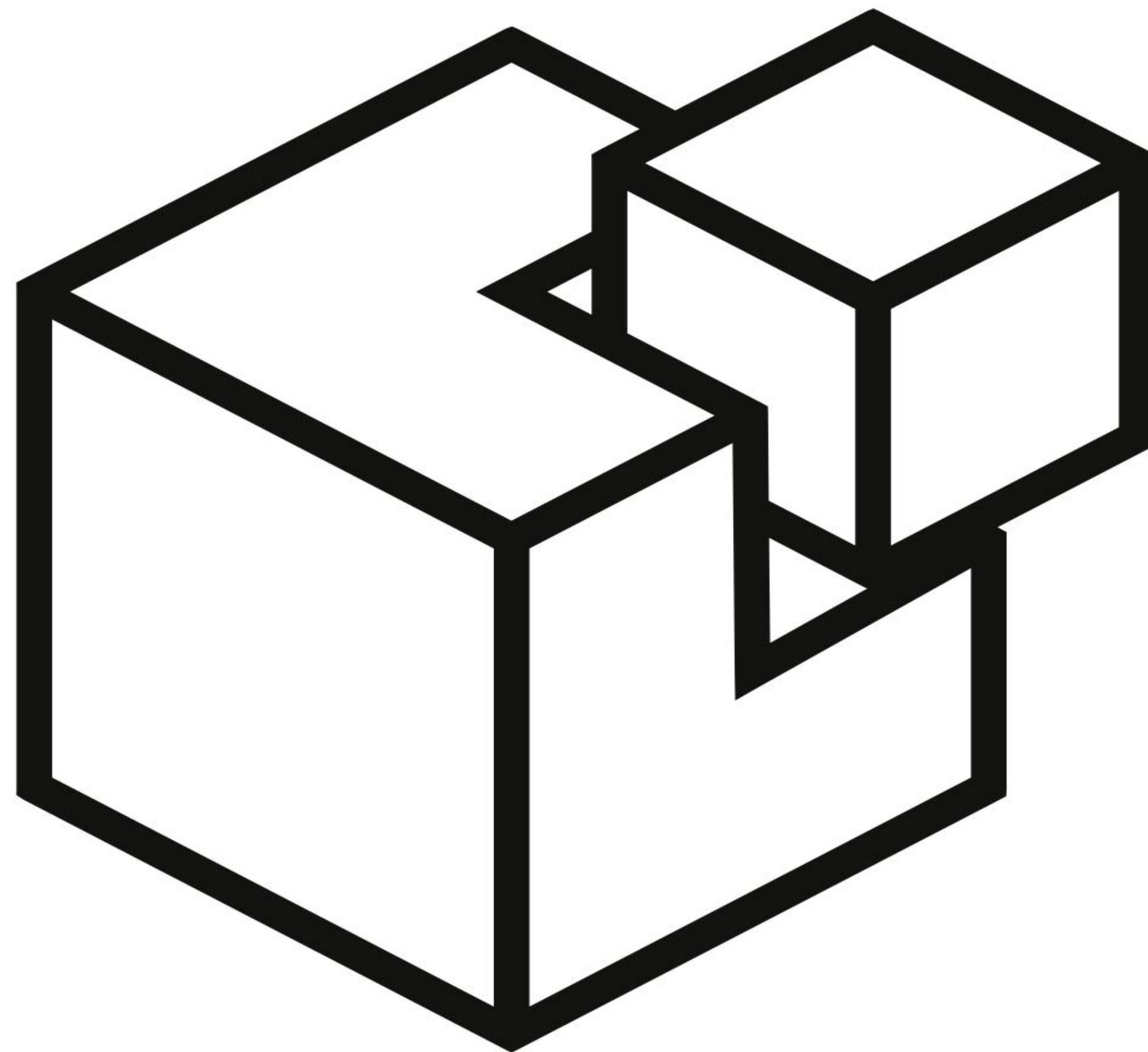


SaltStack for Developers

Extending SaltStack

Hello
my name is

Gareth



SALTSTACK

Open Source Summit North America 2018

 [@garethgreenaway](https://twitter.com/garethgreenaway)

