set
- [] 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
- [X] describe_stacks
- [] describe_type
- [] describe_type_registration
- [] detect_stack_drift
- [] detect_stack_resource_drift
- [] detect_stack_set_drift
- [] estimate_template_cost
- [X] execute_change_set
- [] get_stack_policy
- [] get_template
- [] get_template_summary
- [] import_stacks_to_stack_set
- [X] list_change_sets
- [X] list_exports
- [] list_imports
- [] list_stack_instances
- [X] list_stack_resources
- [] list_stack_set_operation_results
- [] list_stack_set_operations
- [] list_stack_sets
- [X] list_stacks
- [] list_type_registrations

- [] list_type_versions
- [] list_types
- [] publish_type
- [] record_handler_progress
- [] register_publisher
- [] register_type
- [] rollback_stack
- [] set_stack_policy
- [] set_type_configuration
- [] set_type_default_version
- [] signal_resource
- [] stop_stack_set_operation
- [] test_type
- [X] update_stack
- [] update_stack_instances
- [X] update_stack_set
- [] update_termination_protection
- [X] validate_template

2.7.8 cloudtrail

class moto.cloudtrail.models.**CloudTrailBackend**(*region_name*)
Implementation of CloudTrail APIs.

Example usage

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

Implemented features for this service

- [] add_tags
- [X] create_trail
- [X] delete_trail
- [X] describe_trails
- [] get_event_selectors
- [] get_insight_selectors
- [X] get_trail
- [X] get_trail_status
- [] list_public_keys

- [] list_tags
- [X] list_trails
- [] lookup_events
- [] put_event_selectors
- [] put_insight_selectors
- [] remove_tags
- [X] start_logging
- [X] stop_logging
- [] update_trail

2.7.9 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_metric_streams`
- [X] `list_metrics`
- [X] `list_tags_for_resource`
- [] `put_anomaly_detector`
- [] `put_composite_alarm`
- [X] `put_dashboard`
- [] `put_insight_rule`
- [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`

2.7.10 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

2.7.11 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

2.7.12 cognito-identity

Example usage

```
@mock_cognitoidentity
def test_cognito-identity_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
- [] 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

2.7.13 cognito-idp

Example usage

```
@mock_cognitoidp
def test_cognito_idp_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
- [] 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
- [] 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
- [] update_group
- [X] update_identity_provider
- [] update_resource_server
- [] update_user_attributes
- [X] update_user_pool
- [X] update_user_pool_client
- [X] update_user_pool_domain
- [X] verify_software_token
- [] verify_user_attribute

2.7.14 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.
- [X] delete_configuration_aggregator
- [X] delete_configuration_recorder
- [] delete_conformance_pack
- [X] delete_delivery_channel
- [] delete_evaluation_results
- [] delete_organization_config_rule

- [X] 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
- [X] describe_aggregation_authorizations
- [] describe_compliance_by_config_rule
- [] describe_compliance_by_resource
- [] describe_config_rule_evaluation_status
- [X] describe_config_rules Return details for the given ConfigRule names or for all rules.
- [] describe_configuration_aggregator_sources_status
- [X] describe_configuration_aggregators
- [X] describe_configuration_recorder_status
- [X] describe_configuration_recorders
- [] describe_conformance_pack_compliance
- [] describe_conformance_pack_status
- [] describe_conformance_packs
- [] describe_delivery_channel_status
- [X] describe_delivery_channels
- [] describe_organization_config_rule_statuses
- [] describe_organization_config_rules
- [X] describe_organization_conformance_pack_statuses
- [X] 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_discovered_resource_counts`
- [] `get_organization_config_rule_detailed_status`
- [X] `get_organization_conformance_pack_detailed_status`
- [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_stored_query`
- [X] `list_aggregate_discovered_resources` Queries AWS Config listing function that must exist for resource backend.

As far as 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

- [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_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
- [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.

2.7.15 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

2.7.16 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_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_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

2.7.17 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_replication_instance
- [] create_replication_subnet_group
- [X] create_replication_task
- [] delete_certificate
- [] delete_connection
- [] delete_endpoint
- [] delete_event_subscription
- [] 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_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

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

2.7.18 ds

class moto.ds.models.**DirectoryServiceBackend**(*region_name=None*)
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_shared_directories
- [] describe_snapshots
- [] describe_trusts
- [] disable_client_authentication
- [] disable_ldaps
- [] disable_radius
- [X] disable_sso Disable single-sign on for a directory.
- [] enable_client_authentication
- [] enable_ldaps
- [] enable_radius
- [X] enable_sso Enable single-sign on for a directory.
- [X] get_directory_limits Return hard-coded limits for the directories.
- [] get_snapshot_limits
- [] list_certificates
- [] list_ip_routes

- [] list_log_subscriptions
- [] list_schema_extensions
- [X] list_tags_for_resource List all tags on a directory.
- [] register_certificate
- [] register_event_topic
- [] reject_shared_directory
- [] remove_ip_routes
- [] remove_region
- [X] 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_number_of_domain_controllers
- [] update_radius
- [] update_trust
- [] verify_trust

