



Introduction to CFEngine Build
Lars Erik Wik, Ole Herman Schumacher Elgesem

- Background
- How it works
- Website: build.cfengine.com
- Command line interface: cfb
- Module examples
- Demos
- Writing modules
- Contributing

Background

CFEngine Build attempts to improve all these 3 needs from our users:

- An easier getting started experience, with more value “out of the box” (without learning and writing a lot of code)
- An easier way to upgrade (currently it is quite common to maintain your own policy set fork / patches)
- A place (website) with more examples, snippets, ready to use policy / modules

10 years ago - Right idea, limited implementation.

No website.

CFEngine had limited JSON support, git integration, no CMDB.

Less “things” to put in modules (now we have custom promise types, compliance reports, ...).



How it works

01 Explore CFEngine Build and add modules



03 Deploy policy set to hub



02 Build policy set



04 Observe results in reports and Web UI



We create a `cfbs.json` file in the current folder and run `git init` (by default).

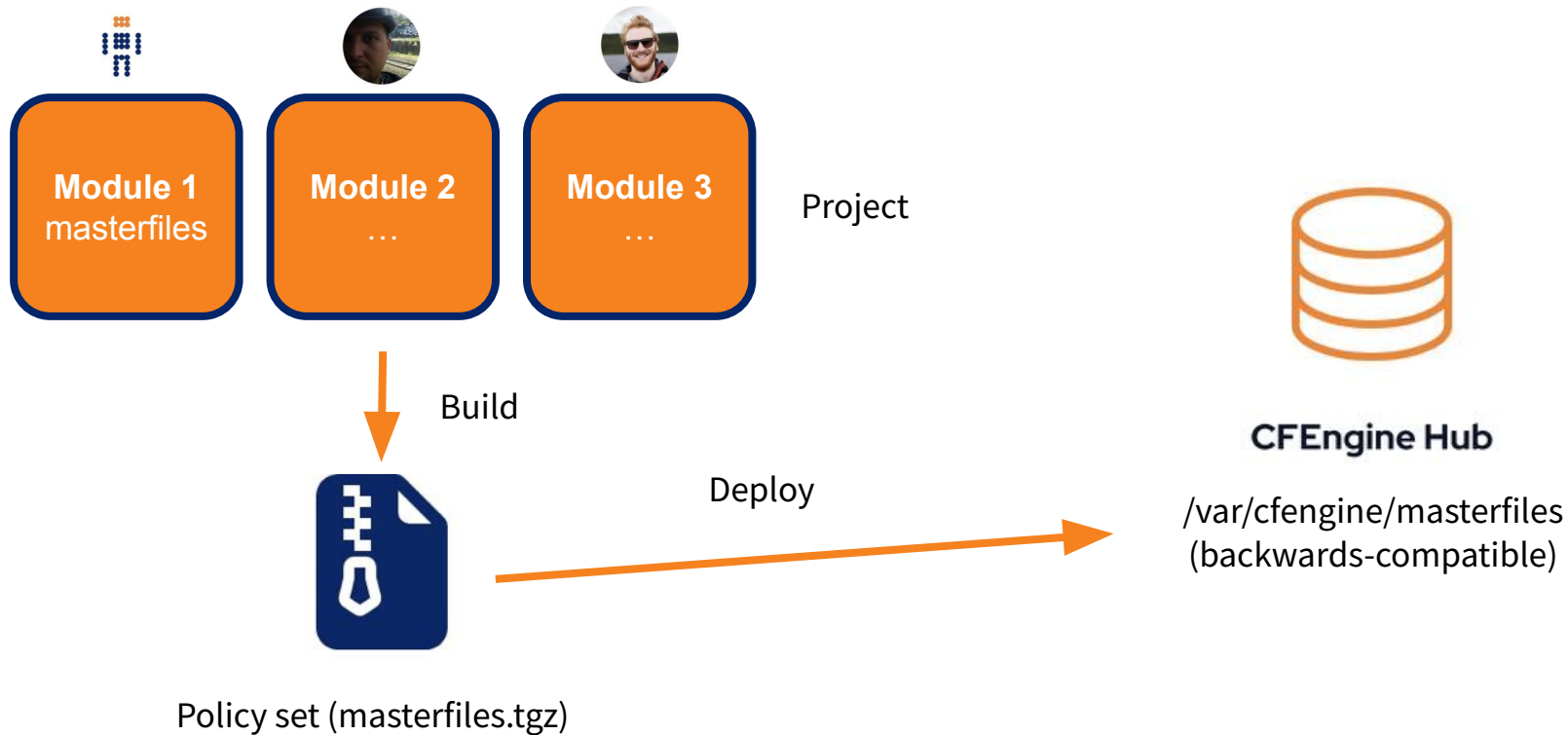
We call this repo/folder a CFEngine Build **project**.

