rdo Documentation

Release 0.2.2

Eric Larson

September 04, 2015

Contents

rdo stands for “Remote DO”

If you work on a project via a virtual machine or on a remote server, *rdo* lets you run commands locally as if they were on the remote machine.

CHAPTER 1

Why *rdo*?

Like many programmers, I work on code that was intended to run on a specific platform. Tools such as docker and Vagrant are helpful in this regard, but it never feels like you're developing locally. The result is that the local tools you have on your machine go unused as you struggle to work on a project through a terminal.

The goal of *rdo* is to allow an easy to way to run your commands as if it were local, while running it on the necessary platform.

CHAPTER 2

Usage

The first step is to create a *.rdo.conf* file. This file is read by the *rdo* command and describes the machine you'll be peforming command on.

Here is an example using *rdo* with Vagrant.

```
[default]
driver = vagrant
directory = /vagrant
```

By default Vagrant will mount the directory of your *Vagrantfile* at */vagrant*. If you have your *Vagrantfile* at the root of your project, this will make *rdo* act like it is running in the same directory.

With your *.rdo.conf* in place you can try running a command.

```
$ rdo ls -la
```

This should provide a list of your project files from the host machine.

Contents:

2.1 Drivers

Drivers for *rdo* define the different ways to connect and run a command on a remote machine.

2.1.1 The Vagrant Driver

The Vagrant driver tries to re-use the *vagrant* command line to run commands. For example *rdo ls -la* is the same as:

```
$ vagrant ssh -c "cd /vagrant && ls -la"
```

The vagrant driver currently supports changing directories before running commands.

2.1.2 The SSH Driver

The SSH driver allows connecting to any machine via the *ssh* command. Here is an annotated example config to show the currently supported options.

```
[default]
driver = ssh
ssh = putty
user = eric
host = example.com
directory = /opt/myapp
ident = ~/.ssh/mycloud.pem
flags = -p 2222
```

Running *rdo ls -la* then would result in the following command:

```
$ putty -i /home/eric/.ssh/mycloud.pem -p 2222 eric@example.com "cd /opt/myapp && ls -la"
```

2.1.3 The Docker Driver

The Docker driver tries to use a docker container to run a command. The config can specify whether to use an or running container.

```
[default]
driver = docker
name = ubuntu
```

Running a *rdo ls -la* then results in the following command:

```
$ docker run -it ubuntu ls -la
```

You can use *exec = true* in the *.rdo.conf* in order to use *exec* rather than *run*, the only caveat is that you need to be sure the name is a running container.

2.1.4 A Note About Escaping

Currently drivers don't do anything terribly special about escaping the command. At the moment, I'm assuing YAGNI, but if I'm wrong, please open an issue.

2.2 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:

2.2.1 Types of Contributions

Report Bugs

Report bugs at https://github.com/ionrock/rdo/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.

Fix Bugs

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

Implement Features

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

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/ionrock/rdo/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 :)

2.2.2 Get Started!

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

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

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

```
$ cd rdo/
$ make bootstrap
```

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:

```
$ make tests
$ make lint
```

   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.

2.2.3 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 2.7, 3.3, and 3.4. Check https://travis-ci.org/ionrock/rdo/pull_requests and make sure that the tests pass for all supported Python versions.

2.2.4 Tips

The Makefile defines how different project tasks are performed. Please use it to tailor development to your own tastes and use the make tasks before making a PR.

The *make bootstrap* creates a virtualenv in the project directory at *venv*.

The tests use py.test for running tests. You can use it directly to run a subset of tests:

```
$ venv/bin/py.test tests/test_foo.py::TestCase::test_func
```

2.3 Credits

2.3.1 Development Lead

- Eric Larson <eric@ionrock.org>

2.3.2 Contributors

None yet. Why not be the first?

CHAPTER 3

Indices and tables

- genindex
- modindex
- search