2.7.19 dynamodb

Example usage

```
@mock_dynamodb2
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_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
- [X] get_item
- [X] list_backups
- [] list_contributor_insights
- [] list_exports
- [] list_global_tables
- [X] list_tables
- [X] list_tags_of_resource
- [X] put_item
- [X] query
- [X] restore_table_from_backup
- [] restore_table_to_point_in_time
- [X] scan
- [X] tag_resource
- [X] transact_get_items
- [X] transact_write_items
- [X] untag_resource
- [X] update_continuous_backups
- [] update_contributor_insights
- [] update_global_table

- [] update_global_table_settings
- [X] update_item
- [X] update_table
- [] update_table_replica_auto_scaling
- [X] update_time_to_live

2.7.20 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

2.7.21 ec2

Example usage

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

Implemented features for this service

- [] 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
- [] allocate_hosts
- [] apply_security_groups_to_client_vpn_target_network
- [X] assign_ipv6_addresses
- [X] assign_private_ip_addresses

- [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
- [X] associate_route_table
- [X] associate_subnet_cidr_block
- [] associate_transit_gateway_multicast_domain
- [X] associate_transit_gateway_route_table
- [] associate_trunk_interface
- [X] associate_vpc_cidr_block
- [] attach_classic_link_vpc
- [X] attach_internet_gateway
- [X] attach_network_interface
- [X] attach_volume
- [X] attach_vpn_gateway
- [] authorize_client_vpn_ingress
- [X] authorize_security_group_egress
- [X] authorize_security_group_ingress
- [] bundle_instance
- [] cancel_bundle_task
- [] cancel_capacity_reservation
- [] cancel_capacity_reservation_fleets
- [] cancel_conversion_task
- [] cancel_export_task
- [] cancel_import_task
- [] cancel_reserved_instances_listing
- [X] cancel_spot_fleet_requests
- [X] cancel_spot_instance_requests
- [] confirm_product_instance
- [] copy_fpga_image
- [X] copy_image
- [X] copy_snapshot
- [] create_capacity_reservation
- [] create_capacity_reservation_fleet

- ☒ create_carrier_gateway
- ☐ create_client_vpn_endpoint
- ☐ create_client_vpn_route
- ☒ 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_key_pair
- ☒ create_launch_template
- ☐ create_launch_template_version
- ☐ create_local_gateway_route
- ☐ create_local_gateway_route_table_vpc_association
- ☒ create_managed_prefix_list
- ☒ create_nat_gateway
- ☒ create_network_acl
- ☒ create_network_acl_entry
- ☐ create_network_insights_path
- ☒ create_network_interface
- ☐ create_network_interface_permission
- ☐ create_placement_group
- ☐ 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

- [] create_spot_datafeed_subscription
- [] create_store_image_task
- [X] create_subnet
- [] create_subnet_cidr_reservation
- [X] create_tags
- [] create_traffic_mirror_filter
- [] create_traffic_mirror_filter_rule
- [] create_traffic_mirror_session
- [] create_traffic_mirror_target
- [X] create_transit_gateway
- [] create_transit_gateway_connect
- [] create_transit_gateway_connect_peer
- [] create_transit_gateway_multicast_domain
- [X] create_transit_gateway_peering_attachment
- [] create_transit_gateway_prefix_list_reference
- [X] create_transit_gateway_route
- [X] create_transit_gateway_route_table
- [X] create_transit_gateway_vpc_attachment
- [X] create_volume
- [X] create_vpc
- [X] create_vpc_endpoint
- [] create_vpc_endpoint_connection_notification
- [] create_vpc_endpoint_service_configuration
- [X] create_vpc_peering_connection
- [X] create_vpn_connection
- [] create_vpn_connection_route
- [X] create_vpn_gateway
- [X] delete_carrier_gateway
- [] delete_client_vpn_endpoint
- [] delete_client_vpn_route
- [X] delete_customer_gateway
- [] delete_dhcp_options
- [X] delete_egress_only_internet_gateway
- [] delete_fleets
- [X] delete_flow_logs
- [] delete_fpga_image

- ☐ delete_instance_event_window
- ☒ delete_internet_gateway
- ☒ delete_key_pair
- ☐ delete_launch_template
- ☐ delete_launch_template_versions
- ☐ delete_local_gateway_route
- ☐ delete_local_gateway_route_table_vpc_association
- ☒ delete_managed_prefix_list
- ☒ delete_nat_gateway
- ☒ delete_network_acl
- ☒ delete_network_acl_entry
- ☐ delete_network_insights_analysis
- ☐ delete_network_insights_path
- ☒ delete_network_interface
- ☐ delete_network_interface_permission
- ☐ delete_placement_group
- ☐ 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_prefix_list_reference
- ☒ delete_transit_gateway_route

- [X] delete_transit_gateway_route_table
- [X] delete_transit_gateway_vpc_attachment
- [X] delete_volume
- [X] delete_vpc
- [] delete_vpc_endpoint_connection_notifications
- [] delete_vpc_endpoint_service_configurations
- [X] delete_vpc_endpoints
- [X] delete_vpc_peering_connection
- [X] delete_vpn_connection
- [] delete_vpn_connection_route
- [X] delete_vpn_gateway
- [] deprovision_byoip_cidr
- [X] deregister_image
- [] deregister_instance_event_notification_attributes
- [] deregister_transit_gateway_multicast_group_members
- [] deregister_transit_gateway_multicast_group_sources
- [] describe_account_attributes
- [X] describe_addresses
- [] describe_addresses_attribute
- [] describe_aggregate_id_format
- [X] describe_availability_zones
- [] describe_bundle_tasks
- [] describe_byoip_cidrs
- [] describe_capacity_reservation_fleets
- [] describe_capacity_reservations
- [X] 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
- [X] describe_dhcp_options

- [X] describe_egress_only_internet_gateways
- [] describe_elastic_gpus
- [] describe_export_image_tasks
- [] describe_export_tasks
- [] describe_fast_snapshot_restores
- [] describe_fleet_history
- [] describe_fleet_instances
- [] describe_fleets
- [X] describe_flow_logs
- [] describe_fpga_image_attribute
- [] describe_fpga_images
- [] describe_host_reservation_offerings
- [] describe_host_reservations
- [] describe_hosts
- [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_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_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
- [] describe_security_group_rules
- [X] describe_security_groups
- [] describe_snapshot_attribute
- [X] describe_snapshots
- [] describe_spot_datafeed_subscription
- [X] describe_spot_fleet_instances
- [] describe_spot_fleet_request_history
- [X] describe_spot_fleet_requests
- [X] describe_spot_instance_requests

- [X] `describe_spot_price_history`
- [] `describe_stale_security_groups`
- [] `describe_store_image_tasks`
- [] `describe_subnets`
- [X] `describe_tags`
- [] `describe_traffic_mirror_filters`
- [] `describe_traffic_mirror_sessions`
- [] `describe_traffic_mirror_targets`
- [X] `describe_transit_gateway_attachments`
- [] `describe_transit_gateway_connect_peers`
- [] `describe_transit_gateway_connects`
- [] `describe_transit_gateway_multicast_domains`
- [X] `describe_transit_gateway_peering_attachments`
- [] `describe_transit_gateway_route_tables`
- [X] `describe_transit_gateway_vpc_attachments`
- [X] `describe_transit_gateways`
- [] `describe_trunk_interface_associations`
- [] `describe_volume_attribute`
- [] `describe_volume_status`
- [X] `describe_volumes`
- [] `describe_volumes_modifications`
- [X] `describe_vpc_attribute`
- [] `describe_vpc_classic_link`
- [] `describe_vpc_classic_link_dns_support`
- [] `describe_vpc_endpoint_connection_notifications`
- [] `describe_vpc_endpoint_connections`
- [] `describe_vpc_endpoint_service_configurations`
- [] `describe_vpc_endpoint_service_permissions`
- [X] `describe_vpc_endpoint_services` Return info on services to which you can create a VPC endpoint.

Currently only the default endpointing 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`

- ☒ describe_vpn_connections
- ☒ describe_vpn_gateways
- ☐ detach_classic_link_vpc
- ☒ detach_internet_gateway
- ☒ detach_network_interface
- ☒ detach_volume
- ☒ detach_vpn_gateway
- ☐ disable_ebs_encryption_by_default
- ☐ disable_fast_snapshot_restores
- ☐ disable_image_deprecation
- ☐ disable_serial_console_access
- ☒ disable_transit_gateway_route_table_propagation
- ☐ disable_vgw_route_propagation
- ☒ disable_vpc_classic_link
- ☒ disable_vpc_classic_link_dns_support
- ☒ disassociate_address
- ☐ disassociate_client_vpn_target_network
- ☐ disassociate_enclave_certificate_iam_role
- ☒ disassociate_iam_instance_profile
- ☐ disassociate_instance_event_window
- ☒ disassociate_route_table
- ☒ disassociate_subnet_cidr_block
- ☐ disassociate_transit_gateway_multicast_domain
- ☒ disassociate_transit_gateway_route_table
- ☐ disassociate_trunk_interface
- ☒ disassociate_vpc_cidr_block
- ☐ enable_ebs_encryption_by_default
- ☐ enable_fast_snapshot_restores
- ☐ enable_image_deprecation
- ☐ 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_capacity_reservation_usage
- [] get_coip_pool_usage
- [] get_console_output
- [] get_console_screenshot
- [] get_default_credit_specification
- [] get_ebs_default_kms_key_id
- [] 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_launch_template_data
- [] get_managed_prefix_list_associations
- [X] get_managed_prefix_list_entries
- [] 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_prefix_list_references
- [] get_transit_gateway_route_table_associations
- [] get_transit_gateway_route_table_propagations
- [] get_vpn_connection_device_sample_configuration
- [] get_vpn_connection_device_types
- [] import_client_vpn_client_certificate_revocation_list
- [] import_image
- [] import_instance
- [X] import_key_pair
- [] import_snapshot

- [] import_volume
- [] 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
- [X] modify_instance_attribute
- [] modify_instance_capacity_reservation_attributes
- [] modify_instance_credit_specification
- [] modify_instance_event_start_time
- [] modify_instance_event_window
- [] modify_instance_metadata_options
- [] modify_instance_placement
- [] modify_launch_template
- [X] modify_managed_prefix_list
- [X] modify_network_interface_attribute
- [] modify_reserved_instances
- [] modify_security_group_rules
- [] modify_snapshot_attribute
- [X] modify_spot_fleet_request
- [X] modify_subnet_attribute
- [] modify_traffic_mirror_filter_network_services
- [] modify_traffic_mirror_filter_rule
- [] modify_traffic_mirror_session
- [X] modify_transit_gateway
- [] modify_transit_gateway_prefix_list_reference
- [X] modify_transit_gateway_vpc_attachment
- [] modify_volume

- ☐ modify_volume_attribute
- ☒ modify_vpc_attribute
- ☐ modify_vpc_endpoint
- ☐ modify_vpc_endpoint_connection_notification
- ☐ modify_vpc_endpoint_service_configuration
- ☐ 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
- ☐ provision_byoip_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
- ☒ 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
- [X] request_spot_fleet
- [X] 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_managed_prefix_list_version
- [] revoke_client_vpn_ingress
- [X] revoke_security_group_egress
- [X] revoke_security_group_ingress
- [X] run_instances
- [] run_scheduled_instances
- [] search_local_gateway_routes
- [] search_transit_gateway_multicast_groups
- [X] search_transit_gateway_routes
- [] send_diagnostic_interrupt
- [X] start_instances
- [] 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
- [] unmonitor_instances
- [X] update_security_group_rule_descriptions_egress
- [X] update_security_group_rule_descriptions_ingress
- [] withdraw_byoip_cidr

2.7.22 ec2-instance-connect

Example usage

