
saws Documentation

Release 0.1.0

Donne Martin

Jun 03, 2018

Contents

1	Source Code	1
2	SAWS	3
2.1	Motivation	3
2.2	Index	4
2.3	Syntax and Output Highlighting	6
2.4	Auto-Completion of Commands, Subcommands, and Options	6
2.5	Auto-Completion of AWS Resources	8
2.6	Customizable Shortcuts	9
2.7	Fish-Style Auto-Suggestions	11
2.8	Executing Shell Commands	11
2.9	Command History	13
2.10	Contextual Help	13
2.11	Toolbar Options	13
2.12	Windows Support	14
2.13	Installation	16
2.14	Developer Installation	17
2.15	Contributing	18
2.16	Credits	18
2.17	Contact Info	18
2.18	License	19

CHAPTER 1

Source Code

- [GitHub Repo](#)
- `genindex`
- `modindex`
- `search`

2.1 Motivation

2.1.1 AWS CLI

Although the [AWS CLI](#) is a great resource to manage your AWS-powered services, it's **tough to remember usage** of:

- 70+ top-level commands
- 2000+ subcommands
- Countless command-specific options
- Resources such as instance tags and buckets

2.1.2 SAWS: A Supercharged AWS CLI

SAWS aims to **supercharge** the AWS CLI with features focusing on:

- **Improving ease-of-use**
- **Increasing productivity**

Under the hood, SAWS is **powered by the AWS CLI** and supports the **same commands** and **command structure**.

SAWS and AWS CLI Usage:

```
aws <command> <subcommand> [parameters] [options]
```

SAWS features:

- Auto-completion of:

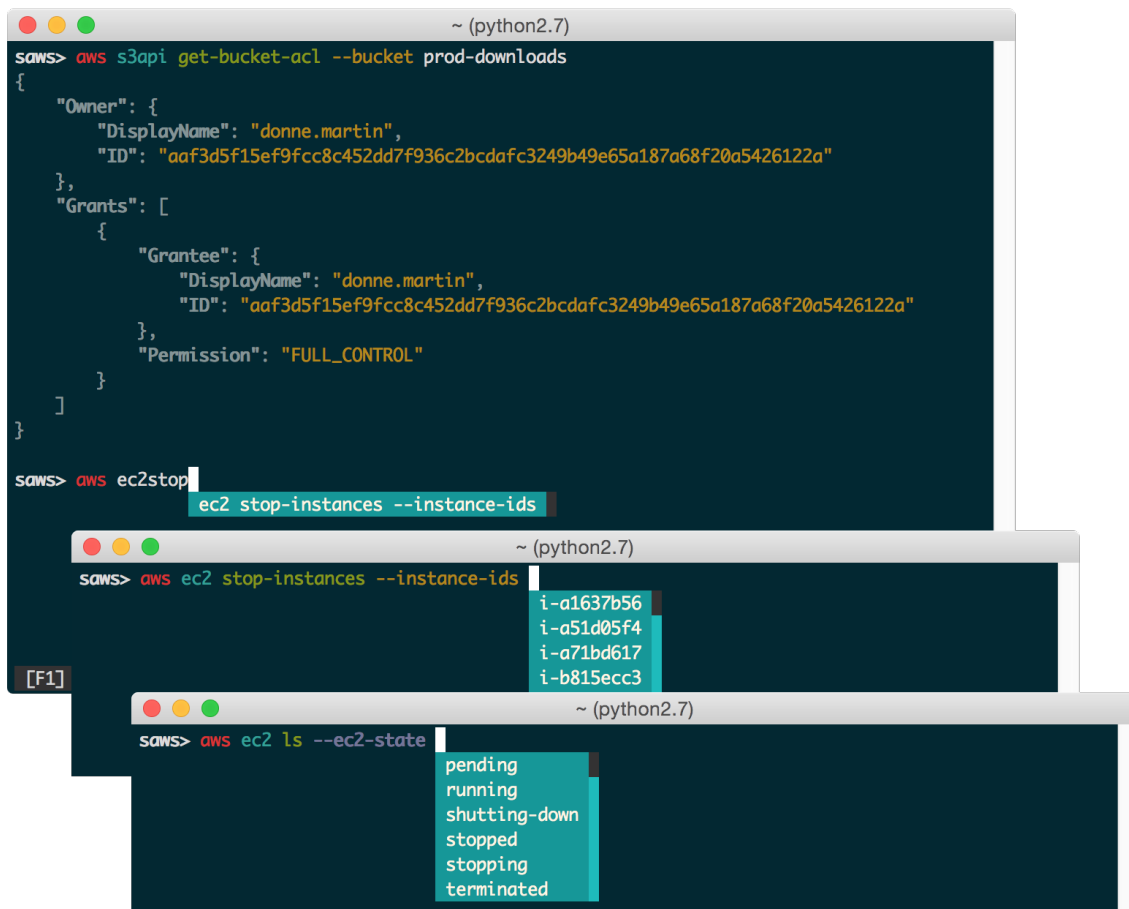
- Commands
 - Subcommands
 - Options
- Auto-completion of resources:
 - Bucket names
 - Instance ids
 - Instance tags
 - *More coming soon!*
- Customizable shortcuts
- Fuzzy completion of resources and shortcuts
- Fish-style auto-suggestions
- Syntax and output highlighting
- Execution of shell commands
- Command history
- Contextual help
- Toolbar options

SAWS is available for Mac, Linux, Unix, and *Windows*.

2.2 Index

2.2.1 Features

- *Syntax and Output Highlighting*
- *Auto-Completion of Commands, Subcommands, and Options*
- *Auto-Completion of AWS Resources*
 - *S3 Buckets*
 - *EC2 Instance Ids*
 - *EC2 Instance Tags*
 - *TODO: Add More Resources*
- *Customizable Shortcuts*
- *Fuzzy Resource and Shortcut Completion*
- *Fish-Style Auto-Suggestions*
- *Executing Shell Commands*
- *Command History*
- *Contextual Help*
 - *Contextual Command Line Help*
 - *Contextual Web Docs*



The image shows three overlapping terminal windows, each with a title bar indicating the environment is `~ (python2.7)`.

The top window shows the command `saws> aws s3api get-bucket-acl --bucket prod-downloads` and its output, a JSON object representing the bucket's ACL. The output shows the owner as `donne.martin` with ID `aaf3d5f15ef9fcc8c452dd7f936c2bcdafc3249b49e65a187a68f20a5426122a`, and a single grant with the same identity and `FULL_CONTROL` permission.

The middle window shows the command `saws> aws ec2 stop-instances --instance-ids` with a list of instance IDs: `i-a1637b56`, `i-a51d05f4`, `i-a71bd617`, and `i-b815ecc3`.