WE ARE
HIRING

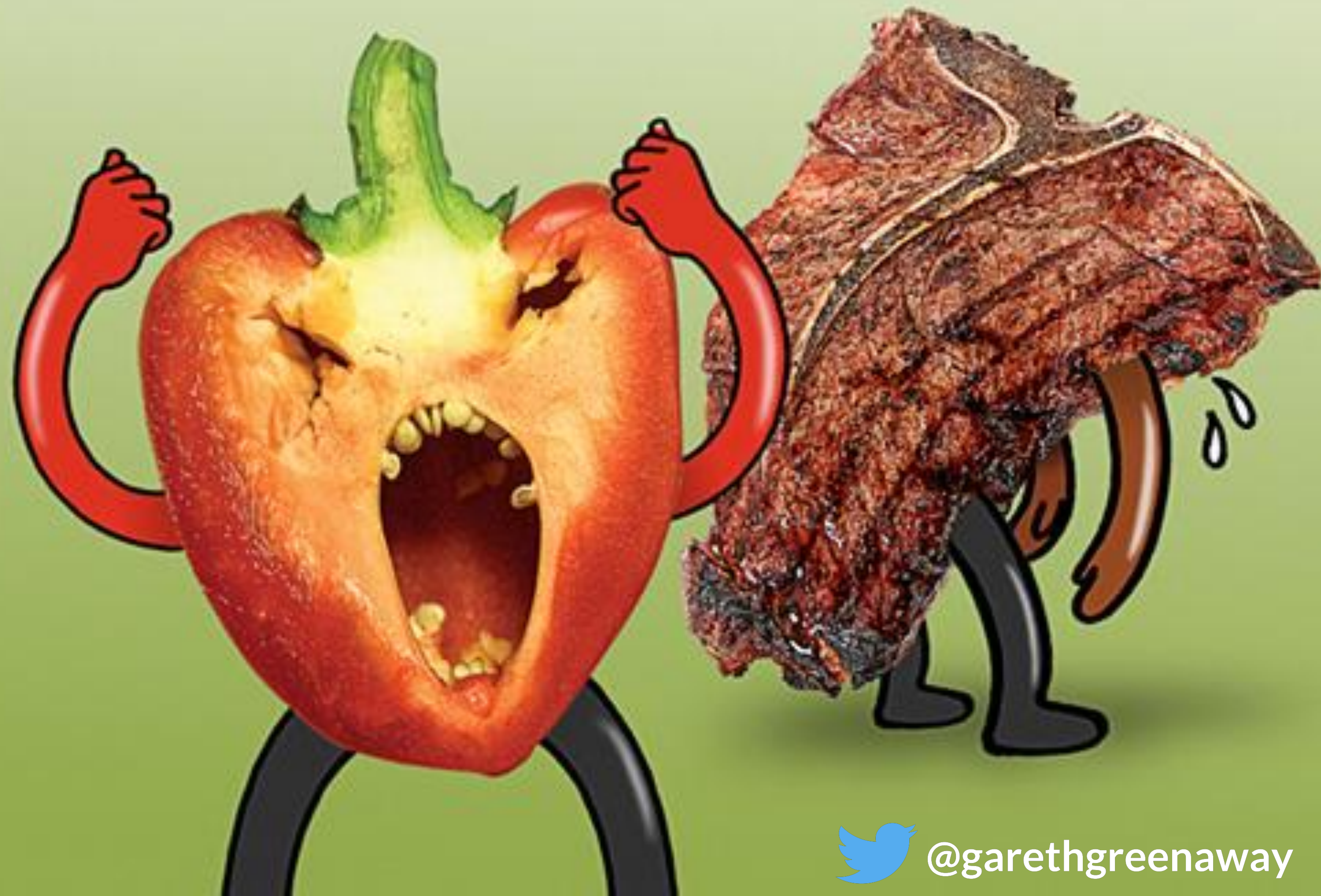


FLOSS WEEKLY



Open Source Summit North America 2018

 @garethgreenaway





SaltStack for Developers

Extending SaltStack

What is SaltStack?