```
@mock_ec2instanceconnect
def test_ec2-instance-connect_behaviour:
    boto3.client("ec2-instance-connect")
    ...
```

Implemented features for this service

- [] send_serial_console_ssh_public_key
- [X] send_ssh_public_key

2.7.23 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
- [] complete_layer_upload
- [X] create_repository
- [X] delete_lifecycle_policy
- [X] delete_registry_policy
- [X] delete_repository
- [X] delete_repository_policy
- [] describe_image_replication_status
- [X] describe_image_scan_findings
- [X] describe_images
- [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
- [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
- [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

2.7.24 ecs

Example usage

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

Implemented features for this service

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

- [X] deregister_container_instance
- [X] deregister_task_definition
- [] describe_capacity_providers
- [X] describe_clusters
- [X] describe_container_instances
- [X] describe_services
- [X] describe_task_definition
- [X] describe_task_sets
- [X] describe_tasks
- [] discover_poll_endpoint
- [] execute_command
- [X] list_account_settings
- [X] list_attributes
- [X] list_clusters
 - maxSize and pagination not implemented
- [X] list_container_instances
- [X] list_services
- [X] list_tags_for_resource Currently implemented only for task definitions and services
- [X] list_task_definition_families
- [X] list_task_definitions
- [X] list_tasks
- [X] put_account_setting
- [] put_account_setting_default
- [X] put_attributes
- [] 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 Currently implemented only for services
- [X] untag_resource Currently implemented only for services
- [] update_capacity_provider

- [] update_cluster
- [] 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
- [X] update_task_set

2.7.25 efs

class moto.efs.models.EFSBackend(region_name=None)

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

- [] 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_tags
- [] 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_tags
- [] describe_access_points

- [] 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

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

Note that as of this writing access points, and thus access point IDs are not supported.

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

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

2.7.26 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_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
- [] list_tags_for_resource
- [] list_updates
- [] register_cluster
- [] tag_resource
- [] untag_resource
- [] update_addon
- [] update_cluster_config
- [] update_cluster_version
- [] update_nodegroup_config
- [] update_nodegroup_version

2.7.27 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

2.7.28 elastictranscoder

Example usage

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

Implemented features for this service

- [] cancel_job
- [] create_job
- [X] create_pipeline
- [] 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
- [] update_pipeline_notifications
- [] update_pipeline_status

2.7.29 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
- [] 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
- [] delete_load_balancer_policy
- [] deregister_instances_from_load_balancer
- [] describe_account_limits
- [] describe_instance_health
- [] describe_load_balancer_attributes
- [] describe_load_balancer_policies
- [] describe_load_balancer_policy_types
- [X] describe_load_balancers
- [] describe_tags
- [] detach_load_balancer_from_subnets
- [] disable_availability_zones_for_load_balancer
- [] enable_availability_zones_for_load_balancer
- [] 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

2.7.30 elbv2

Example usage

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

Implemented features for this service

- [] add_listener_certificates
- [] 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
- [] describe_listener_certificates
- [X] describe_listeners
- [X] describe_load_balancer_attributes
- [X] describe_load_balancers
- [X] describe_rules
- [] describe_ssl_policies
- [] 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
- [] modify_target_group_attributes
- [X] register_targets
- [] remove_listener_certificates

- [] remove_tags
- [X] set_ip_address_type
- [X] set_rule_priorities
- [X] set_security_groups
- [X] set_subnets

2.7.31 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
- [X] describe_job_flows
- [] describe_notebook_execution
- [] describe_release_label
- [] describe_security_configuration
- [X] describe_step
- [] describe_studio
- [] get_auto_termination_policy
- [] get_block_public_access_configuration
- [] get_managed_scaling_policy
- [] get_studio_session_mapping
- [X] list_bootstrap_actions

- [X] list_clusters
- [] list_instance_fleets
- [X] list_instance_groups
- [X] list_instances
- [] list_notebook_executions
- [] list_release_labels
- [] list_security_configurations
- [X] list_steps
- [] list_studio_session_mappings
- [] list_studios
- [X] modify_cluster
- [] modify_instance_fleet
- [X] modify_instance_groups
- [X] put_auto_scaling_policy
- [] put_auto_termination_policy
- [] put_block_public_access_configuration
- [] put_managed_scaling_policy
- [X] remove_auto_scaling_policy
- [] remove_auto_termination_policy
- [] remove_managed_scaling_policy
- [X] remove_tags
- [X] run_job_flow
- [X] set_termination_protection
- [X] set_visible_to_all_users
- [] start_notebook_execution
- [] stop_notebook_execution
- [X] terminate_job_flows
- [] update_studio
- [] update_studio_session_mapping

2.7.32 emr-containers

class moto.emrcontainers.models.**EMRContainersBackend**(*region_name=None*)
Implementation of EMRContainers APIs.

Example usage

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

Implemented features for this service

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

2.7.33 events

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
- [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

- [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

- [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
- [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
- [] 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

2.7.34 firehose

class moto.firehose.models.**FirehoseBackend**(*region_name=None*)

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.

2.7.35 forecast

Example usage

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

Implemented features for this service

- [] create_dataset
- [X] create_dataset_group
- [] create_dataset_import_job
- [] create_forecast
- [] create_forecast_export_job
- [] create_predictor
- [] create_predictor_backtest_export_job
- [] delete_dataset
- [X] delete_dataset_group
- [] delete_dataset_import_job
- [] delete_forecast
- [] delete_forecast_export_job
- [] delete_predictor
- [] delete_predictor_backtest_export_job
- [] delete_resource_tree
- [] describe_dataset
- [X] describe_dataset_group
- [] describe_dataset_import_job
- [] describe_forecast
- [] describe_forecast_export_job
- [] describe_predictor
- [] describe_predictor_backtest_export_job
- [] get_accuracy_metrics
- [X] list_dataset_groups
- [] list_dataset_import_jobs
- [] list_datasets
- [] list_forecast_export_jobs
- [] list_forecasts
- [] list_predictor_backtest_export_jobs

- [] list_predictors
- [] list_tags_for_resource
- [] stop_resource
- [] tag_resource
- [] untag_resource
- [X] update_dataset_group

2.7.36 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

2.7.37 glue

Example usage

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

Implemented features for this service

- [] batch_create_partition
- [] batch_delete_connection
- [] batch_delete_partition
- [] batch_delete_table
- [] batch_delete_table_version
- [] batch_get_blueprints
- [] batch_get_crawlers
- [] batch_get_dev_endpoints
- [] batch_get_jobs