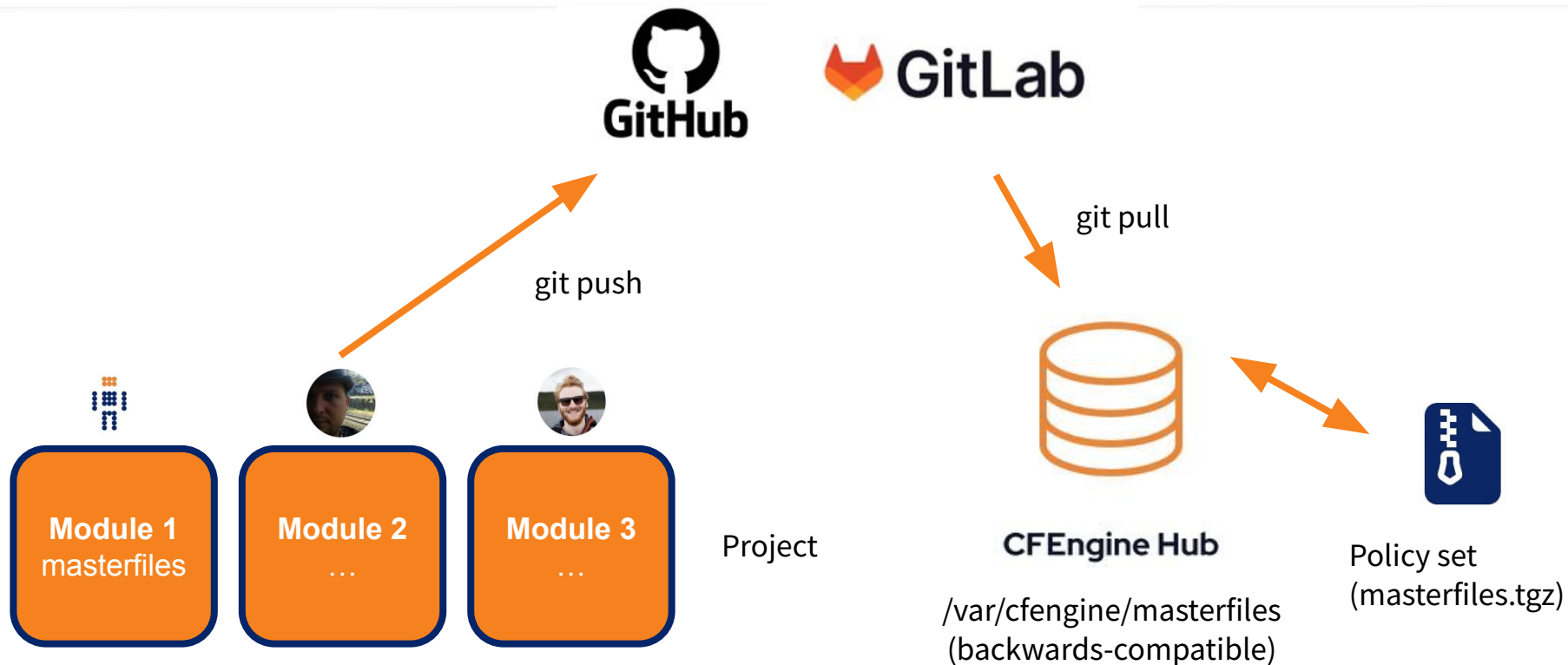
A **project** consists of multiple **modules** and some metadata.

