

Ops Challenge - Slack

rCAT

Rudimentary configuration management tool

Satishkumar Dhule  
12-4-2021

## Contents

Context .....	2
Technology Stack .....	2
Features .....	2
Flow diagram.....	3
How to Run .....	4
A. Sample Run #1: Creating resources from scratch .....	5
B. Sample Run #2: Delete resources .....	6
C. Sample Run #3: Showing idempotent nature of execution .....	7
UML diagram.....	3

## Context

- Construct a rudimentary configuration management tool and use it to configure two servers for production service of a simple PHP web application.
- You are not allowed to use off-the-shelf tools like (but not limited to) Puppet, Chef, Fabric, or Ansible.
- Instead, we would like you to implement a tool a bit like Puppet or Chef that meets the following specifications and then use that tool to configure the two servers.
- Tasks
- If your tool has dependencies not available on a standard Ubuntu instance you may include a bootstrap.sh program to resolve them
- Your tool must provide an abstraction that allows specifying a file's content and metadata (owner, group, mode)
- Your tool must provide an abstraction that allows installing and removing Debian packages
- Your tool must provide some mechanism for restarting a service when relevant files or packages are updated
- Your tool must be idempotent - it must be safe to apply your configuration over and over again
- Don't forget to document the basic architecture of your tool, how to install it, how to write configurations, and how to invoke them
- Your configuration must specify a web server capable of running the PHP application below
- Both servers must respond 200 OK and include the string "Hello, world!" in their response to requests from curl -sv http://(using the public IP address)

Here is the PHP application in question

```
<?php
header("Content-Type: text/plain");

echo "Hello, world!\n";
?>
```

## Technology Stack

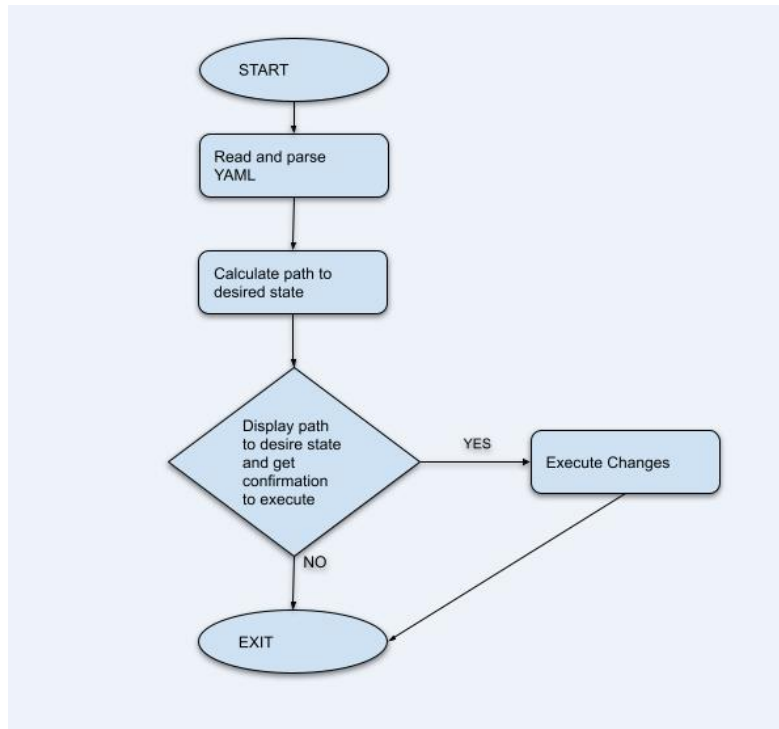
Python 3 and core modules

## Features

- abstraction that allows specifying a file's content and metadata (owner, group, mode)
- abstraction that allows installing and removing Debian packages
- mechanism for restarting a service when relevant files or packages are updated
- Idempotent, safe to apply your configuration over and over again

