



c0mbat - Product Documentation

Release 3.7a

Weqaar Janjua

Jan 09, 2019

CONTENTS

1	Introduction	1
1.1	General Introduction	1
1.2	Product	1
1.3	Availability of code	2
1.4	Architecture / Code Walk-through	2
1.4.1	Structure	2
1.4.2	Code re-usability	3
1.5	5. Usage	3
2	Support	5

INTRODUCTION

1.1 General Introduction

c0mbat is a Configuration Management tool developed with [Slack](#).

c0mbat is short for Zero-Configuration Management and Build [Automorphic](#) Tool.

1.2 Product

c0mbat is intended for configuration management in Cloud Infrastructures.

- Features

1. Multi-threaded (uses multiple cores as scheduled by the OS or by user i.e. numa utils)
2. Parallel deployment currently over SSH channel
3. Application wide configuration file: c0mbat.conf
4. APT package manager handling: install, uninstall (remove), update
5. Services handling: start, stop, restart, reload
6. Inventory and Artifacts handled separately under dir “deploy”
 - (a) deploy/artifacts
 - (b) deploy/inventory
7. Idempotency with deployments: keeps MD5 Hashes as deployment cache for each host and it's artifact relationships
 - (a) Filesystem hash of individual artifacts under deploy/artifacts
 - (b) Configuration hash of individual artifacts, defined in deploy/artifacts/artifacts.json
 - (c) Hashes are maintained as cache under “cache/cache.json” that is built after initial run
 - (d) In ideal situation this should be a distributed cache accessible over IP i.e. NoSQL DB, to allow multiple client's to sync
8. Two types of deployments currently supported
 - (a) APT package manager based
 - (b) Manual
9. Manual type deployments support Permissions, this is specified optionally as artifact's metadata (see deploy/artifacts/artifacts.json)

10. Logging under dir “logs/”
11. Cleanup runtime/transient data with the script “clean.sh”. Note this cleans up the logs and cache as well.

1.3 Availability of code

Available on Github at [c0mbat repo](#)

1.4 Architecture / Code Walk-through

1.4.1 Structure

Code is organized in this hierarchy

```
├── docs
│   ├── build
│   ├── source
│   └── Makefile
├── src
│   ├── cache
│   ├── deploy
│   ├── logs
│   ├── packages
│   ├── runtime
│   ├── sh
│   ├── c0mbat.conf
│   ├── c0mbat.py
│   └── clean.sh
├── bootstrap.sh
├── README.txt
└── requirements.txt
```

Code is functionally divided as packages

```
src/packages
├── artifacts
│   ├── initartificats.py
│   └── __init__.py
├── conf
│   ├── configinit.py
│   ├── __init__.py
│   └── sysconfigx.py
├── inventory
│   ├── initinventory.py
│   └── __init__.py
├── queue
│   ├── __init__.py
│   └── initqueues.py
├── remoteaccess
│   ├── __init__.py
│   └── ssh.py
├── runtime
│   ├── cache.py
│   ├── __init__.py
│   └── sysinit.py
├── threads
│   ├── deploymentworker.py
│   ├── __init__.py
├── globalvars.py
└── __init__.py
```

1.4.2 Code re-usability

Source code allows for easy reuse of various components

1. All variables and objects are initialized as run-time from “packages/conf/configinit.py”
2. Initialized objects are available at run-time in “packages/globalvars.py”
3. Re-use is as simple as calling the desired object with `globalvars.<object>`
 - (a) Example: to log a debug message use - `globalvars._stats_logger.debug(“YOUR MESSAGE!”)`

1.5 5. Usage

1. Installation instructions [README.txt](#)
2. c0mbat provides a CLI Interface, run with: `python c0mbat.py -h`

CHAPTER
TWO

SUPPORT

Note: Contact weqaar.janjua@gmail.com