- [] batch_get_partition
- [] batch_get_triggers
- [] batch_get_workflows
- [] batch_stop_job_run
- [] batch_update_partition
- [] cancel_ml_task_run
- [] check_schema_version_validity
- [] create_blueprint

- [] create_classifier
- [] create_connection
- [X] create_crawler
- [X] create_database
- [] create_dev_endpoint
- [X] create_job
- [] create_ml_transform
- [] create_partition
- [] create_partition_index
- [] create_registry
- [] create_schema
- [] create_script
- [] create_security_configuration
- [X] 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
- [X] delete_crawler
- [] 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
- [X] 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
- [X] get_crawler
- [] get_crawler_metrics
- [X] get_crawlers
- [] get_data_catalog_encryption_settings
- [X] get_database
- [X] 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
- [] get_partitions
- [] get_plan

- [] `get_registry`
- [] `get_resource_policies`
- [] `get_resource_policy`
- [] `get_schema`
- [] `get_schema_by_definition`
- [] `get_schema_version`
- [] `get_schema_versions_diff`
- [] `get_security_configuration`
- [] `get_security_configurations`
- [X] `get_table`
- [] `get_table_version`
- [] `get_table_versions`
- [X] `get_tables`
- [] `get_tags`
- [] `get_trigger`
- [] `get_triggers`
- [] `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`
- [] `list_crawlers`
- [] `list_dev_endpoints`
- [] `list_jobs`
- [] `list_ml_transforms`
- [] `list_registries`
- [] `list_schema_versions`
- [] `list_schemas`
- [] `list_triggers`
- [] `list_workflows`
- [] `put_data_catalog_encryption_settings`
- [] `put_resource_policy`
- [] `put_schema_version_metadata`

- [] put_workflow_run_properties
- [] query_schema_version_metadata
- [] register_schema_version
- [] remove_schema_version_metadata
- [] reset_job_bookmark
- [] resume_workflow_run
- [] search_tables
- [] start_blueprint_run
- [X] start_crawler
- [] start_crawler_schedule
- [] start_export_labels_task_run
- [] start_import_labels_task_run
- [] 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_trigger
- [] stop_workflow_run
- [] tag_resource
- [] untag_resource
- [] update_blueprint
- [] update_classifier
- [] update_column_statistics_for_partition
- [] update_column_statistics_for_table
- [] update_connection
- [] update_crawler
- [] update_crawler_schedule
- [] update_database
- [] update_dev_endpoint
- [] update_job
- [] update_ml_transform
- [] update_partition
- [] update_registry

- [] update_schema
- [] update_table
- [] update_trigger
- [] update_user_defined_function
- [] update_workflow

2.7.38 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
- [] 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
- [] 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`
- [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`
- [] `get_service_linked_role_deletion_status`
- [X] `get_ssh_public_key`
- [X] `get_user`
- [X] `get_user_policy`
- [] `list_access_keys`
- [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`
- [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`
- [] `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`
- ☐ `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
- [X] tag_user
- [] untag_instance_profile
- [] untag_mfa_device
- [] untag_open_id_connect_provider
- [X] untag_policy
- [X] untag_role
- [] untag_saml_provider
- [] untag_server_certificate
- [X] untag_user
- [X] update_access_key
- [X] update_account_password_policy
- [] update_assume_role_policy
- [] update_group
- [X] update_login_profile
- [] update_open_id_connect_provider_thumbprint
- [X] update_role
- [X] update_role_description
- [X] update_saml_provider
- [] update_server_certificate
- [] update_service_specific_credential
- [X] update_signing_certificate
- [X] update_ssh_public_key
- [X] update_user
- [X] upload_server_certificate
- [X] upload_signing_certificate
- [X] upload_ssh_public_key

2.7.39 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
- [] clear_default_authorizer
- [] confirm_topic_rule_destination
- [] create_audit_suppression
- [] create_authorizer
- [] create_billing_group
- [] create_certificate_from_csr
- [] create_custom_metric
- [] create_dimension
- [] create_domain_configuration
- [] create_dynamic_thing_group
- [] create_fleet_metric
- [X] create_job
- [] create_job_template
- [X] create_keys_and_certificate
- [] create_mitigation_action
- [] create_ota_update
- [X] create_policy
- [X] create_policy_version
- [] create_provisioning_claim
- [] create_provisioning_template
- [] create_provisioning_template_version
- [] create_role_alias
- [] create_scheduled_audit

- [] create_security_profile
- [] create_stream
- [X] create_thing
- [X] create_thing_group
- [X] create_thing_type
- [X] create_topic_rule
- [] create_topic_rule_destination
- [] delete_account_audit_configuration
- [] delete_audit_suppression
- [] delete_authorizer
- [] delete_billing_group
- [] delete_ca_certificate
- [X] delete_certificate
- [] delete_custom_metric
- [] delete_dimension
- [] delete_domain_configuration
- [] delete_dynamic_thing_group
- [] delete_fleet_metric
- [X] delete_job
- [X] delete_job_execution
- [] delete_job_template
- [] delete_mitigation_action
- [] delete_ota_update
- [X] delete_policy
- [X] 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
- [X] delete_thing_group
- [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
- [] describe_ca_certificate
- [X] describe_certificate
- [] describe_custom_metric
- [] describe_default_authorizer
- [] describe_detect_mitigation_actions_task
- [] describe_dimension
- [] 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_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
- [] 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
- [] list_certificates_by_ca
- [] list_custom_metrics
- [] list_detect_mitigation_actions_executions
- [] list_detect_mitigation_actions_tasks

- [] list_dimensions
- [] 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
- [] 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_role_aliases
- [] list_scheduled_audits
- [] list_security_profiles
- [] list_security_profiles_for_target
- [] list_streams
- [] list_tags_for_resource
- [] list_targets_for_policy
- [] list_targets_for_security_profile
- [X] list_thing_groups
- [X] list_thing_groups_for_thing
- [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
- [] list_topic_rule_destinations

- [X] list_topic_rules
- [] list_v2_logging_levels
- [] list_violation_events
- [] put_verification_state_on_violation
- [] register_ca_certificate
- [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
- [] search_index
- [] 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
- [] update_ca_certificate
- [X] update_certificate
- [] update_custom_metric
- [] update_dimension

- [] 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
- [X] update_thing_group
- [X] update_thing_groups_for_thing
- [] update_topic_rule_destination
- [] validate_security_profile_behaviors

2.7.40 iot-data

Example usage

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

Implemented features for this service

- [X] delete_thing_shadow after deleting, get_thing_shadow will raise ResourceNotFound.

But version of the shadow keep increasing...

- [] 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’

2.7.41 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
- [] deregister_stream_consumer
- [] describe_limits
- [X] describe_stream
- [] describe_stream_consumer
- [X] describe_stream_summary
- [] disable_enhanced_monitoring
- [] enable_enhanced_monitoring
- [X] get_records
- [X] get_shard_iterator
- [X] increase_stream_retention_period
- [X] list_shards
- [] list_stream_consumers
- [X] list_streams
- [X] list_tags_for_stream
- [X] merge_shards
- [X] put_record
- [X] put_records
- [] register_stream_consumer
- [X] remove_tags_from_stream
- [X] split_shard
- [] start_stream_encryption
- [] stop_stream_encryption
- [] subscribe_to_shard

- [] update_shard_count

2.7.42 kinesis-video-archived-media

Example usage