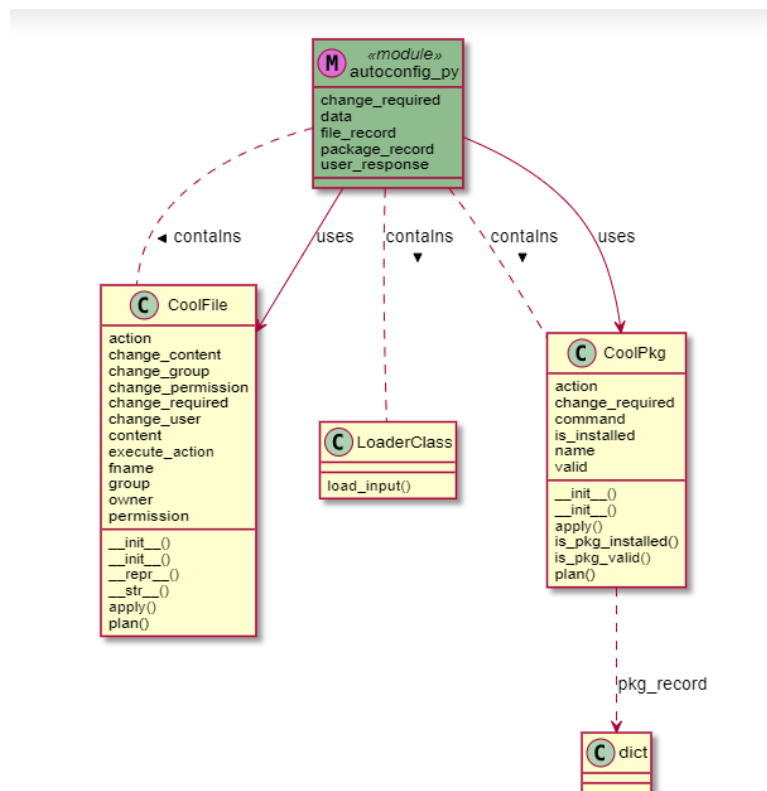
## Flow diagram

Below is the flow diagram how we are planning to execute this.



## UML diagram

- Please find the below UML for reference



## How to Run

Copy python script from repo and create yaml file by referring below sample

Packages node is used to define package manipulations

- name : Name of apt package
- action : add/remove/install/uninstall
- command : semicolon separated commands to be executed after installation

File node is used to define file manipulations

- name : Name of apt file
- action : create/delete
- permission : specify permission for file
- group : desired group assignment
- owner : desired owner assignment
- content : desired content of file
- type : Not implemented (reserved for file/directory)

```
#testy.yml
packages:
- name: php
  action: add
- name: apache2
  action: add
  command: sudo a2dismod mpm_event;sudo a2enmod php*;sudo /etc/init.d/apache2 restart

files:
- name: /var/www/html/index.html
  type: file
  action: delete
- name: /var/www/html/index.php
  type: file
  action: create
  permission: 0644
  group: root
  owner: root
  content: |
    <?php
    header("Content-Type: text/plain");
    echo "Hello, world!\n";
    ?>
```

## A. Sample Run #1: Creating resources from scratch

- Using above yaml we are installing apache2 and php packages
- Execution shows what changes are expected and asks for confirmation
- If user aborts no changes will be made to system
- After users' confirmation, you can see packages are installed and files are manipulated
- Additionally, you can see browser's screenshot attached which shows, we are able to access php site using public IP

```
root@ip-172-31-255-68:~/script# python3 autoconfig.py
Plan for pkg add:php -->
- Apply add php

Plan for pkg add:apache2 -->
- Apply add apache2
- Apply command sudo a2dismod mpm_event;sudo a2enmod php7.4;sudo /etc/init.d/apache2 restart

Plan for file delete:/var/www/html/index.html -->
- No changes required for /var/www/html/index.html

Plan for file create:/var/www/html/index.php -->
- Apply create /var/www/html/index.php
- Apply change content
- Apply change permissions: 0o644
- Apply change group: root
- Apply change owner: root

Do you want to execute this plan: [Yes|No]
Yes
Installing php ... :
SUCCESS

deleteing /var/www/html/index.html ... : SUCCESS
createing /var/www/html/index.php and/or adding content... : SUCCESS
Changing permission for /var/www/html/index.php ... : SUCCESS
Changing group/owner for /var/www/html/index.php ... : SUCCESS
root@ip-172-31-255-68:~/script# curl -sv http://localhost
* Rebuilt URL to: http://localhost/
* Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET / HTTP/1.1
> Host: localhost
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 04 Dec 2021 05:47:01 GMT
< Server: Apache/2.4.29 (Ubuntu)
< Content-Length: 14
< Content-Type: text/plain; charset=UTF-8
<
Hello, world!
* Connection #0 to host localhost left intact
```

