Ubuntu AMI Locator Documentation

Release 1.3.0

Thomas Orozco

July 06, 2014

Contents

Contents:

CHAPTER 1

Usage

1.1 Installation

Use `pip`:

```
$ pip install --upgrade ubuntufinder
```

1.2 CLI Usage

```
$ ubuntufinder -r precise -a amd64 -i ebs -v paravirtual us-east-1
ami-fa7dba92
```

Run `ubuntufinder -h` for usage information.

1.3 Library Usage

```
>>> import ubuntufinder
>>> ubuntufinder.find_image("us-east-1")
<Image: raring@us-east-1: 2013-08-24 00:00:00 (amd64 ebs paravirtual)>
>>> image = ubuntufinder.find_image("us-west-1", "precise", "amd64", "ebs", "paravirtual")
>>> image.ami_id
'ami-c4072e81'
```

1.4 Advanced Usage

`find_image` uses sane defaults, but accepts extra arguments that let you customize your search.

`ubuntufinder.`**`find_image`**(*region*, *release='latest'*, *architecture='amd64'*, *instance_type='ebs'*, *virtualization='paravirtual'*, *_session=None*)

Return a full image specification according to the query parameters

Parameters

- **region** (`str`) – The AWS region code to locate the Image in
- **release** (`str`) – The codename of the Ubuntu Release to locate. Defaults to the latest release
- **architecture** (`str`) – The CPU Architecture to find the image for. Defaults to amd64

- **instance_type** (`str`) – The Instance type to find the image for (ebs or instance-store). Defaults to ebs.
- **virtualization** (`str`) – The virtualization technology to find the image for (paravirtual or hvm).

Returns An Image corresponding to your search

Return type `ubuntufinder.models.Image`

Raises `ubuntufinder.exceptions.ImageNotFound` if no match is found.

Raises `ubuntufinder.exceptions.ServiceUnavailable` if Cloud Images can't be accessed.

CHAPTER 2

Installation

At the command line:

```
$ easy_install ubuntufinder
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv ubuntufinder
$ pip install ubuntufinder
```

CHAPTER 3

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at https://github.com/krallin/ubuntufinder/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

3.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

3.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

3.1.4 Write Documentation

Ubuntu AMI Locator could always use more documentation, whether as part of the official Ubuntu AMI Locator docs, in docstrings, or even on the web in blog posts, articles, and such.

3.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/krallin/ubuntufinder/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

3.2 Get Started!

Ready to contribute? Here's how to set up *ubuntufinder* for local development.

1. Fork the *ubuntufinder* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/ubuntufinder.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv ubuntufinder
$ cd ubuntufinder/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 ubuntufinder tests
    $ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/krallin/ubuntufinder/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_ubuntufinder
```

CHAPTER 4

Credits

4.1 Development Lead

- Thomas Orozco <thomas@orozco.fr>

4.2 Contributors

None yet. Why not be the first?

CHAPTER 5

History

5.1 1.3.0 (2014-07-06)

- Python 3 support

5.2 1.2.0 (2014-07-06)

- CLI: Add meaningful exit codes

5.3 1.1.0 (2014-07-06)

- CLI: Add backwards compatibility with `--instance-type`

5.4 1.0.0 (2014-07-06)

- CLI: Document in README
- CLI: Add help text
- CLI: Rename `--instance-type` parameter `--image-type`

5.5 0.4.0 (2013-08-25)

- improved exception handling

5.6 0.3.0 (2013-08-25)

- Packaging housekeeping

5.7 0.2.0 (2013-08-24)

- Added tests

5.8 0.1.0 (2013-08-24)

- First release on PyPI.
- Locate Images on Ubuntu Cloud Images

CHAPTER 6

Indices and tables

- *genindex*
- *modindex*
- *search*