The bottom window shows the command `saws> aws ec2 ls --ec2-state` with a list of EC2 instance states: `pending`, `running`, `shutting-down`, `stopped`, `stopping`, and `terminated`.

```
saws> aws s3api get-bucket-acl --bucket prod-downloads
{
  "Owner": {
    "DisplayName": "donne.martin",
    "ID": "aaf3d5f15ef9fcc8c452dd7f936c2bcdafc3249b49e65a187a68f20a5426122a"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "donne.martin",
        "ID": "aaf3d5f15ef9fcc8c452dd7f936c2bcdafc3249b49e65a187a68f20a5426122a"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}

saws> aws ec2 stop-instances --instance-ids
i-a1637b56
i-a51d05f4
i-a71bd617
i-b815ecc3

saws> aws ec2 ls --ec2-state
pending
running
shutting-down
stopped
stopping
terminated
```

- *Toolbar Options*
- *Windows Support*

2.2.2 Installation and Tests

- *Installation*
 - *Pip Installation*
 - *Virtual Environment and Docker Installation*
 - *AWS Credentials and Named Profiles*
 - *Supported Python Versions*
 - *Supported Platforms*
- *Developer Installation*
 - *Continuous Integration*
 - *Dependencies Management*
 - *Unit Tests and Code Coverage*
 - *Documentation*

2.2.3 Misc

- *Contributing*
- *Credits*
- *Contact Info*
- *License*

2.3 Syntax and Output Highlighting

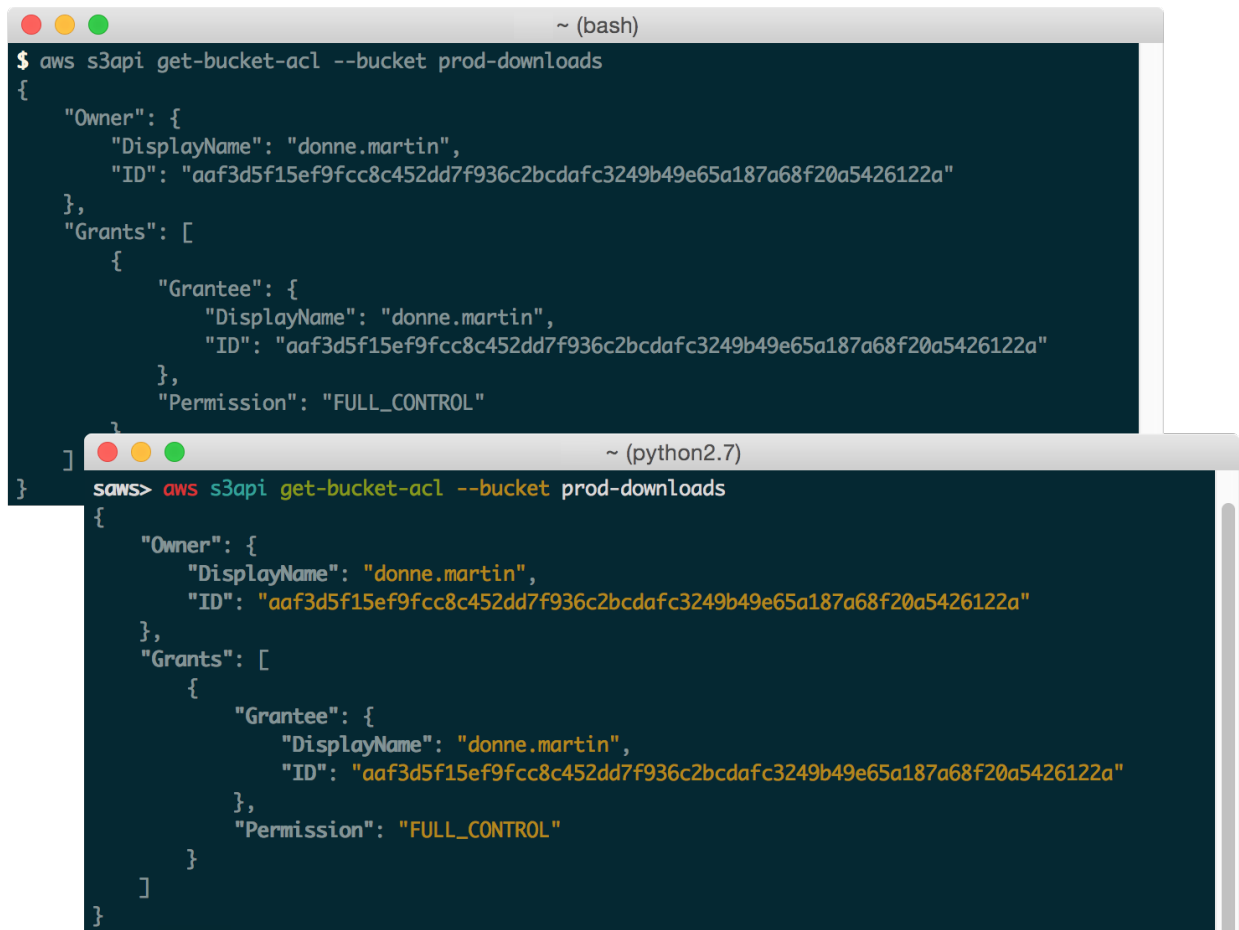
You can control which theme to load for syntax highlighting by updating your `~/.sawsrc` file:

```
# Visual theme. Possible values: manni, igor, xcode, vim, autumn, vs, rrt,  
# native, perldoc, borland, tango, emacs, friendly, monokai, paraiso-dark,  
# colorful, murphy, bw, pastie, paraiso-light, trac, default, fruity  
theme = vim
```