When you build a project (`cfbs build`) you turn it into a **policy set**.

The **policy set** is what you **deploy** on your hub(s), in `/var/cfengine/masterfiles`.

This **policy set** is backwards compatible, to your hub and hosts it looks the same as before when you were not using CFEngine Build.







Dashboard



Hosts



Reports



Measurements



Policy analyzer



Build

CFEngine Build

You have not set up a remote repository for your "Local project". The changes you make will not be synchronized or backed up anywhere.

[Set repository](#)Project [Edit](#)

Local project

Added modules

masterfiles

cron-access

etc-motd-access

compliance-report-imp...

autoun

compliance-report-os-is...

Search modules



Deploy locally



cron-access

Limits access to cron-related files in /etc by setting user, group, and permission bits.

[Remove module](#)

Details

Downloads 9
Updated 10 Dec 2022
Version 0.0.1
[Repository](#) [Report issue](#)

Input data

This module does not support input data

[Edit](#)

Description

The `crontab` utility enables running commands or programs periodically. System-wide `crontab`-related configuration files exist in the `/etc` directory. For these files, enforcing strict access is essential, as editing them allows you to run commands as any user, including `root`.

Recommendation: Limit access to the various `crontab`-related files and directories in `/etc`. All of them should be owned by

Choice of technology

cfbs (and cf-remote) are written in Python

build.cfengine.com is built on Hugo (HTML templates, CSS, JS)

GitHub Actions for CI on PRs and uploading when merged

JSON with modules (index) and versions is in GitHub

zip folders of all versions of modules are stored in AWS s3

- Snapshotted when author's PR to build-index is merged
- Module will still be available, even if author removes repo

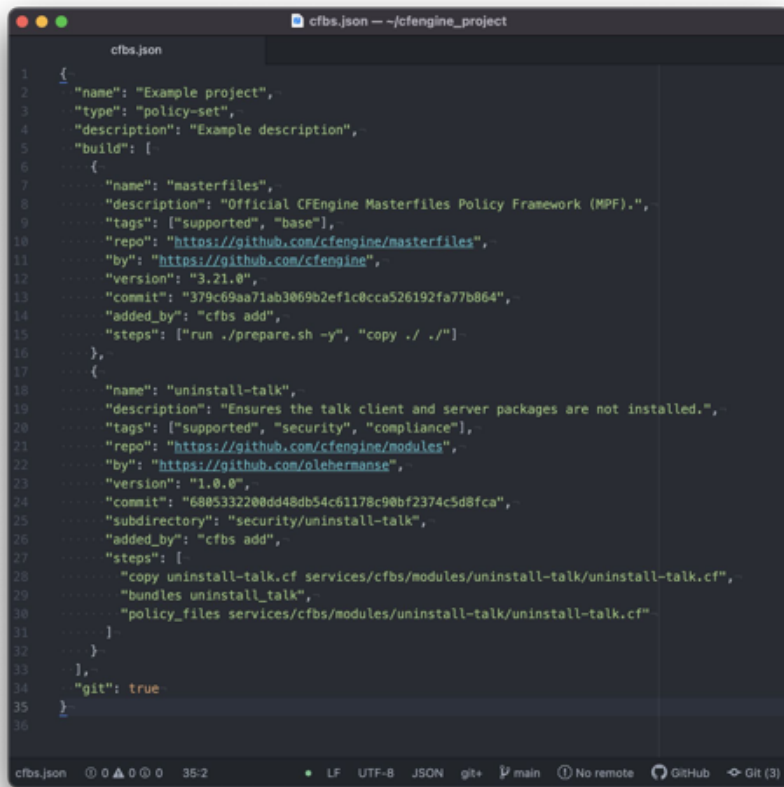
It should not matter when or where you **build** a project.