```
@mock_kinesisvideoarchivedmedia
def test_kinesis-video-archived-media_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_media_for_fragment_list
- [] list_fragments

2.7.43 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
- [] describe_signaling_channel
- [X] describe_stream
- [X] get_data_endpoint
- [] get_signaling_channel_endpoint
- [] list_signaling_channels
- [X] list_streams
- [] list_tags_for_resource
- [] list_tags_for_stream
- [] tag_resource
- [] tag_stream
- [] untag_resource

- [] untag_stream
- [] update_data_retention
- [] update_signaling_channel
- [] update_stream

2.7.44 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
- [] create_grant
- [X] create_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_random
- [X] get_key_policy

- [X] get_key_rotation_status
- [] get_parameters_for_import
- [] get_public_key
- [] import_key_material
- [] list_aliases
- [] list_grants
- [] list_key_policies
- [X] list_keys
- [X] list_resource_tags
- [] list_retirable_grants
- [X] put_key_policy
- [X] re_encrypt
- [] replicate_key
- [] retire_grant
- [] revoke_grant
- [X] schedule_key_deletion
- [] sign
- [X] tag_resource
- [X] untag_resource
- [] update_alias
- [] update_custom_key_store
- [X] update_key_description
- [] update_primary_region
- [] verify

2.7.45 lambda

Example usage

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

Implemented features for this service

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

- [X] `create_event_source_mapping`
- [X] `create_function`
- [] `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`
- [] `delete_layer_version`
- [] `delete_provisioned_concurrency_config`
- [] `get_account_settings`
- [] `get_alias`
- [] `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`
- [] `get_layer_version`
- [] `get_layer_version_by_arn`
- [] `get_layer_version_policy`
- [X] `get_policy`
- [] `get_provisioned_concurrency_config`
- [X] `invoke`
- [] `invoke_async`
- [] `list_aliases`
- [] `list_code_signing_configs`
- [X] `list_event_source_mappings`
- [] `list_function_event_invoke_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
- [] remove_layer_version_permission
- [X] remove_permission
- [X] tag_resource
- [X] untag_resource
- [] update_alias
- [] update_code_signing_config
- [X] update_event_source_mapping
- [X] update_function_code
- [X] update_function_configuration
- [] update_function_event_invoke_config

2.7.46 logs

Example usage

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

Implemented features for this service

- [] associate_kms_key
- [] cancel_export_task
- [] create_export_task
- [X] create_log_group
- [X] create_log_stream
- [] 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`
- [X] `get_log_events`
- [] `get_log_group_fields`
- [] `get_log_record`
- [] `get_query_results`
- [X] `list_tags_log_group`
- [] `put_destination`
- [] `put_destination_policy`
- [X] `put_log_events`
- [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`
- [] `test_metric_filter`
- [X] `untag_log_group`

2.7.47 managedblockchain

Example usage

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

Implemented features for this service

- [X] create_member
- [X] create_network
- [X] create_node
- [X] create_proposal
- [X] delete_member
- [X] delete_node
- [X] get_member
- [X] get_network
- [X] get_node
- [X] get_proposal
- [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

2.7.48 mediaconnect

Example usage

```
@mock_mediaconnect
def test_mediaconnect_behaviour:
    boto3.client("mediaconnect")
    ...
```

Implemented features for this service

- [] add_flow_media_streams
- [X] add_flow_outputs
- [] add_flow_sources
- [X] add_flow_vpc_interfaces
- [X] create_flow
- [X] delete_flow
- [X] describe_flow
- [] describe_offering
- [] describe_reservation
- [] 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
- [] revoke_flow_entitlement
- [X] start_flow
- [X] stop_flow
- [X] tag_resource
- [] untag_resource
- [] update_flow
- [] update_flow_entitlement
- [] update_flow_media_stream
- [] update_flow_output

- [] update_flow_source

2.7.49 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
- [X] create_input
- [] 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
- [] reject_input_device_transfer
- [X] start_channel
- [] 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

2.7.50 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

2.7.51 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
- [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

2.7.52 mediastore-data

Example usage

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

Implemented features for this service

- [X] delete_object
- [] describe_object
- [X] get_object
- [X] list_items
- [X] put_object

2.7.53 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
- [X] describe_apps
- [] describe_commands
- [] describe_deployments
- [] describe_ecs_clusters
- [] describe_elastic_ips
- [] describe_elastic_load_balancers
- [X] describe_instances