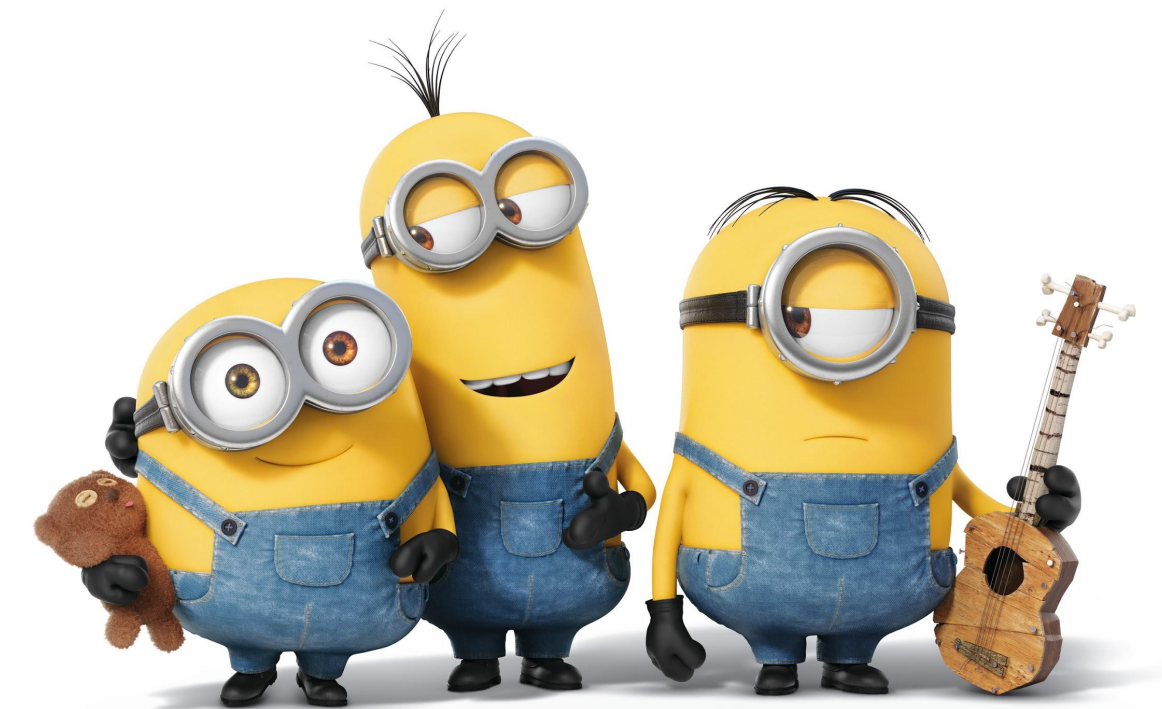
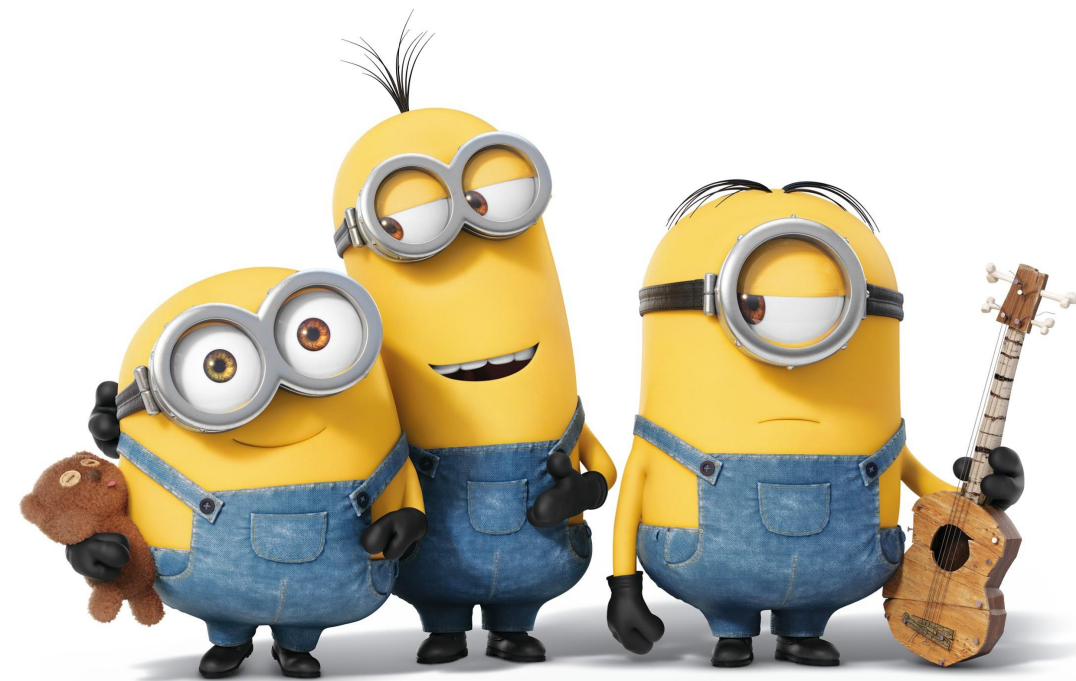
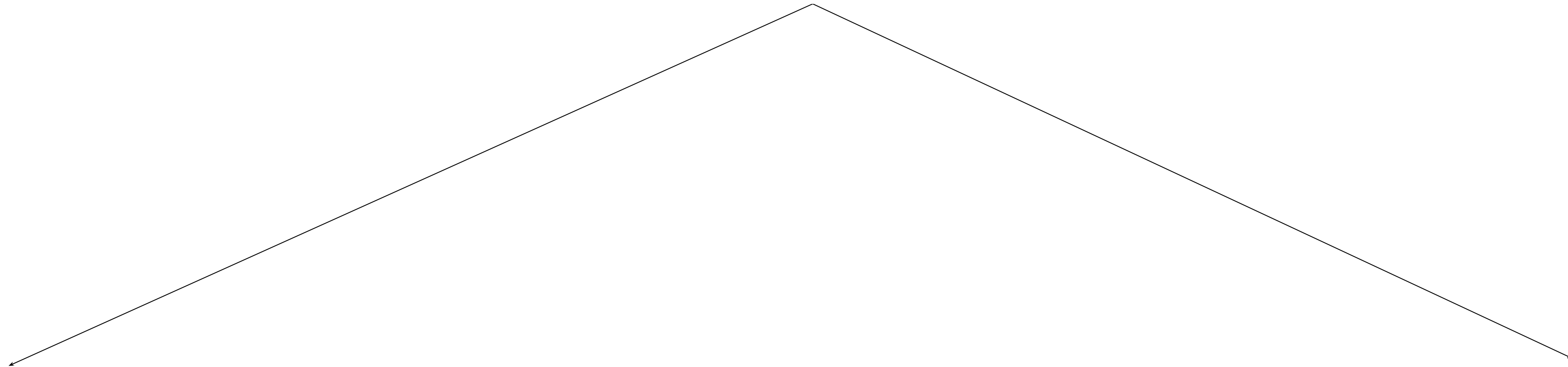
SaltStack == Salt