For the same project the resulting build should be the same.

You build it locally today, your colleague builds it on their machine tomorrow, and on monday your CFEngine Hub builds it and deploys it.

When you run cfbs, the command line tool, it looks in the current working directory for a cfbs.json file with:

- Metadata (name, description)
- Configuration for cfbs (git, ...)
- **Modules** (what to build)



```
1 {
2   "name": "Example project",
3   "type": "policy-set",
4   "description": "Example description",
5   "build": [
6     {
7       "name": "masterfiles",
8       "description": "Official CFEngine Masterfiles Policy Framework (MPF).",
9       "tags": ["supported", "base"],
10      "repo": "https://github.com/cfengine/masterfiles",
11      "by": "https://github.com/cfengine",
12      "version": "3.21.0",
13      "commit": "379c69aa71ab3069b2ef1c8cca526192fa77b864",
14      "added_by": "cfbs add",
15      "steps": ["run ./prepare.sh -y", "copy ./ ./"]
16    },
17    {
18      "name": "uninstall-talk",
19      "description": "Ensures the talk client and server packages are not installed.",
20      "tags": ["supported", "security", "compliance"],
21      "repo": "https://github.com/cfengine/modules",
22      "by": "https://github.com/olehermanse",
23      "version": "1.0.0",
24      "commit": "6805332200dd48db54c61178c90bf2374c5d8fca",
25      "subdirectory": "security/uninstall-talk",
26      "added_by": "cfbs add",
27      "steps": [
28        "copy uninstall-talk.cf services/cfbs/modules/uninstall-talk/uninstall-talk.cf",
29        "bundles uninstall_talk",
30        "policy_files services/cfbs/modules/uninstall-talk/uninstall-talk.cf"
31      ]
32    }
33  ],
34  "git": true
35 }
```



```
17 {
18   "name": "uninstall-talk",
19   "description": "Ensures the talk client and server packages are not installed.",
20   "tags": ["supported", "security", "compliance"],
21   "repo": "https://github.com/cfengine/modules",
22   "by": "https://github.com/olehermanse",
23   "version": "1.0.0",
24   "commit": "6805332200dd48db54c61178c90bf2374c5d8fca",
25   "subdirectory": "security/uninstall-talk",
26   "added_by": "cfbs add",
27   "steps": [
28     "copy uninstall-talk.cf services/cfbs/modules/uninstall-talk/uninstall-talk.cf",
29     "bundles uninstall_talk",
30     "policy_files services/cfbs/modules/uninstall-talk/uninstall-talk.cf"
31   ]
32 }
```



Website

Welcome to CFEngine Build

CFEngine Build is a catalogue of policy and modules created by CFEngine, our partner and community that helps DevSecOps teams to simplify the automation process. With CFEngine Build you can automate more with less effort.

[Getting started >](#)

[Find modules >](#)

Learn how to [contribute](#) modules to CFEngine Build and help others in the community



Featured modules

Modules recommended by CFEngine team



inventory-openssl-versions

by Ole Herman Schumacher Eigesem

Adds an inventory attribute containing all versions of OpenSSL found on the system.

0.2.0



uninstall-ftp

by Craig Comstock

supported

Ensures the ftp server package is not installed on the system(s).

0.0.2



promise-type-http

by Vlastislav Podzimek

supported

Promise type to perform HTTP(S) requests from policy.

1.1.0

Popular modules

Modules with the most downloads



masterfiles

by CFEngine

supported

Official CFEngine Masterfiles Policy Framework (MPF).