- [X] 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
- [X] 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

2.7.54 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] create_account
- [] create_gov_cloud_account
- [X] create_organization
- [X] create_organizational_unit
- [X] create_policy
- [] decline_handshake
- [X] delete_organization
- [] delete_organizational_unit
- [X] delete_policy
- [X] deregister_delegated_administrator
- [X] describe_account
- [X] describe_create_account_status
- [] describe_effective_policy
- [] describe_handshake
- [X] describe_organization
- [X] describe_organizational_unit

- [X] describe_policy
- [X] detach_policy
- [X] disable_aws_service_access
- [X] disable_policy_type
- [] enable_all_features
- [X] enable_aws_service_access
- [X] enable_policy_type
- [] invite_account_to_organization
- [] leave_organization
- [X] list_accounts
- [X] list_accounts_for_parent
- [X] list_aws_service_access_for_organization
- [X] list_children
- [X] list_create_account_status
- [X] list_delegated_administrators
- [X] list_delegated_services_for_account
- [] list_handshakes_for_account
- [] list_handshakes_for_organization
- [X] list_organizational_units_for_parent
- [X] list_parents
- [X] list_policies
- [X] list_policies_for_target
- [X] list_roots
- [X] list_tags_for_resource
- [X] list_targets_for_policy
- [X] move_account
- [X] register_delegated_administrator
- [] remove_account_from_organization
- [X] tag_resource
- [X] untag_resource
- [X] update_organizational_unit
- [X] update_policy

2.7.55 polly

Example usage

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

Implemented features for this service

- [X] delete_lexicon
- [X] describe_voices
- [X] get_lexicon
- [] get_speech_synthesis_task
- [X] list_lexicons
- [] list_speech_synthesis_tasks
- [X] put_lexicon
- [] start_speech_synthesis_task
- [] synthesize_speech

2.7.56 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
- [X] create_resource_share
- [X] delete_resource_share
- [] disassociate_resource_share
- [] disassociate_resource_share_permission
- [X] enable_sharing_with_aws_organization
- [] get_permission
- [] get_resource_policies
- [] get_resource_share_associations
- [] get_resource_share_invitations

- [X] get_resource_shares
- [] list_pending_invitation_resources
- [] 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

2.7.57 rds

Example usage

```
@mock_rds2
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
- [] cancel_export_task
- [] copy_db_cluster_parameter_group
- [] copy_db_cluster_snapshot
- [] copy_db_parameter_group
- [] copy_db_snapshot
- [] copy_option_group
- [] create_custom_availability_zone
- [] 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
- [X] 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
- [X] create_option_group
- [] delete_custom_availability_zone
- [] delete_custom_db_engine_version
- [X] delete_db_cluster
- [] delete_db_cluster_endpoint
- [] delete_db_cluster_parameter_group
- [] delete_db_cluster_snapshot
- [] delete_db_instance
- [] delete_db_instance_automated_backup
- [X] 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_installation_media
- [X] delete_option_group
- [] deregister_db_proxy_targets
- [] describe_account_attributes
- [] describe_certificates
- [] describe_custom_availability_zones

- [] 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
- [X] describe_db_clusters
- [] describe_db_engine_versions
- [] describe_db_instance_automated_backups
- [] describe_db_instances
- [] describe_db_log_files
- [X] 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_installation_media
- [X] describe_option_group_options
- [X] 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
- [] import_installation_media
- [X] list_tags_for_resource
- [] 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
- [X] modify_db_parameter_group
- [] modify_db_proxy
- [] modify_db_proxy_endpoint
- [] modify_db_proxy_target_group
- [] modify_db_snapshot
- [] modify_db_snapshot_attribute
- [X] modify_db_subnet_group
- [] modify_event_subscription
- [] modify_global_cluster
- [X] modify_option_group
- [] promote_read_replica
- [] promote_read_replica_db_cluster
- [] purchase_reserved_db_instances_offering
- [X] 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
- [X] 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
- [] start_db_instance
- [] start_db_instance_automated_backups_replication
- [] start_export_task
- [] stop_activity_stream
- [X] stop_db_cluster
- [] stop_db_instance
- [] stop_db_instance_automated_backups_replication

2.7.58 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

- [X] describe_cluster_security_groups
- [X] describe_cluster_snapshots
- [X] describe_cluster_subnet_groups
- [] describe_cluster_tracks
- [] describe_cluster_versions
- [X] 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_offerings
- [] describe_reserved_nodes
- [] describe_resize
- [] describe_scheduled_actions
- [X] describe_snapshot_copy_grants
- [] describe_snapshot_schedules
- [] describe_storage
- [] describe_table_restore_status
- [X] describe_tags
- [] describe_usage_limits
- [] disable_logging
- [X] disable_snapshot_copy
- [] disassociate_data_share_consumer
- [] enable_logging
- [X] enable_snapshot_copy

- ☒ `get_cluster_credentials`
- ☐ `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`

2.7.59 resource-groups

Example usage

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

Implemented features for this service

- [X] create_group
- [X] delete_group
- [X] get_group
- [X] get_group_configuration
- [] get_group_query
- [X] get_tags
- [] group_resources
- [] list_group_resources
- [X] list_groups
- [X] put_group_configuration
- [] search_resources
- [X] tag
- [] ungroup_resources
- [X] untag
- [X] update_group
- [X] update_group_query

2.7.60 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

2.7.61 route53

Example usage

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

Implemented features for this service

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

- [] get_account_limit
- [] get_change
- [] get_checker_ip_ranges
- [] get_dnssec
- [] get_geo_location
- [] get_health_check
- [] get_health_check_count
- [] get_health_check_last_failure_reason
- [] get_health_check_status
- [X] get_hosted_zone
- [] get_hosted_zone_count
- [] get_hosted_zone_limit
- [X] get_query_logging_config Return query logging config, if it exists.
- [] get_reusable_delegation_set
- [] get_reusable_delegation_set_limit
- [] get_traffic_policy
- [] get_traffic_policy_instance
- [] get_traffic_policy_instance_count
- [] list_geo_locations
- [X] list_health_checks
- [X] list_hosted_zones
- [X] list_hosted_zones_by_name
- [] list_hosted_zones_by_vpc
- [X] list_query_logging_configs Return a list of query logging configs.
- [] list_resource_record_sets
- [] list_reusable_delegation_sets
- [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
- [] update_health_check

- [] update_hosted_zone_comment
- [] update_traffic_policy_comment
- [] update_traffic_policy_instance

2.7.62 route53resolver

class moto.route53resolver.models.**Route53ResolverBackend**(*region_name=None*)
Implementation of Route53Resolver APIs.

Example usage

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

Implemented features for this service

- [] associate_firewall_rule_group
- [] associate_resolver_endpoint_ip_address
- [] associate_resolver_query_log_config
- [] 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
- [] create_resolver_rule
- [] delete_firewall_domain_list
- [] delete_firewall_rule
- [] delete_firewall_rule_group
- [X] delete_resolver_endpoint Delete a resolver endpoint.
- [] delete_resolver_query_log_config
- [] delete_resolver_rule
- [] disassociate_firewall_rule_group
- [] disassociate_resolver_endpoint_ip_address
- [] disassociate_resolver_query_log_config
- [] 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` Return info for specified resolver endpoint.
- [] `get_resolver_query_log_config`
- [] `get_resolver_query_log_config_association`
- [] `get_resolver_query_log_config_policy`
- [] `get_resolver_rule`
- [] `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` List IP endresses for specified resolver endpoint.
- [X] `list_resolver_endpoints` List all resolver endpoints, using filters if specified.
- [] `list_resolver_query_log_config_associations`
- [] `list_resolver_query_log_configs`
- [] `list_resolver_rule_associations`
- [] `list_resolver_rules`
- [X] `list_tags_for_resource` List all tags for the given resource.
- [] `put_firewall_rule_group_policy`
- [] `put_resolver_query_log_config_policy`
- [] `put_resolver_rule_policy`
- [X] `tag_resource` Add or overwrite one or more tags for specified resource.
- [X] `untag_resource` Removes tags from a 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 name of Resolver endpoint.
- [] update_resolver_rule

2.7.63 s3

class moto.s3.models.S3Backend

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
- [] 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
- [] 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
- [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`
- [] `put_bucket_ownership_controls`
- [X] `put_bucket_policy`
- [X] `put_bucket_replication`
- [] `put_bucket_request_payment`
- [X] `put_bucket_tagging`
- [] `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

2.7.64 sagemaker

Example usage