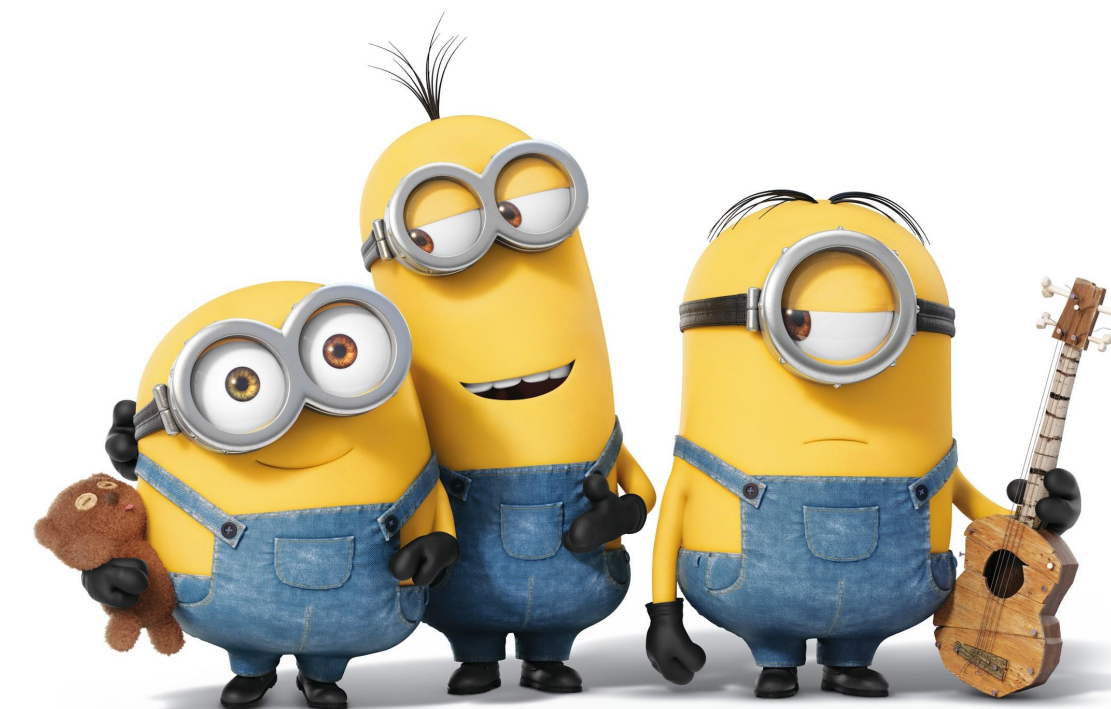
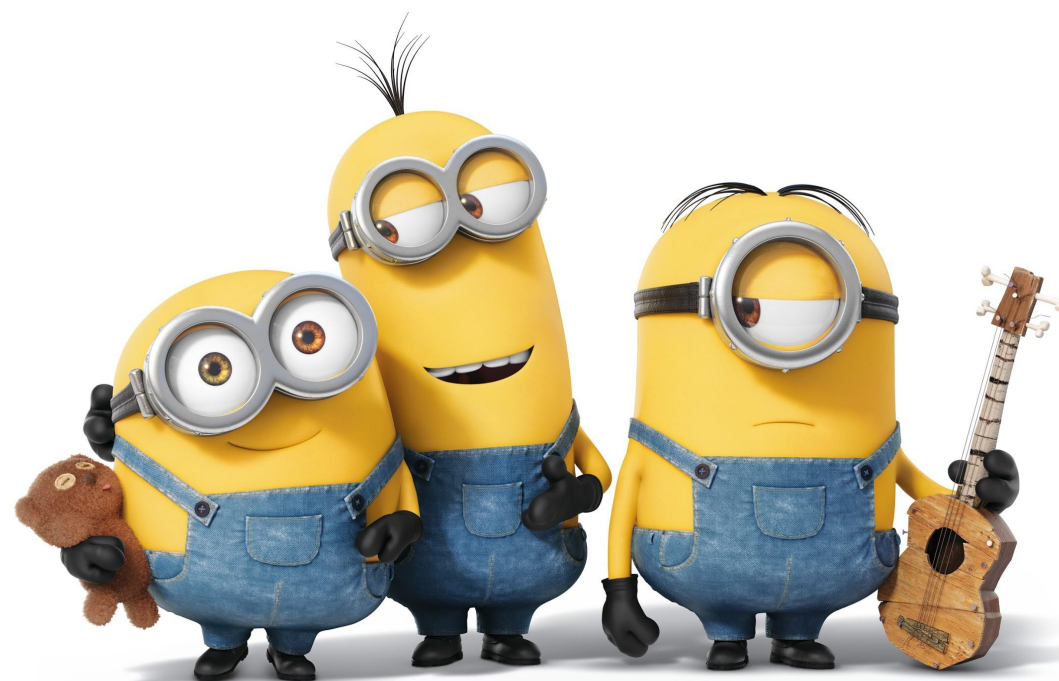