3.21.0



autorun

by Ole Herman Schumacher Eigesem

supported

Enables autorun functionality.

1.0.1



library-for-promise-types-in-python

by CFEngine

supported

Library enabling promise types implemented in python.

0.1.1



Found 7 results

SORT BY Alphabetically ▾



[inventory-ssh-host-key-fingerprints](#)

by Nick Anderson

Version:0.0.2

Updated: Jan 24, 2023

Total downloads: 505

Adds reporting data (inventory) for the SSH host key fingerprints.

inventory security ssh experimental



[library-sshd-config](#)

by Nick Anderson

Version:0.1.0

Updated: Dec 3, 2021

Total downloads: 518

Library used by other modules to manage sshd configuration.

library security ssh experimental



[ssh-ciphers-strong](#)

by Nick Anderson

Version:1.0.4

Updated: Dec 3, 2021

Total downloads: 512

Ensures that the SSH daemon uses strong ciphers.

security ssh experimental

Tags

supported

Modules that are supported and tested by CFEngine.

supported

management

inventory

security

experimental

reporting

cfengine-enterprise

compliance

compliance-report

cve

sudo

kernel

ssh

systemd

library

base

promise-type

http

lib

tmp

uninstall-telnet-server

Ensures the telnet server package is not installed on the system.

[Repository](#) [Report issue](#)

Installation

```
cfbs add uninstall-telnet-server
```



Description

Dependencies

Comments

Telnet is a protocol for providing an **insecure** remote virtual terminal access to hosts over network. It is generally considered obsolete and dangerous and its use has been a root cause of security issues. It should not be used except for very specific use cases.

This module makes sure that the Telnet **server** packages are not installed on hosts and thus also not running.

Maintainer



Vratislav Podzimek

Module stats

📄 Total Downloads: 5

🕒 Updated: Nov 21, 2022

Installation version

Version

0.0.3 

Released on Nov 21, 2022

Tags

supported

security

compliance

Description	Dependencies	Comments
-------------	--------------	----------

Telnet is a protocol for providing an **insecure** remote virtual terminal access to hosts over network. It is generally considered obsolete and dangerous and its use has been a root cause of security issues. It should not be used except for very specific use cases.

This module makes sure that the Telnet **server** packages are not installed on hosts and thus also not running.

Examples

Example of a *cf-agent* run on a host that has the `telnet-server` package installed:

```
[root@hub]# cf-agent -KI
info: Successfully removed package 'telnet-server'
```

Adding exceptions

If Telnet server packages are really needed on some specific hosts, they can be marked as such by defining the `hardening_telnet_server_allowed` class in either `augments` or `CMDB`.



Command Line Interface: cfbs

Allows you to create, edit, build, and install CFEngine Build projects.

Running `cfbs build` in a project gives you a policy set

The policy set is what you want to deploy;

- Locally with `cfbs install`
- Remotely with `cf-remote deploy`
- Or set up your policy server to pull with `git`

Important commands

cfbs init - Create new project

cfbs add - Add module to project

cfbs build - Build project into policy set

cfbs install - Install policy set (locally)

Once you have a project, there are multiple options for deploying to your hub(s):

- `cfbs build && cfbs install`
- `cfbs build && cf-remote deploy`
- `git push` (set up hub to pull)
- Use GUI / Deploy button in CFEngine Enterprise Mission Portal

Module examples

Base policy set / libraries

masterfiles

surf-cfengine-library

library-sshd-config

Enforce security requirements

uninstall-telnet-server

ssh-protocol-2

delete-files

file-permissions

uninstall-packages

CVEs (Fix / mitigate / investigate)

cve-2021-3156-sudo

cve-2021-44228-log4j

Change settings

every-minute

client-initiated-reporting

Promise types

promise-type-http

promise-type-git

promise-type-systemd

Import reports

compliance-report-lynis

compliance-report-os-is-vendor-supported

Add reporting data (inventory)