```
@mock_sagemaker
def test_sagemaker_behaviour:
    boto3.client("sagemaker")
    ...
```

Implemented features for this service

- [] add_association
- [] 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_packaging_job
- [X] create_endpoint
- [X] create_endpoint_config

- [X] create_experiment
- [] create_feature_group
- [] create_flow_definition
- [] create_human_task_ui
- [] create_hyper_parameter_tuning_job
- [] create_image
- [] create_image_version
- [] create_labeling_job
- [X] create_model
- [] create_model_bias_job_definition
- [] create_model_explainability_job_definition
- [] create_model_package
- [] create_model_package_group
- [] create_model_quality_job_definition
- [] create_monitoring_schedule
- [X] create_notebook_instance
- [X] create_notebook_instance_lifecycle_config
- [] create_pipeline
- [] create_presigned_domain_url
- [] create_presigned_notebook_instance_url
- [X] create_processing_job
- [] create_project
- [] create_studio_lifecycle_config
- [X] create_training_job
- [] create_transform_job
- [X] create_trial
- [X] 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
- [X] delete_endpoint
- [X] delete_endpoint_config
- [X] delete_experiment
- [] delete_feature_group
- [] delete_flow_definition
- [] delete_human_task_ui
- [] delete_image
- [] delete_image_version
- [X] delete_model
- [] delete_model_bias_job_definition
- [] 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
- [X] delete_notebook_instance
- [X] delete_notebook_instance_lifecycle_config
- [] delete_pipeline
- [] delete_project
- [] delete_studio_lifecycle_config
- [] delete_tags
- [X] delete_trial
- [X] 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_packaging_job
- [X] describe_endpoint
- [X] describe_endpoint_config
- [X] describe_experiment
- [] describe_feature_group
- [] describe_flow_definition
- [] describe_human_task_ui
- [] describe_hyper_parameter_tuning_job
- [] describe_image
- [] describe_image_version
- [] describe_labeling_job
- [X] describe_model
- [] describe_model_bias_job_definition
- [] describe_model_explainability_job_definition
- [] describe_model_package
- [] describe_model_package_group
- [] describe_model_quality_job_definition
- [] describe_monitoring_schedule
- [] describe_notebook_instance
- [X] describe_notebook_instance_lifecycle_config
- [] describe_pipeline
- [] describe_pipeline_definition_for_execution
- [] describe_pipeline_execution
- [X] describe_processing_job
- [] describe_project
- [] describe_studio_lifecycle_config

- [] describe_subscribed_workteam
- [X] describe_training_job
- [] describe_transform_job
- [X] describe_trial
- [X] describe_trial_component
- [] describe_user_profile
- [] describe_workforce
- [] describe_workteam
- [] disable_sagemaker_servicecatalog_portfolio
- [X] disassociate_trial_component
- [] enable_sagemaker_servicecatalog_portfolio
- [] get_device_fleet_report
- [] get_model_package_group_policy
- [] get_sagemaker_servicecatalog_portfolio_status
- [] get_search_suggestions
- [] list_actions
- [] list_algorithms
- [] 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_packaging_jobs
- [] list_endpoint_configs
- [] list_endpoints
- [X] list_experiments
- [] list_feature_groups
- [] list_flow_definitions

- [] list_human_task_uis
- [] list_hyper_parameter_tuning_jobs
- [] list_image_versions
- [] list_images
- [] list_labeling_jobs
- [] list_labeling_jobs_for_workteam
- [] list_model_bias_job_definitions
- [] list_model_explainability_job_definitions
- [] list_model_package_groups
- [] list_model_packages
- [] list_model_quality_job_definitions
- [X] list_models
- [] list_monitoring_executions
- [] list_monitoring_schedules
- [] list_notebook_instance_lifecycle_configs
- [] list_notebook_instances
- [] list_pipeline_execution_steps
- [] list_pipeline_executions
- [] list_pipeline_parameters_for_execution
- [] list_pipelines
- [X] list_processing_jobs
- [] list_projects
- [] list_studio_lifecycle_configs
- [] list_subscribed_workteams
- [] 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
- [] register_devices
- [] render_ui_template

- [] retry_pipeline_execution
- [X] search
- [] send_pipeline_execution_step_failure
- [] send_pipeline_execution_step_success
- [] start_monitoring_schedule
- [X] start_notebook_instance
- [] start_pipeline_execution
- [] stop_auto_ml_job
- [] stop_compilation_job
- [] stop_edge_packaging_job
- [] stop_hyper_parameter_tuning_job
- [] stop_labeling_job
- [] stop_monitoring_schedule
- [X] 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_image
- [] update_model_package
- [] update_monitoring_schedule
- [] update_notebook_instance
- [] update_notebook_instance_lifecycle_config
- [] update_pipeline
- [] update_pipeline_execution

- [] update_project
- [] update_training_job
- [] update_trial
- [] update_trial_component
- [] update_user_profile
- [] update_workforce
- [] update_workteam

2.7.65 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

2.7.66 secretsmanager

Example usage

```
@mock_secretsmanager
def test_secretsmanager_behaviour:
    boto3.client("secretsmanager")
    ...
```

Implemented features for this service

- [] 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

Returns secrets from secretsmanager. The result is paginated and page items depends on the token value, because token contains start element number of secret list. Response example: {

SecretList: [

```
{ ARN: 'arn:aws:secretsmanager:us-east-1:1234567890:secret:test1-gEcah', Name:
  'test1', ...
```