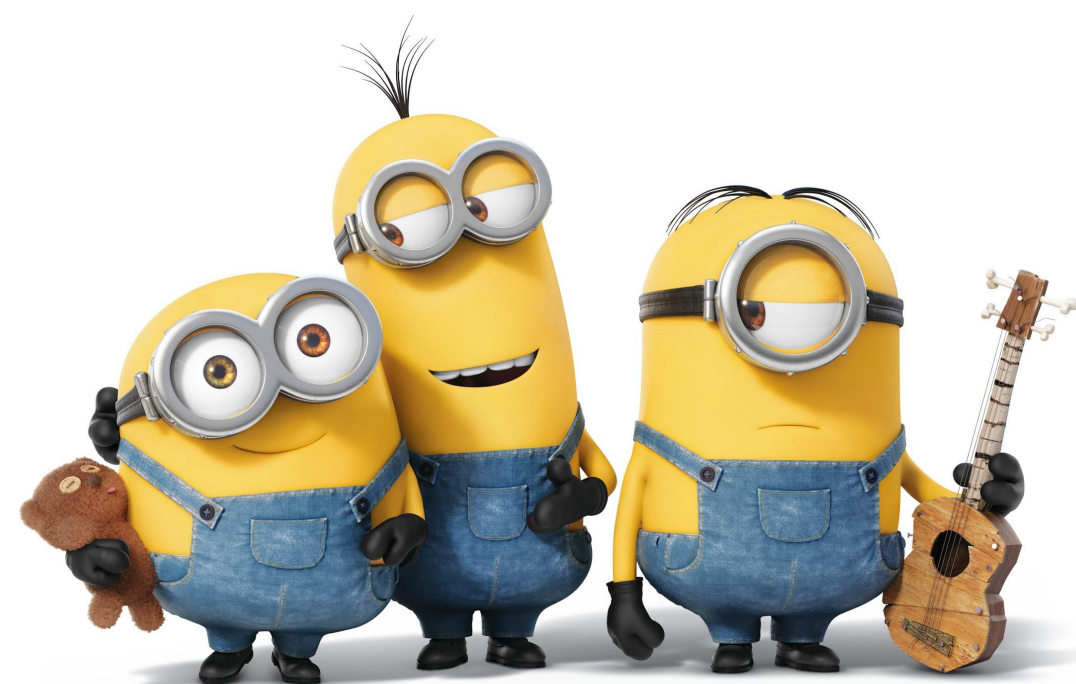
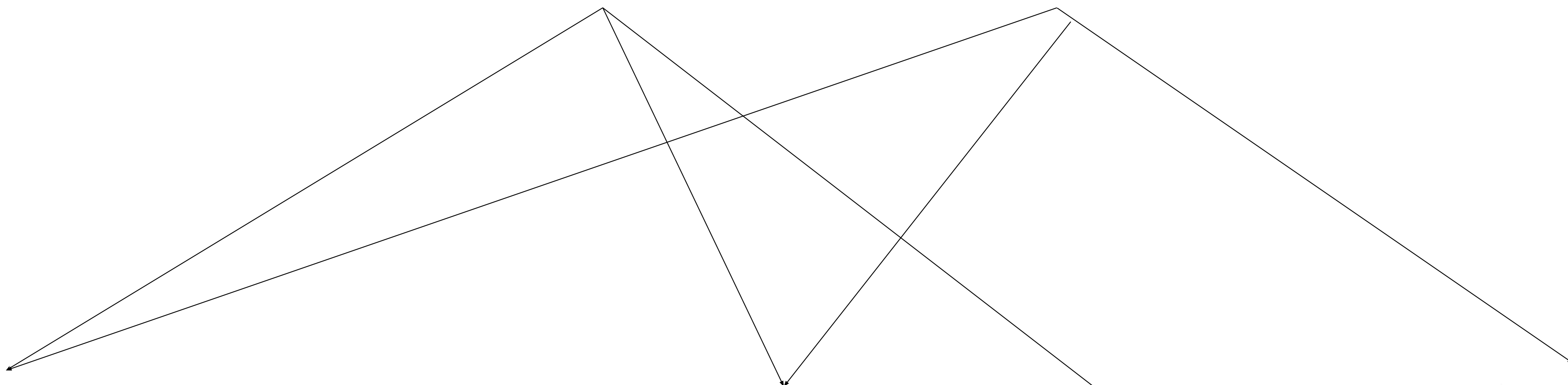
Salt == SaltStack