inventory-openssl-versions

inventory-sudoers

inventory-unshadowed-users

inventory-lastlog

inventory-etc-hosts

Demos

Creating a project and adding a module

cfbs init

cfbs add uninstall-telnet-server

cfbs build

cf-remote deploy

Create project

```
devbox ~ vagrant@hub: ~/ghent-2023 ~ ssh - vagrant ssh hub - 88x28
vagrant@hub:~/ghent-2023$ cfbs init
Please enter the name of this CFEngine Build project [Example project]: Ghent 2023
Please enter the description of this CFEngine Build project [Example description]: Config Management Camp demo
Do you want cfbs to initialize a git repository and make commits to it? [YES/y/no/n] y
Please enter user name to use for git commits [cfbs]:
Please enter user email to use for git commits [cfbs@hub]:
Initialized empty Git repository in /home/vagrant/ghent-2023/.git/
The default commit message is 'Initialized a new CFEngine Build project' - edit it? [yes/y/NO/n]
Committing using git:

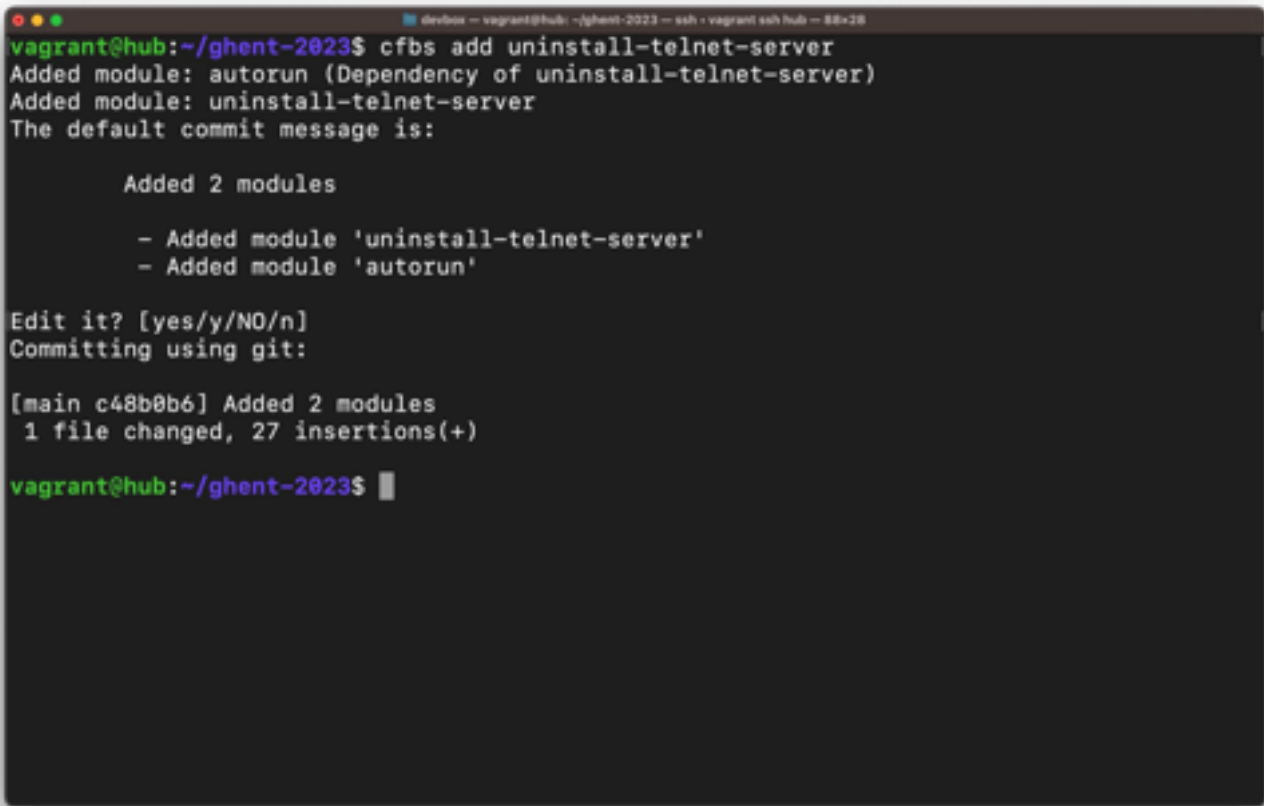
[main (root-commit) ff0286b] Initialized a new CFEngine Build project
 1 file changed, 7 insertions(+)
 create mode 100644 cfbs.json

Initialized an empty project called 'Ghent 2023' in 'cfbs.json'
Do you wish to build on top of the default policy set, masterfiles? (Recommended) [YES/y/no/n]:
Added module: masterfiles
The default commit message is 'Added module 'masterfiles'' - edit it? [yes/y/NO/n]
Committing using git:

[main 33c9ffb] Added module 'masterfiles'
 1 file changed, 13 insertions(+), 1 deletion(-)

vagrant@hub:~/ghent-2023$
```