2.4 Auto-Completion of Commands, Subcommands, and Options

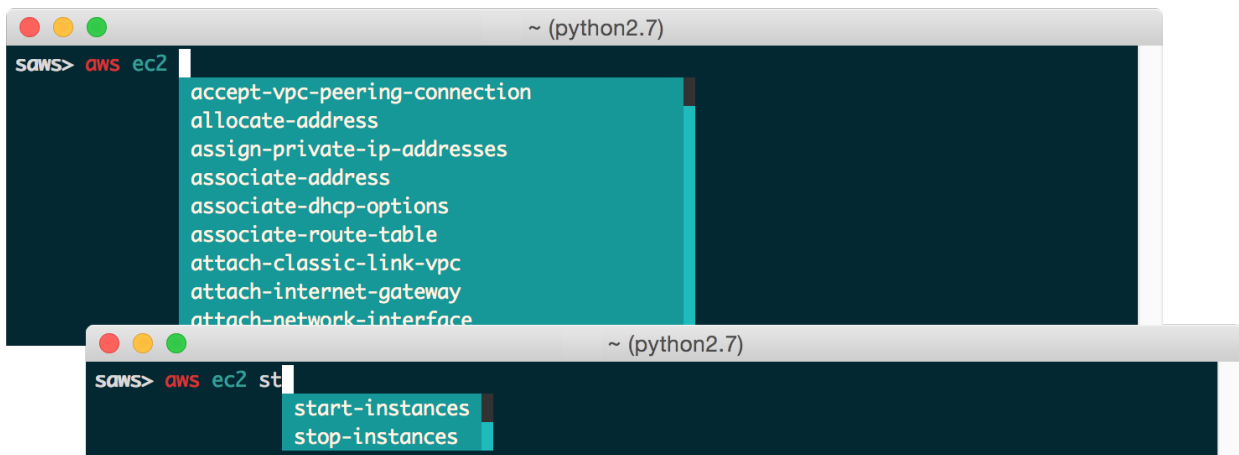
SAWS provides smart autocompletion as you type. Entering the following command will interactively list and auto-complete all subcommands **specific only** to `ec2`:

```
aws ec2
```



```
~ (bash)
$ aws s3api get-bucket-acl --bucket prod-downloads
{
  "Owner": {
    "DisplayName": "donne.martin",
    "ID": "aaf3d5f15ef9fcc8c452dd7f936c2bcdafc3249b49e65a187a68f20a5426122a"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "donne.martin",
        "ID": "aaf3d5f15ef9fcc8c452dd7f936c2bcdafc3249b49e65a187a68f20a5426122a"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

```
~ (python2.7)
saws> aws s3api get-bucket-acl --bucket prod-downloads
{
  "Owner": {
    "DisplayName": "donne.martin",
    "ID": "aaf3d5f15ef9fcc8c452dd7f936c2bcdafc3249b49e65a187a68f20a5426122a"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "donne.martin",
        "ID": "aaf3d5f15ef9fcc8c452dd7f936c2bcdafc3249b49e65a187a68f20a5426122a"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```



```
~ (python2.7)
saws> aws ec2
accept-vpc-peering-connection
allocate-address
assign-private-ip-addresses
associate-address
associate-dhcp-options
associate-route-table
attach-classic-link-vpc
attach-internet-gateway
attach-network-interface
```

```
~ (python2.7)
saws> aws ec2 st
start-instances
stop-instances
```

2.5 Auto-Completion of AWS Resources

In addition to the default commands, subcommands, and options the AWS CLI provides, *SAWS* supports auto-completion of your AWS resources. Currently, bucket names, instance ids, and instance tags are included, with additional support for more resources *under development*.

2.5.1 S3 Buckets

Option for `s3api`:

```
--bucket
```

Sample Usage:

```
aws s3api get-bucket-acl --bucket
```

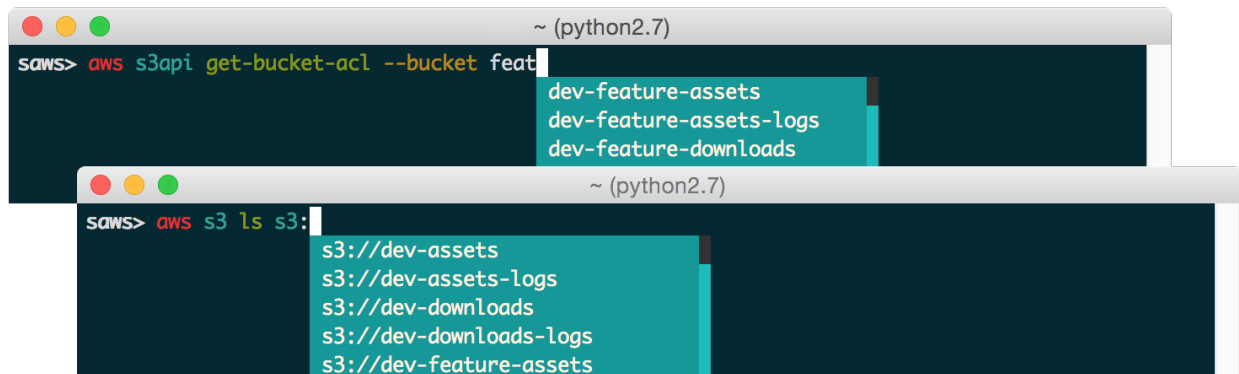
Syntax for `s3`:

```
s3://
```

Sample Usage:

```
aws s3 ls s3://
```

Note: The example below demonstrates the use of *fuzzy resource completion*:



2.5.2 EC2 Instance Ids

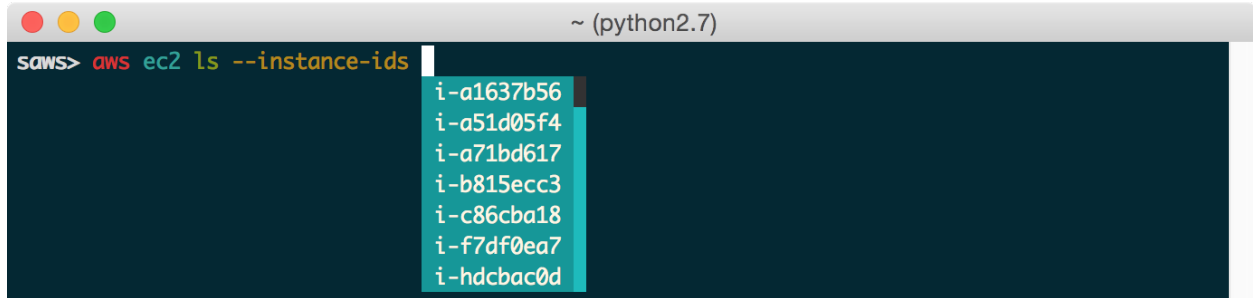
Option for `ec2`:

```
--instance-ids
```

Sample Usage:

```
aws ec2 describe-instances --instance-ids
aws ec2 ls --instance-ids
```

Note: The `ls` command demonstrates the use of *customizable shortcuts*:



```
saws> aws ec2 ls --instance-ids
i-a1637b56
i-a51d05f4
i-a71bd617
i-b815ecc3
i-c86cba18
i-f7df0ea7
i-hdcba0d
```

2.5.3 EC2 Instance Tags

Option for `ec2`:

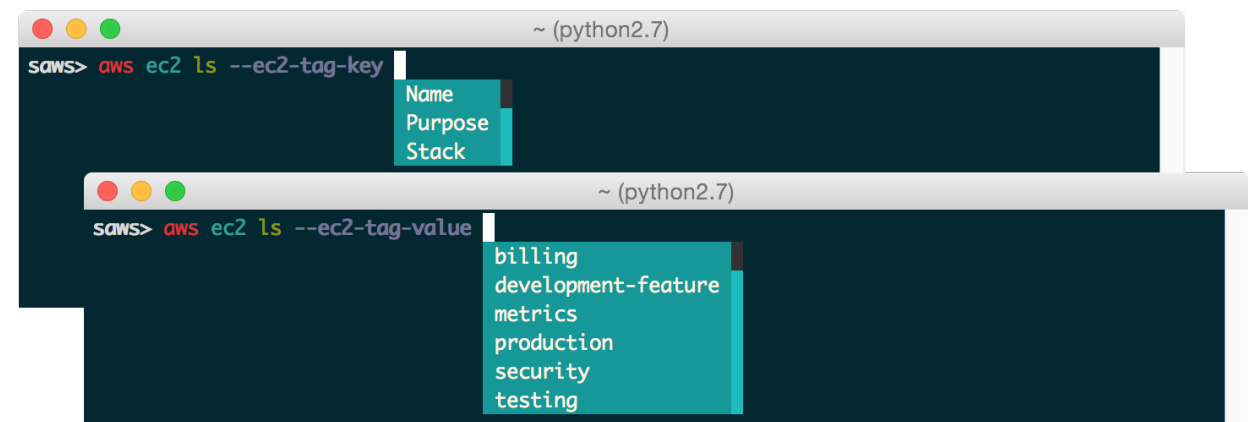
```
--ec2-tag-key
--ec2-tag-value
```

Sample Usage:

```
aws ec2 ls --ec2-tag-key
aws ec2 ls --ec2-tag-value
```

Tags support wildcards with the `*` character.

Note: `ls`, `--ec2-tag-value`, and `--ec2-tag-key` demonstrate the use of *customizable shortcuts*:



```
saws> aws ec2 ls --ec2-tag-key
Name
Purpose
Stack

saws> aws ec2 ls --ec2-tag-value
billing
development-feature
metrics
production
security
testing
```

2.5.4 TODO: Add More Resources

Feel free to *submit an issue or a pull request* if you'd like support for additional resources.

2.6 Customizable Shortcuts

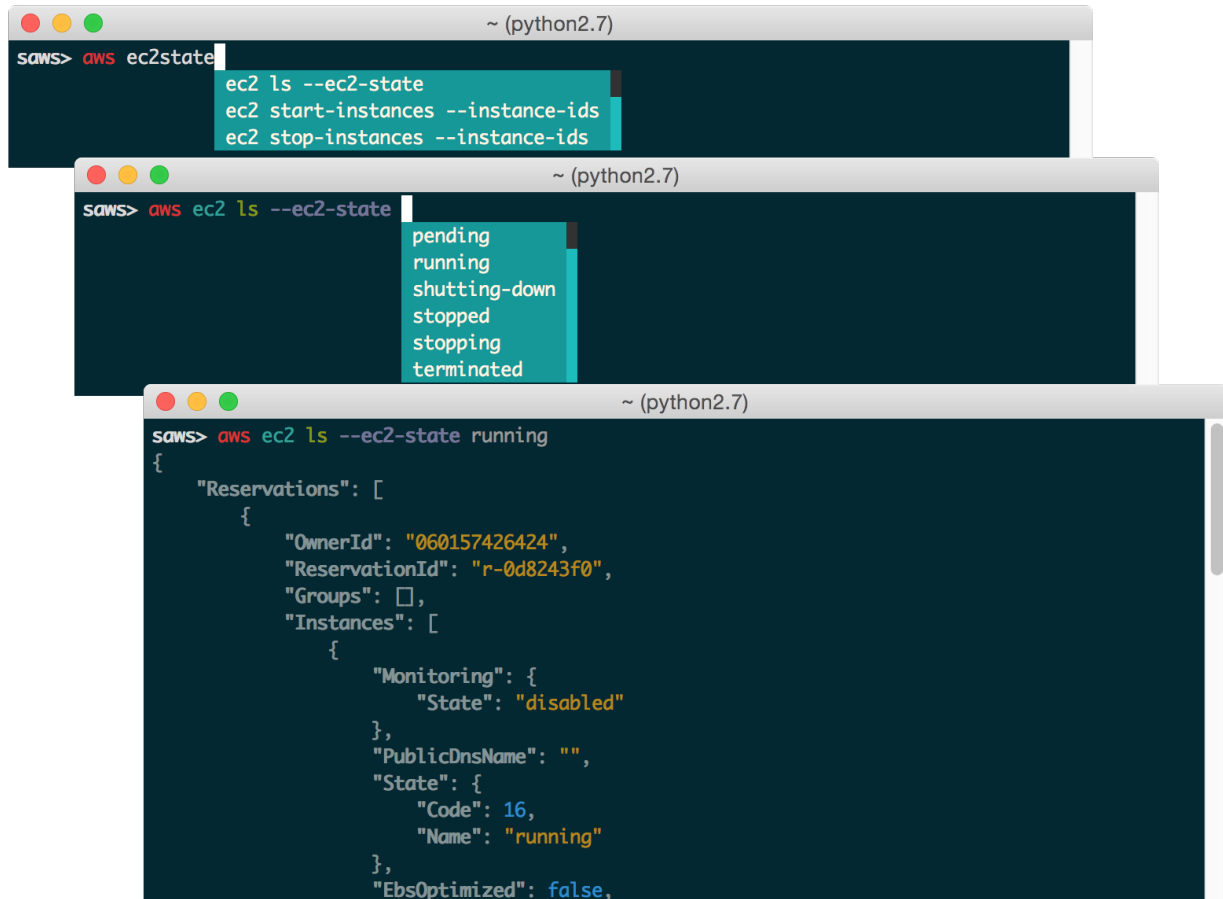
The `~/.saws.shortcuts` file contains shortcuts that you can modify. It comes pre-populated with several *handy shortcuts* out of the box. You can combine shortcuts with *fuzzy completion* for even less keystrokes. Below are a few examples.

List all EC2 instances:

```
aws ec2 ls
```

List all running EC2 instances:

```
aws ec2 ls --ec2-state running # fuzzy shortcut: aws ecstate
```



List all EC2 instances with a matching tag (supports wildcards *):

```
aws ec2 ls --ec2-tag-key # fuzzy shortcut: aws ectagk
aws ec2 ls --ec2-tag-value # fuzzy shortcut: aws ectagv
```

List EC2 instance with matching id:

```
aws ec2 ls --instance-ids # fuzzy shortcut: aws eclsi
```

List all DynamoDB tables:

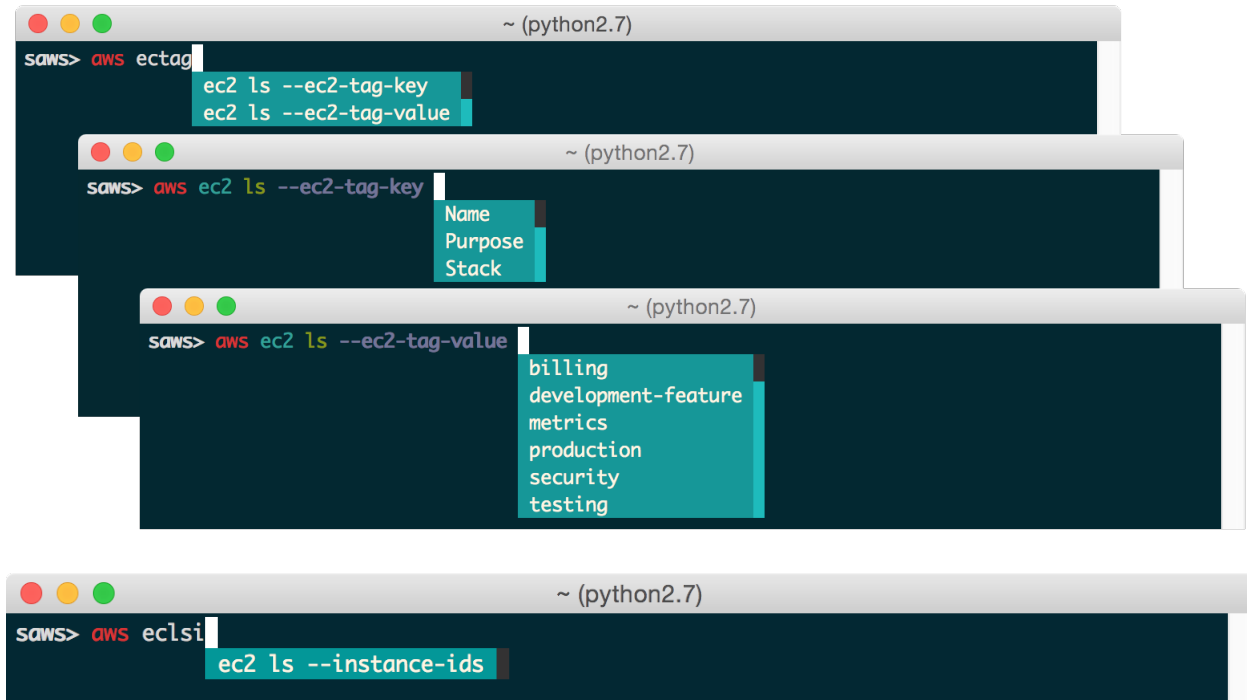
```
aws dynamodb ls # fuzzy shortcut: aws dls
```

List all EMR clusters:

```
aws emr ls # fuzzy shortcut: aws emls
```

Add/remove/modify shortcuts in your `~/.saws.shortcuts` file to suit your needs.

Feel free to submit:



- An issue to request additional shortcuts
- A pull request if you'd like to share your shortcuts (see *contributing guidelines*)

2.6.1 Fuzzy Resource and Shortcut Completion

To toggle fuzzy completion of AWS resources and shortcuts, use F3 key.

Sample fuzzy shortcuts to start and stop EC2 instances:

```
aws ecstop
aws ecstart
```

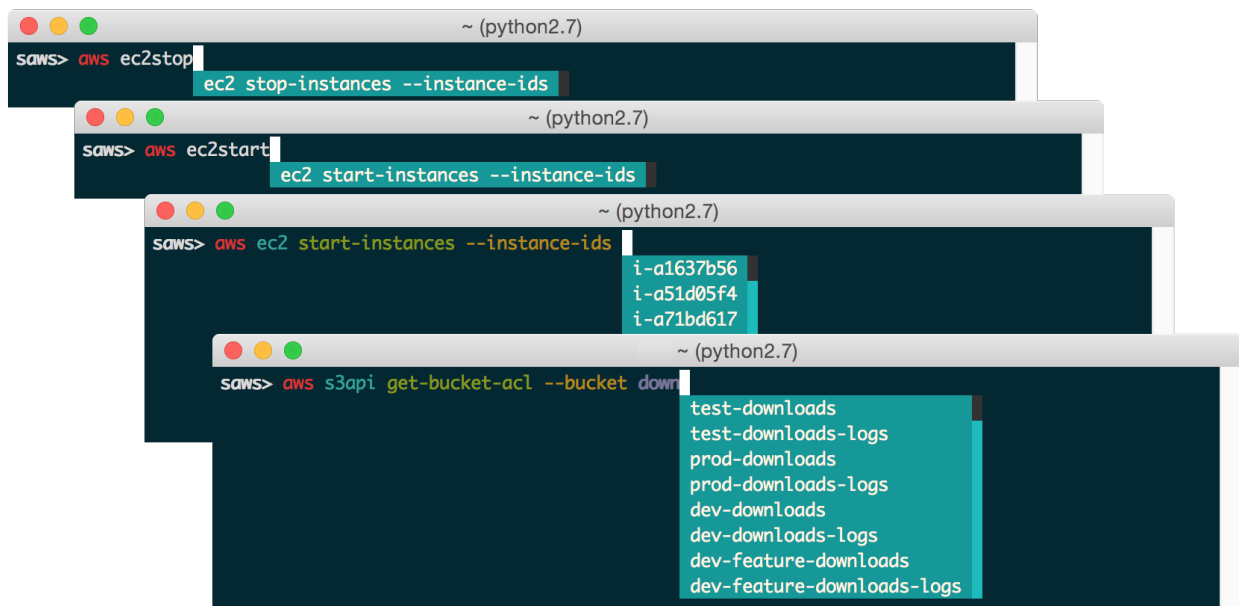
Note: Fuzzy completion currently only works with AWS *resources* and *shortcuts*.

2.7 Fish-Style Auto-Suggestions

SAWS supports Fish-style auto-suggestions. Use the `right arrow` key to complete a suggestion.

2.8 Executing Shell Commands

SAWS allows you to execute shell commands from the `saws>` prompt.