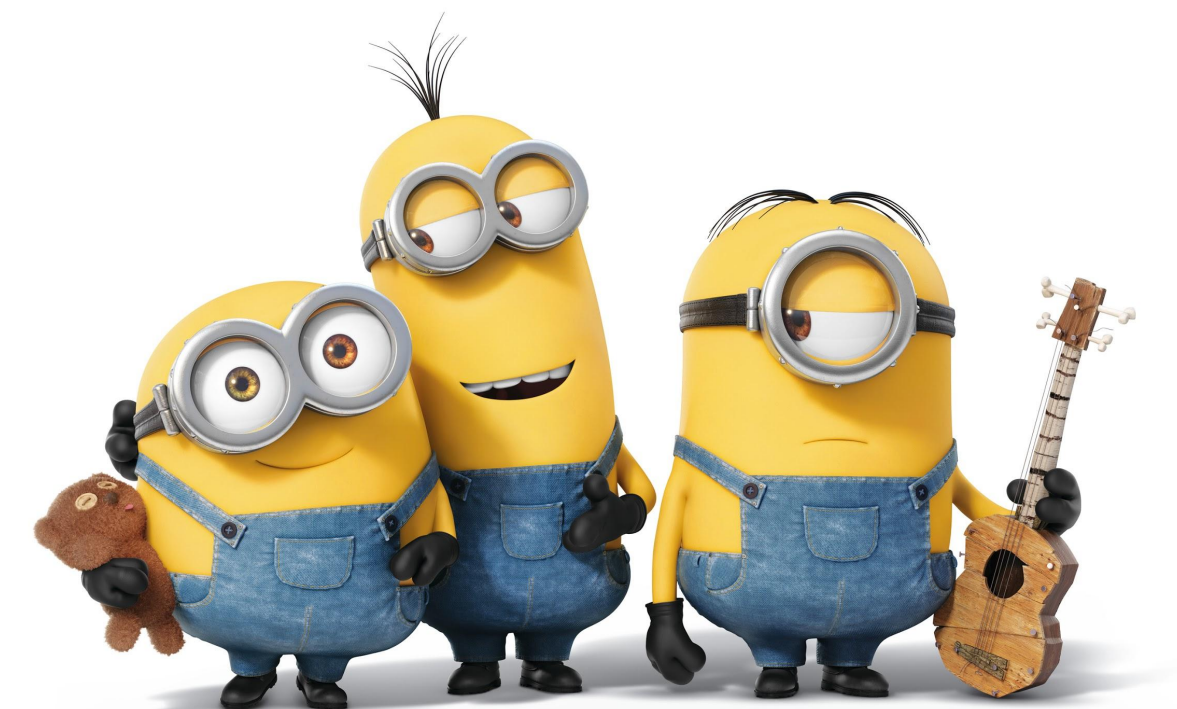
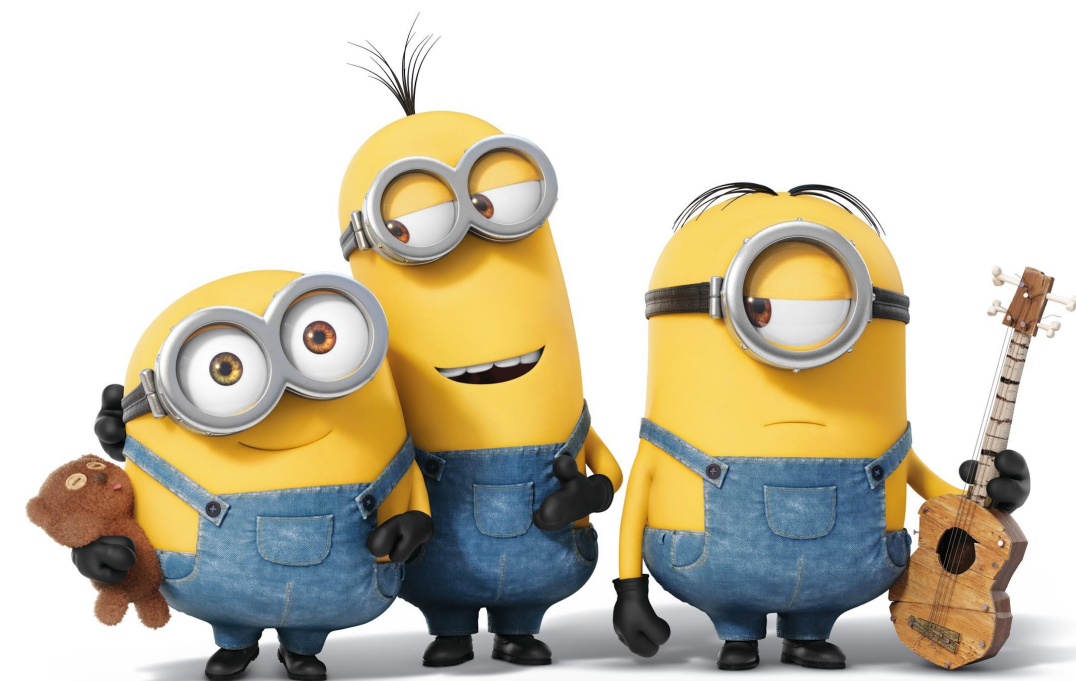
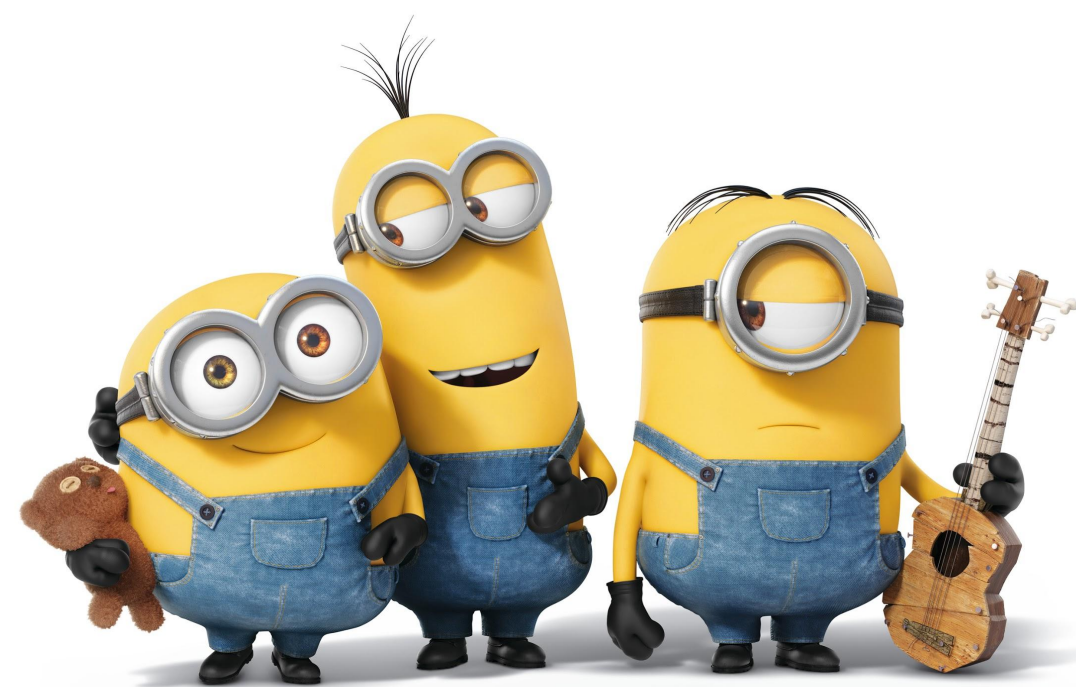
Remote Execution Configuration Management