```
}, {
```

```
  ARN: 'arn:aws:secretsmanager:us-east-1:1234567890:secret:test2-KZwml',
  Name: 'test2', ...
```

```
}
```

```
], NextToken: '2'
```

```
}
```

param filters (List) Filter parameters.

param max_results (int) Max number of results per page.

param next_token (str) Page token.

return (Tuple[List,str]) Returns result list and next token.

- [] 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

2.7.67 ses

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
- [] 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
- [] get_identity_mail_from_domain_attributes
- [X] get_identity_notification_attributes

- [] `get_identity_policies`
- [] `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`
- [] `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`
- [] `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

2.7.68 sns

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_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
- [] publish_batch
- [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

2.7.69 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
- [] receive_message
- [X] remove_permission
- [X] send_message
- [X] send_message_batch
- [X] set_queue_attributes
- [X] tag_queue
- [X] untag_queue

2.7.70 ssm

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
- [] 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
- [] 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_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
- [X] 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
- [X] list_tags_for_resource
- [X] modify_document_permission
- [] put_compliance_items
- [] put_inventory
- [X] put_parameter
- [] register_default_patch_baseline
- [] register_patch_baseline_for_patch_group
- [] register_target_with_maintenance_window
- [] register_task_with_maintenance_window
- [X] remove_tags_from_resource
- [] reset_service_setting
- [] resume_session
- [] send_automation_signal
- [X] 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
- [X] update_document
- [X] 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

2.7.71 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
- [X] describe_state_machine
- [] describe_state_machine_for_execution
- [] get_activity_task
- [X] get_execution_history

- [] list_activities
- [X] list_executions
- [X] list_state_machines
- [] 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
- [X] update_state_machine

2.7.72 sts

Example usage

```
@mock_sts
def test_sts_behaviour():
    boto3.client("sts")
    ...
```

Implemented features for this service

- [X] assume_role
- [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

2.7.73 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
- [] describe_attachment
- [X] describe_cases
- [] 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
- [X] refresh_trusted_advisor_check
- [X] resolve_case

2.7.74 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

2.7.75 timestream-write

Example usage

```
@mock_timestreamwrite
def test_timestream-write_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
- [] list_tags_for_resource
- [] tag_resource
- [] untag_resource
- [X] update_database
- [X] update_table
- [X] write_records

2.7.76 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
- [X] delete_medical_vocabulary
- [X] delete_transcription_job
- [X] delete_vocabulary
- [] delete_vocabulary_filter
- [] describe_language_model
- [] get_call_analytics_category
- [] get_call_analytics_job
- [X] get_medical_transcription_job
- [X] get_medical_vocabulary
- [X] get_transcription_job
- [X] get_vocabulary
- [] get_vocabulary_filter
- [] list_call_analytics_categories
- [] list_call_analytics_jobs
- [] list_language_models
- [X] list_medical_transcription_jobs
- [X] list_medical_vocabularies
- [] list_tags_for_resource
- [X] list_transcription_jobs
- [X] list_vocabularies
- [] list_vocabulary_filters
- [] start_call_analytics_job
- [X] start_medical_transcription_job
- [X] start_transcription_job
- [] tag_resource
- [] untag_resource
- [] update_call_analytics_category
- [] update_medical_vocabulary
- [] update_vocabulary
- [] update_vocabulary_filter

2.7.77 wafv2

`class moto.wafv2.models.WAFV2Backend(region_name=None)`
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

- [] `associate_web_acl`
- [] `check_capacity`
- [] `create_ip_set`
- [] `create_regex_pattern_set`
- [] `create_rule_group`
- [X] `create_web_acl`
- [] `delete_firewall_manager_rule_groups`
- [] `delete_ip_set`
- [] `delete_logging_configuration`
- [] `delete_permission_policy`
- [] `delete_regex_pattern_set`
- [] `delete_rule_group`
- [] `delete_web_acl`
- [] `describe_managed_rule_group`
- [] `disassociate_web_acl`
- [] `get_ip_set`
- [] `get_logging_configuration`
- [] `get_managed_rule_set`
- [] `get_permission_policy`
- [] `get_rate_based_statement_managed_keys`
- [] `get_regex_pattern_set`
- [] `get_rule_group`
- [] `get_sampled_requests`
- [] `get_web_acl`
- [] `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_regex_pattern_sets`
- [] `list_resources_for_web_acl`
- [] `list_rule_groups`
- [] `list_tags_for_resource`
- [X] `list_web_acls`
- [] `put_logging_configuration`
- [] `put_managed_rule_set_versions`
- [] `put_permission_policy`
- [] `tag_resource`
- [] `untag_resource`
- [] `update_ip_set`
- [] `update_managed_rule_set_version_expiry_date`
- [] `update_regex_pattern_set`
- [] `update_rule_group`
- [] `update_web_acl`

2.8 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 [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.

2.9 Installation

This is a guide how to install Moto for contributors.

The following software is assumed to be present:

- Python 3.x
- Docker

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.

2.10 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.

2.10.1 Decorator Architecture

When using decorators, Moto works by intercepting the HTTP request that is send 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 send. 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.

2.10.2 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 ore 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 send 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.

2.10.3 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

2.11 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:

```
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
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.

2.12 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/spulec/moto/issues>

2.13 FAQ for Developers

2.13.1 When running the linter...

Why does black give different results?

Different versions of black produce different results. To ensure that our CI passes, please format the code using `black==19.10b0`.

2.13.2 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.

2.13.3 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.

2.13.4 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]"
```

2.14 Development Tips

Below you can find some tips that might help during development.

2.14.1 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()")
```

2.15 Intercepting URL's

2.15.1 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)
```

2.16 Writing tests

One test should only verify a single feature/method. I.e., one test for *create_resource()*, another for *update_resource()*, etc.

2.16.1 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.

2.16.2 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/..
```


2.16.3 Parallel tests

To speed up our CI, the ServerMode 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 Utilities

2.17.1 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.

2.17.2 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,
        #
        # The collection of keys/attributes that should guarantee uniqueness for a
↳ given resource.
        # For most resources it will just be an ID, or ARN, which is always unique.
        # In order to know what is the last resource that we're sending back, an
↳ encoded version of these attributes is used as the NextToken.
        "page_ending_range_keys": ["start_date", "execution_arn"],
    },
    # 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 list the 'next_token'/'max_results'-arguments
# The decorator uses them, but our logic doesn't need them
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

```

INDEX

C

CloudTrailBackend (*class in moto.cloudtrail.models*),
[29](#)

D

DirectoryServiceBackend (*class in moto.ds.models*),
[46](#)

E

EFSBackend (*class in moto.efs.models*), [67](#)

EMRContainersBackend (*class in*
moto.emrcontainers.models), [76](#)

F

FirehoseBackend (*class in moto.firehose.models*), [79](#)

R

Route53ResolverBackend (*class in*
moto.route53resolver.models), [128](#)

S

S3Backend (*class in moto.s3.models*), [130](#)

W

WAFV2Backend (*class in moto.wafv2.models*), [156](#)