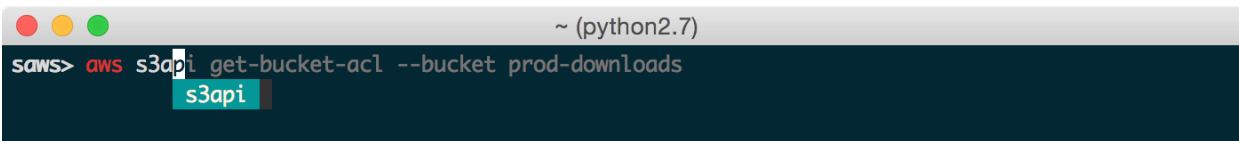


```
~ (python2.7)
saws> aws ec2stop
ec2 stop-instances --instance-ids

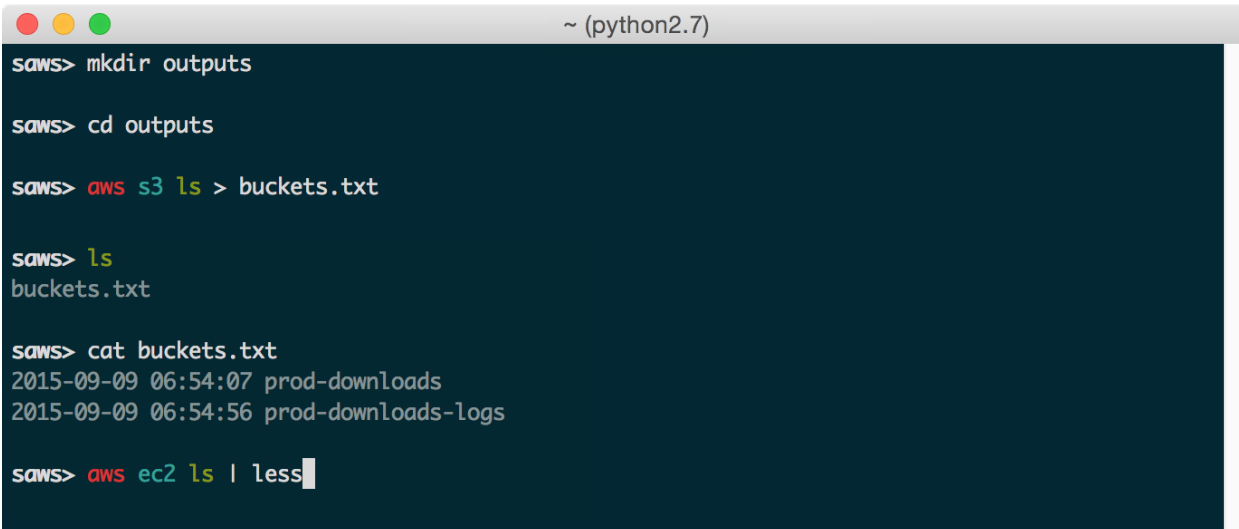
~ (python2.7)
saws> aws ec2start
ec2 start-instances --instance-ids

~ (python2.7)
saws> aws ec2 start-instances --instance-ids
i-a1637b56
i-a51d05f4
i-a71bd617

~ (python2.7)
saws> aws s3api get-bucket-acl --bucket down
test-downloads
test-downloads-logs
prod-downloads
prod-downloads-logs
dev-downloads
dev-downloads-logs
dev-feature-downloads
dev-feature-downloads-logs
```



```
~ (python2.7)
saws> aws s3api get-bucket-acl --bucket prod-downloads
s3api
```



```
~ (python2.7)
saws> mkdir outputs

saws> cd outputs

saws> aws s3 ls > buckets.txt

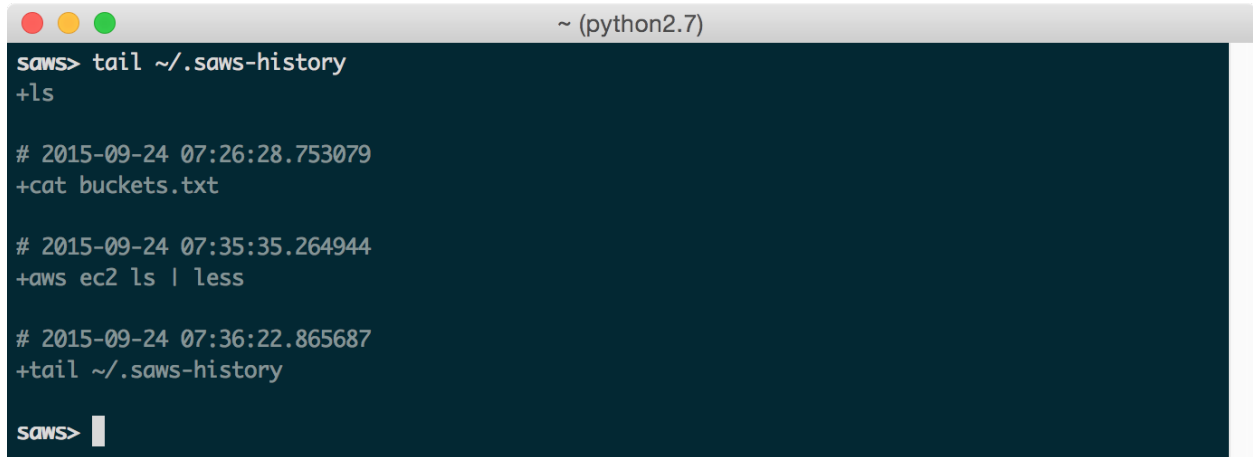
saws> ls
buckets.txt

saws> cat buckets.txt
2015-09-09 06:54:07 prod-downloads
2015-09-09 06:54:56 prod-downloads-logs

saws> aws ec2 ls | less
```


2.9 Command History

SAWS keeps track of commands you enter and stores them in `~/.saws-history`. Use the up and down arrow keys to cycle through the command history.



```

~ (python2.7)
saws> tail ~/.saws-history
+ls

# 2015-09-24 07:26:28.753079
+cat buckets.txt

# 2015-09-24 07:35:35.264944
+aws ec2 ls | less

# 2015-09-24 07:36:22.865687
+tail ~/.saws-history

saws>

```

2.10 Contextual Help

SAWS supports contextual command line help and contextual web docs.

2.10.1 Contextual Command Line Help

The `help` command is powered by the AWS CLI and outputs help within the command line.

Usage:

```
aws <command> <subcommand> help
```

2.10.2 Contextual Web Docs

Sometimes you're not quite sure what specific command/subcommand/option combination you need to use. In such cases, browsing through several combinations with the `help` command line is cumbersome versus browsing the online AWS CLI docs through a web browser.