**What is SaltStack?
Event Driven Orchestration**

DevOps Platform







Open Source Summit North America 2018

 **@garethgreenaway**

SaltStack Loader

Standard Modules

Virtual Modules

Lazy Loader

Types of Modules

Remote Execution Modules

The heavy lifters of SaltStack.

Typically calling out to other programs and services.

Used by other modules.

Examples:

- file
- cmd
- docker
- pkg (aptpkg, yumpkg, freebsd_pkg, mac_pkgutil)

State Modules

Used for configuration management.

Examples:

- pkg
- x509
- schedule
- file
- apache_module


```
gen_csr:
  x509.csr_managed:
    - name: /etc/pki/salt/{{ grains.id }}.csr
    - private_key: /etc/pki/salt/{{ grains.id }}.key
    - private_key_passphrase: {{ private_pass }}
    - CN: {{ grains.host }}.{{ domain }}
    - C: US
    - ST: California
    - L: San Jose
    - O: MDSLAb.eBay.com
    - OU: IT Department
    - basicConstraints: 'CA:FALSE'
    - keyUsage: 'nonRepudiation, digitalSignature, keyEncipherment'
    - 1.3.6.1.4.1.311.20.2: 'ASN1:UTF8String:eBayComputerCES'
    - ext_mapping:
        '1.3.6.1.4.1.311.20.2': 'DomainController'
    - require:
        - x509: gen_private_key
```

Grain Modules

Used to report information about machines running SaltStack.

Populated or updated on server start or minion software start.

Updated periodically.

Similar to facts from Facter, used by Puppet & Ansible or Chef Ohai attributes.

Examples: Salt version, Python version, operating system.

Runner Modules

Execution modules run on minions, runners run on the master calling out to the minions.

Examples:

- Jobs runner
- State runner
- Event runner

Returners Modules

Gives a minion a way to return data to something besides the master.

Examples:

- MySQL
- Slack
- SMS
- SMTP

Pillar

Centralized system to define configuration.

Examples:

- consul
- etcd
- postgres
- mysql
- mongodb
- s3
- gpg

Outputter modules

Determine the output from the command line interfaces

Examples:

- json
- yaml
- progress

Cloud modules

Used to manage virtual machines on various cloud providers.

Examples:

- Amazon EC2
- Google Cloud
- Azure
- Digital Ocean
- VMWare
- Openstack

Beacons

Allows various software, both Salt components and external applications to report data to Salt.

Examples:

- **diskusage**
- **inotify**
- **wtmp/btmp**
- **ps**

Engines

Allow Salt to provide internal information and services to long-running external processes.

Examples:

- Docker
- Slack
- Logstash
- SQS Events

Agenda

1. Tools & Best Practices
2. Briefly look at the Salt loader
3. Custom Salt Modules
4. ~~Updating~~ Replacing an existing beacon module
5. Custom Salt engines

Tools

Editor Bits

- <https://github.com/saltstack/salt-vim>
- <https://github.com/saltstack/sublime-text>
- Emacs salt-mode
- <https://github.com/w0rp/ale>
- <https://github.com/Valloric/YouCompleteMe>

Testing

- <https://github.com/cachedout/barnacle>
- <https://github.com/saltstack/kitchen-salt>
- PyENV

Documentation

- <https://docs.saltstack.com/en/latest/>
- Salt sys.doc, state_doc, runner_doc, etc.

```
local:
```

```
-----
```

```
sys.doc:
```

Return the docstrings for all modules. Optionally, specify a module or a function to narrow the selection.

The strings are aggregated into a single document on the master for easy reading.

Multiple modules/functions can be specified.

CLI Example:

```
salt '*' sys.doc
salt '*' sys.doc sys
salt '*' sys.doc sys.doc
salt '*' sys.doc network.traceroute user.info
```

Modules can be specified as globs.

New in version 2015.5.0

```
salt '*' sys.doc 'sys.*'
salt '*' sys.doc 'sys.list_*
```


Debugging

- PyCharm Professional Edition
- PyDev (<http://www.pydev.org>)
- pdb
- pudb
- ptvsd: Visual Studio Remote Debugging Server

SaltConf 17
Megan Wilhite
<https://bit.ly/2KXuJUB>

salt-extend
<https://bit.ly/2Gd0E6z>

SaltStack Development Basics

The `__virtual__` function

- Not required, but definitely best practice.
- Executed when Salt minion starts up.
- Optionally provides identification for the name of the module.
- Commonly used to identify whether the module should be loaded and made available.