Invocation and Plan . Confirmation

List of commands which will be executed

Installation after confirmation

Verification that we are able to access the URL

Not secure | 54.86.99.156

Hello, world!

Verification for Public IP

## B. Sample Run #2: Delete resources

- testy.yml in below screenshot shows we are removing packages
- Execution shows what changes are expected and asks for confirmation
- If user aborts no changes will be made to system
- After users' confirmation, you can see packages are removed
- Additionally, you can see browser's screenshot attached which shows, we are NOT able to access php site using public IP

```
root@ip-172-31-255-68:~/script# cat testy.yml
packages:
  - name: php
    action: remove
  - name: apache2
    action: remove
  command: sudo a2dismod mpm_event;sudo a2enmod php7.4;sudo /etc/init.d/apache2 restart

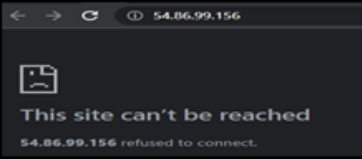
files:
  - name: /var/www/html/Index.html
    type: file
    action: delete
  - name: /var/www/html/index.php
    type: file
    action: remove
    permission: 0644
    group: root
    owner: root
    content: |
      <?php
      header("Content-Type: text/plain");
      echo "Hello, world!\n";
      ?>

root@ip-172-31-255-68:~/script# python3 autoconfig.py
Plan for pkg remove:php -->
- Apply remove php
Plan for pkg remove:apache2 -->
- Apply remove apache2
Plan for file delete:/var/www/html/index.html -->
- No changes required for /var/www/html/index.html
Plan for file remove:/var/www/html/index.php -->
- No changes required for /var/www/html/index.php
Do you want to execute this plan: [Yes|No]
Yes
removeing php... : SUCCESS
removeing apache2... : SUCCESS

root@ip-172-31-255-68:~/script# curl -sv http://localhost
* Rebuilt URL to: http://localhost/
* Trying 127.0.0.1...
* TCP_NODELAY set
* connect to 127.0.0.1 port 80 failed: Connection refused
* Failed to connect to localhost port 80: Connection refused
* Closing connection 0
root@ip-172-31-255-68:~/script#
```

Removing Packages by editing testy.yml

Plan and get confirmation



Page has stopped working after we done uninstallation

### C. Sample Run #3: Showing idempotent nature of execution

- In below screenshot, we are trying to apply same configuration twice
- In second attempt program detects that there are no changes required as configuration is matching with the installation

```
root@ip-172-31-255-68:~/script# python3 autoconfig.py
Plan for pkg remove:php -->
- Apply remove php

Plan for pkg remove:apache2 -->
- Apply remove apache2

Plan for file delete:/var/www/html/index.html -->
- No changes required for /var/www/html/index.html

Plan for file delete:/var/www/html/index.php -->
- Apply delete /var/www/html/index.php

Do you want to execute this plan: [Yes|No]

Yes
removeing php... : SUCCESS

removeing apache2... : SUCCESS

deleteing /var/www/html/index.php ... : SUCCESS
root@ip-172-31-255-68:~/script# python3 autoconfig.py
Plan for pkg remove:php -->
- No action required

Plan for pkg remove:apache2 -->
- No action required

Plan for file delete:/var/www/html/index.html -->
- No changes required for /var/www/html/index.html

Plan for file delete:/var/www/html/index.php -->
- No changes required for /var/www/html/index.php
```



## Conclusion

- This tool seems to be best fit for deploying tiny applications with low deployment complexity
- There is scope of improvements, some of them are given below
  - a. Keeping track of changes which took place in past
  - b. Roll back/forward feature
  - c. Remote execution