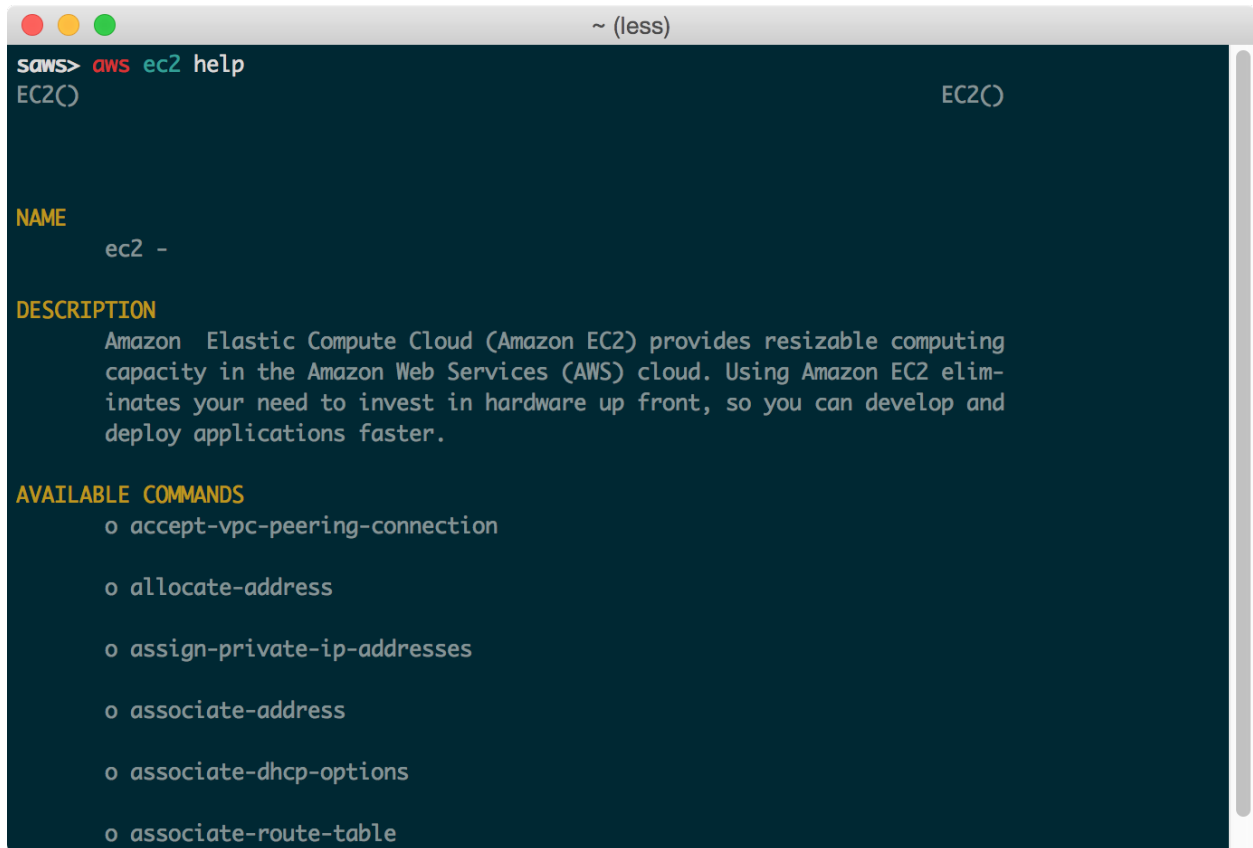
SAWS supports contextual web docs with the `docs` command or the `F9` key. SAWS will display the web docs specific to the currently entered command and subcommand.

Usage:

```
aws <command> <subcommand> docs
```

2.11 Toolbar Options

SAWS supports a number of toolbar options:



```
saws> aws ec2 help
EC2()

NAME
    ec2 -

DESCRIPTION
    Amazon Elastic Compute Cloud (Amazon EC2) provides resizable computing
    capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 elim-
    inates your need to invest in hardware up front, so you can develop and
    deploy applications faster.

AVAILABLE COMMANDS
    o accept-vpc-peering-connection

    o allocate-address

    o assign-private-ip-addresses

    o associate-address

    o associate-dhcp-options

    o associate-route-table
```

- F2 toggles *output syntax highlighting*
- F3 toggles *fuzzy completion of AWS resources and shortcuts*
- F4 toggles *completion of shortcuts*
- F5 refreshes *resources for auto-completion*
- F9 displays the *contextual web docs*
- F10 or control d exits SAWS

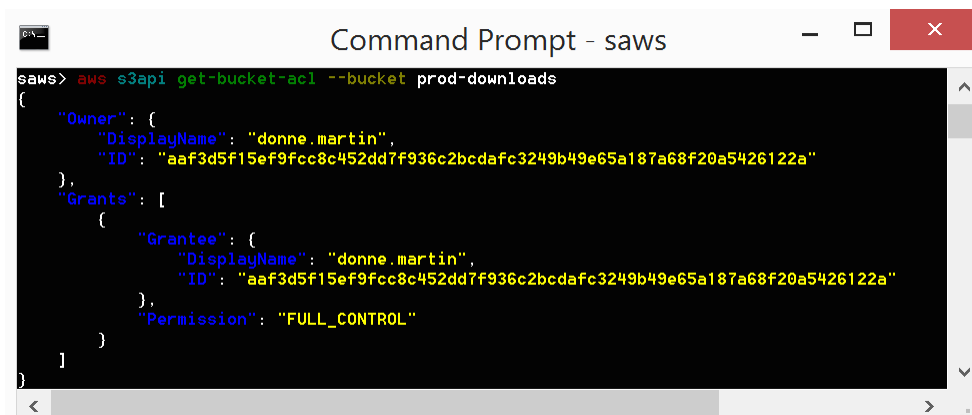
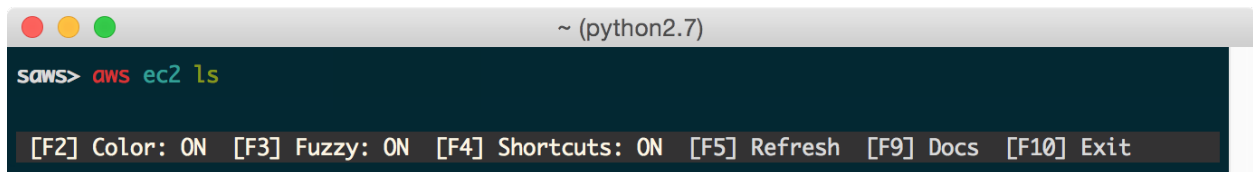
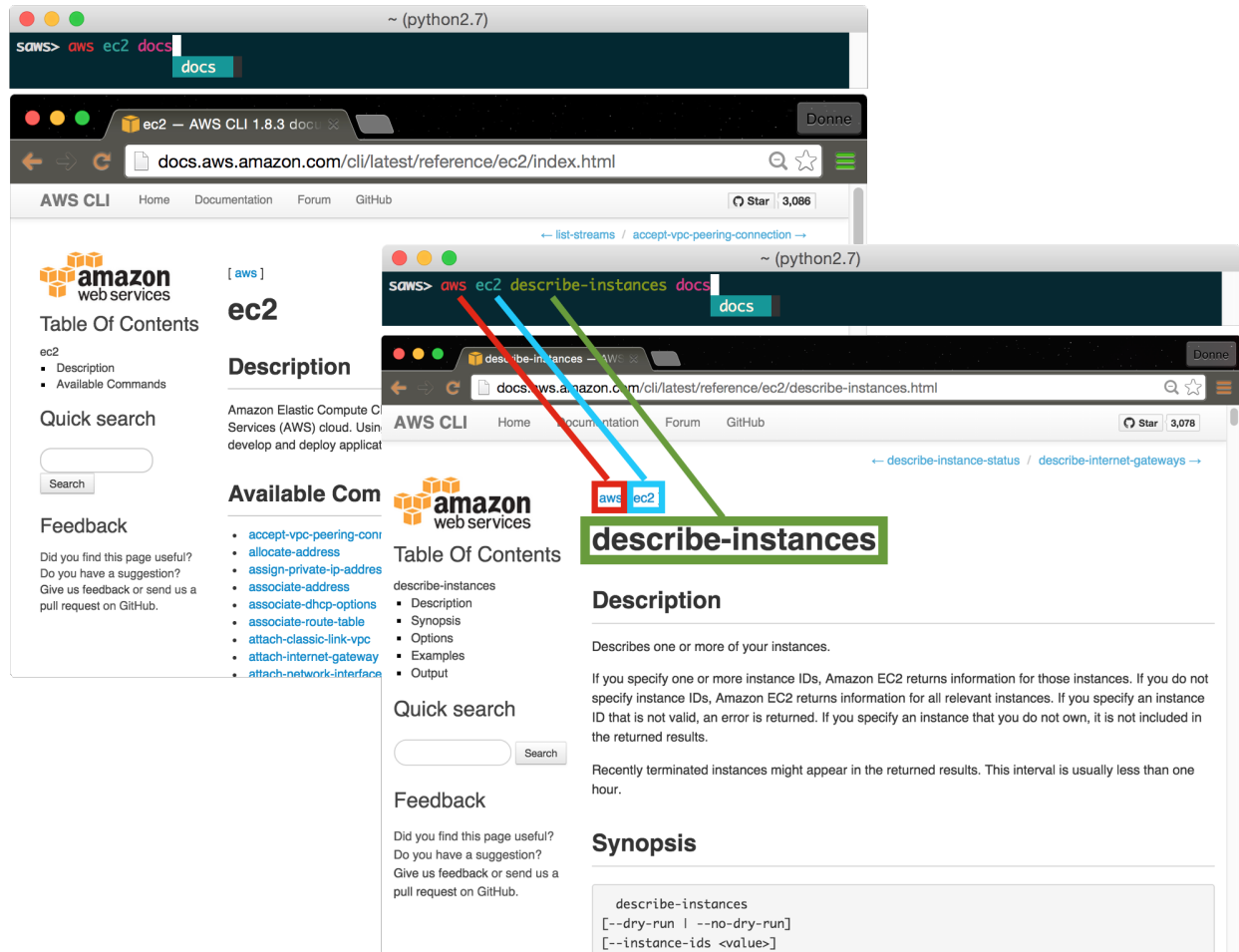
2.12 Windows Support