- `__virtual_aliases__` allows you to provide multiple names

Commonly Used Dunder Variables

`_opts_`

__salt__

`__grains__`

__pillar__

__utils__

Docstrings

```
'''
```

```
    My function in my Salt module
```

```
CLI Example:
```

```
.. code-block:: bash
```

```
    salt '*' really_cool.function argument keyword=value
```

```
'''
```

Private Functions

- Name starts with an underscore.
- Only used internally by the module.
- Will not show up in results from sys.doc and friends

Public Functions

- Called directly by Salt commands
- Will show up in the results returned from `sys.doc` and friends, such as `list_functions`.

Logging

```
import logging
```

```
log = logging.getLogger(__name__)
```

```
log.info('Useful for users')
```

```
log.warn('Something went wrong but I am still good to go.')
```

```
log.error('Something has gone wrong and should be fixed.')
```

```
log.debug('Useful information for developers.')
```

Python2/Python3 Compatibility

```
from __future__ import absolute_import, unicode_literals, print_function
```

Custom Modules

Why?

**SaltConf 17
Module Counts**



beacons: 27
clouds: 34
engines: 17
grains: 21
log handlers: 5
output: 17
modules: 481
proxy: 18
renderers: 20
returners: 39
sdb: 14
states: 317
utils: 129

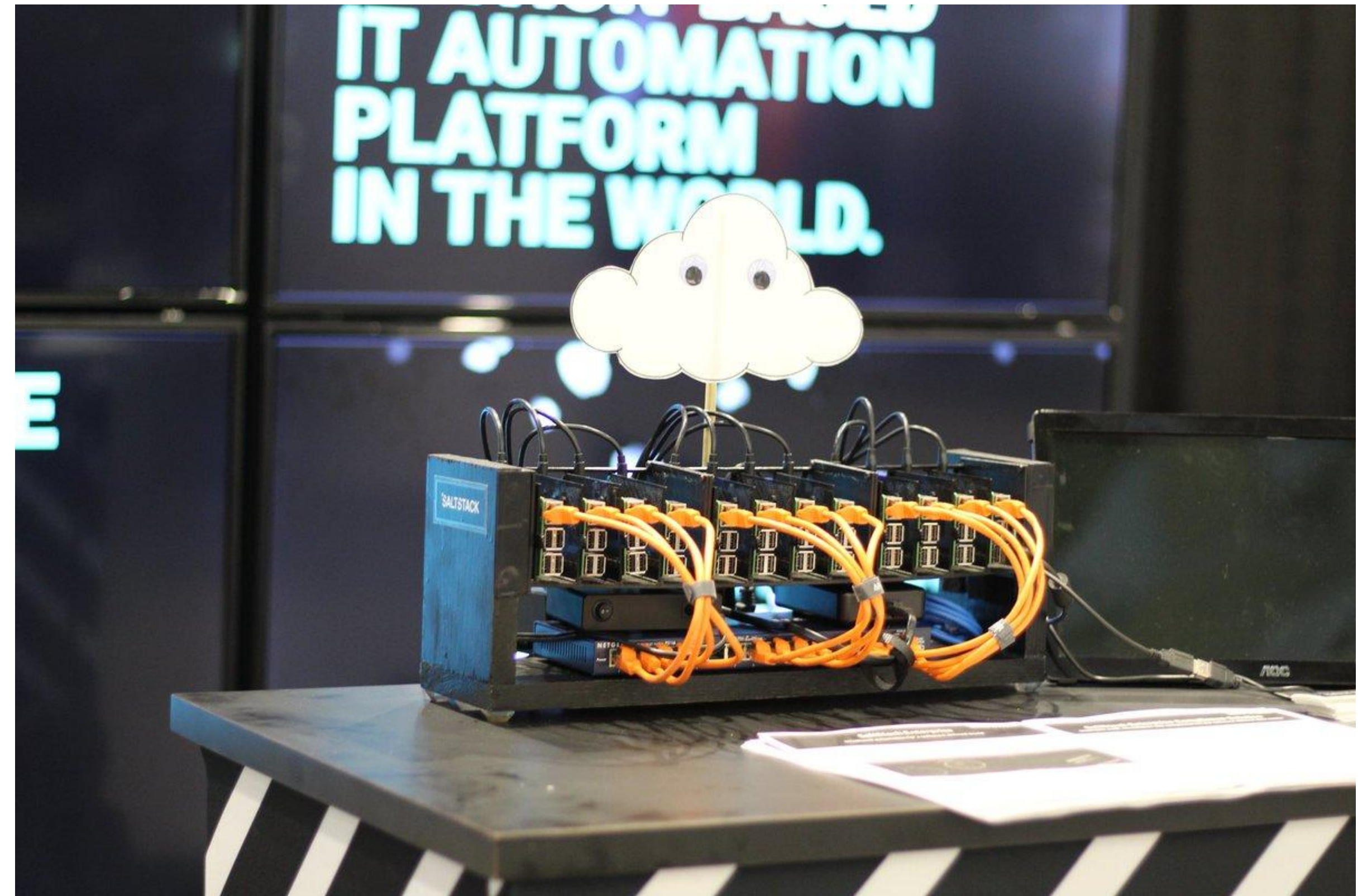
Current Module Counts

beacons: 28
cloud: 34
engines: 18
grains: 25
log handlers: 5
output: 17
modules: 497
proxy: 19
renderers: 21
returners: 40
sdb: 14
states: 338
utils: 135

So why?

Let's Look At Some Code!

All Code Examples
<https://bit.ly/2rLheVH>



Questions?

Thanks for coming!
