django-storages-ec2 Documentation

Release 1.1.0

Chathan Driehuys

Jun 14, 2017

Contents

CHAPTER 1

django-storages-ec2

Extension for django-storages when running on EC2 instances.

Project Home Page: https://github.com/cdriehuys/django-storages-ec2

Documentation: http://django-storages-ec2.readthedocs.io

Issues: https://github.com/cdriehuys/django-storages-ec2/issues

Rationale

Using S3 to store static files makes a lot of sense. The django-storages app makes this easy. What doesn't make sense is having to provide dedicated credentials to allow S3 access when the project is already running on an EC2 instance. This app allows you to use the IAM role attached to your EC2 instance to grant the instance access to S3.

Development

To get started with development, clone the repository and install the requirements:

```
$ git clone https://github.com/cdriehuys/django-storages-ec2
$ cd django-storages-ec2
$ python setup.py install
$ pip install -r requirements-dev.txt
```

Contributing

Contributions to this project are welcome. If you would like to add on a feature, create a branch, commit your changes, and submit a pull request:

```
$ git checkout -b my-new-feature
$ ...
$ git commit -a -m 'Add my new feature'
```

Tests

Tests are run with pytest. You can run the project's tests by running:

```
$ pytest
```

Tests are automatically run with Travis CI. In addition to running the tests, it will also lint the project's code with flake8.

License

This project is licensed under the MIT license.

CHAPTER 2

Table of Contents

Overview

django-storages-ec2 provides an S3 storage class building off the one provided by django-storages for use on EC2 instances. It allows authentication without requiring specific AWS credentials by using the instance's metadata to obtain temporary credentials with permissions dictated by the role attached to the instance.

Requirements

- Python 2.7, 3.4, 3.5, or 3.6
- Django >= 1.8
- django-storages > 1.5 (when the backend we build off of was introduced)
- requests >= 2.17

Installation

To get the latest release, install the package through pip:

```
$ pip install django-storages-ec2
```

Configuration

First, add the app to `INSTALLED_APPS` in `settings.py`:

```
INSTALLED_APPS = [
    ...
    'storages_ec2',
    ...
]
```

To configure one of the provided backends, see *Backends*.

Backends

AutoConnectS3Boto3Storage

This backend allows you to store static files using S3 without requiring you to set `AWS_ACCESS_KEY_ID` and `AWS_SECRET_KEY` in your project settings.

To use this backend, in your `settings.py`, set:

```
DEFAULT_FILE_STORAGE = 'storages_ec2.backends.AutoConnectS3Boto3Storage'
```

To allow the `collectstatic` command to store static files in S3, set:

```
STATICFILES_STORAGE = 'storages.ec2.backends.AutoConnectS3Boto3Storage'
```

The available configuration options can be viewed at the django-storages documentation.

Developer Interface

Backends

Storage backends provided by the `storages_ec2` app.

class `storages_ec2.backends.`**`AutoConnectS3Boto3Storage`**`(`**args*, ***kwargs*`)`

S3 storage backend that automatically authenticates.

This storage class generates temporary credentials using EC2 metadata and then authenticates S3 reqeuests with those credentials.

This class is a wrapper around the `S3Boto3Storage` class from the django-storages app.

`bucket`

Get a connection to the instance's bucket.

We can only cache the bucket for the time that the instance's credentials are valid for, so a new connection will be generated if the instance's credentials are about to expire.

`connection`

Get a connection to S3.

`credentials`

dict – Get temporary credentials used to authenticate a request to S3.

`credentials_valid`

bool – Determine if the instance's credentials are valid.

`instance_role`
str – The name of the role being used by the current instance.

Changelog

v1.1

This release adds caching of connections, so it should result in decreased response times when resources are fetched from S3.

Features:

- #3: Cache S3 connection.
- #6: Cache bucket connection.

Bugfixes:

- #5: Fixed issue with connection failing when temporary credentials expired.

v1.0

Initial production ready release. The app is now being used successfully in a production environment, so we can deploy the first major version release. No code changes have actually been made since v0.1.1.

v0.1.1

Bugfixes:

- #2: Fix wrong URL for determining the instance's role name

v0.1

Initial release.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index