SAWS has been tested on Windows 7 and Windows 10.

On Windows, the `.sawsrc` file can be found in `%userprofile%`. For example:

```
C:\Users\dmartin\.sawsrc
```

Although you can use the standard Windows command prompt, you'll probably have a better experience with either `cmd.exe` or `conemu`.



2.13 Installation

2.13.1 Pip Installation

SAWS is hosted on [PyPI](#). The following command will install SAWS along with dependencies such as the [AWS CLI](#):

```
$ pip install saws
```

You can also install the latest SAWS from GitHub source which can contain changes not yet pushed to PyPI:

```
$ pip install git+https://github.com/donnemartin/saws.git
```

If you are not installing in a [virtualenv](#), run with `sudo`:

```
$ sudo pip install saws
```

Once installed, start SAWS:

```
$ saws
```

2.13.2 Virtual Environment and Docker Installation

It is recommended that you install Python packages in a [virtualenv](#) to avoid potential [issues with dependencies or permissions](#).

To view SAWS [virtualenv](#) and [Docker](#) installation instructions, click [here](#).

2.13.3 Mac OS X 10.11 El Capitan Users

There is a known issue with Apple and its included python package dependencies (more info at <https://github.com/pypa/pip/issues/3165>). We are investigating ways to fix this issue but in the meantime, to install saws, you can run:

```
$ sudo pip install saws --upgrade --ignore-installed six
```

2.13.4 AWS Credentials and Named Profiles

[Configure your credentials](#) with the AWS CLI:

```
$ aws configure
```

If you'd like to use a specific named profile with SAWS, run the following commands on OS X, Linux, or Unix:

```
$ export AWS_DEFAULT_PROFILE=user1  
$ saws
```

Or as a one-liner:

```
$ AWS_DEFAULT_PROFILE=user1 saws
```

Windows users can run the following commands:

```
> set AWS_DEFAULT_PROFILE=user1
> saws
```

Command line options for starting SAWS with a specific profile are [under development](#). For more details on how to install and configure the AWS CLI, refer to the following [documentation](#).

2.13.5 Supported Python Versions

- Python 2.6
- Python 2.7
- Python 3.3
- Python 3.4
- Pypy

Light testing indicates that SAWS also seems to be compatible with Python 3.5.

Pypy3 is not supported due to [lack of support](#) from [boto](#).

2.13.6 Supported Platforms

- Mac OS X
 - Tested on OS X 10.10
- Linux, Unix
 - Tested on Ubuntu 14.04 LTS
- Windows
 - Tested on Windows 7 and 10

2.14 Developer Installation

If you're interested in contributing to SAWS, run the following commands:

```
$ git clone https://github.com/donnemartin/saws.git
$ pip install -e .
$ pip install -r requirements-dev.txt
$ saws
```

2.14.1 Continuous Integration

Continuous integration details are available on [Travis CI](#).

2.14.2 Dependencies Management

Dependencies management details are available on [Gemnasium](#).

2.14.3 Unit Tests and Code Coverage

Run unit tests in your active Python environment:

```
$ python tests/run_tests.py
```

Run unit tests with `tox` on multiple Python environments:

```
$ tox
```

2.14.4 Documentation

Source code documentation is available on [Readthedocs.org](#).

Run the following to build the docs:

```
$ scripts/update_docs.sh
```

2.15 Contributing

Contributions are welcome!

Review the [Contributing Guidelines](#) for details on how to:

- Submit issues
- Submit pull requests

2.16 Credits

- [AWS CLI](#) by [AWS](#) for powering SAWS under the hood
- [Python Prompt Toolkit](#) by [jonathanslenders](#) for simplifying the creation of SAWS
- [Wharfee](#) by [j-bennet](#) for inspiring the creation of SAWS and for some handy utility functions

2.17 Contact Info

Feel free to contact me to discuss any issues, questions, or comments.

- Email: donne.martin@gmail.com
- Twitter: [donne_martin](#)
- GitHub: [donnemartin](#)
- LinkedIn: [donnemartin](#)

- Website: donnemartin.com

2.18 License

Copyright 2015 Donne Martin

Licensed under the Apache License, Version 2.0 (the "License");
you may **not** use this file **except in** compliance **with** the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law **or** agreed to **in** writing, software
distributed under the License **is** distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express **or** implied.
See the License **for** the specific language governing permissions **and**
limitations under the License.