Add module

A terminal window with a dark background and light text. The window title bar shows 'devbox - vagrant@hub: ~/ghent-2023 - ssh - vagrant ssh hub - 88x28'. The terminal content shows the execution of 'cfbs add uninstall-telnet-server', which adds 'uninstall-telnet-server' and its dependency 'autorun'. It then shows the commit message, a list of added modules, a confirmation to edit, and the final commit output.

```
vagrant@hub:~/ghent-2023$ cfbs add uninstall-telnet-server
Added module: autorun (Dependency of uninstall-telnet-server)
Added module: uninstall-telnet-server
The default commit message is:

    Added 2 modules

    - Added module 'uninstall-telnet-server'
    - Added module 'autorun'

Edit it? [yes/y/NO/n]
Committing using git:

[main c48b0b6] Added 2 modules
1 file changed, 27 insertions(+)

vagrant@hub:~/ghent-2023$
```

Build project

```
devbox - vagrant@hub: ~/ghent-2023 - ssh - vagrant ssh hub - 88x28
vagrant@hub:~/ghent-2023$ cfbs build

Modules:
001 masterfiles           @ 379c69aa71ab3869b2ef1c8cca526192fa77b864 (Downloaded)
002 autorun               @ c3b7329b240cf7ad062a0a64ee8b607af2cb912a (Downloaded)
003 uninstall-telnet-server @ b0c5d6e9f2a9fb5904cb1eb9cd948ee7907969ea (Downloaded)

Steps:
001 masterfiles           : run './prepare.sh -y'
001 masterfiles           : copy './' 'masterfiles/'
002 autorun               : json 'def.json' 'masterfiles/def.json'
003 uninstall-telnet-server : copy './telnet_server_policy.cf' 'masterfiles/services/autorun/telnet_server_policy.cf'

Generating tarball...

Build complete, ready to deploy 🍷
-> Directory: out/masterfiles
-> Tarball:   out/masterfiles.tgz

To install on this machine: sudo cfbs install
To deploy on remote hub(s): cf-remote deploy
vagrant@hub:~/ghent-2023$
```


Deploy project

```
devbox - vagrant@hub: ~/ghent-2023 - ssh - vagrant ssh hub - 88x27
vagrant@hub:~/ghent-2023$ cf-remote deploy
Found saved/spawned hubs: vagrant@localhost
Found cfbs policy set: 'out/masterfiles.tgz'

Deploying to:

vagrant@localhost
OS           : Ubuntu 22
Architecture : x86_64
CFEngine     : 3.22.0a.73a23f06f (Community)
Policy server : None
Binaries     : dpkg, apt

Copying: 'out/masterfiles.tgz' to 'vagrant@localhost'
[localhost]:+ out/masterfiles.tgz -> masterfiles.tgz
Running: 'rm -rf /var/cfengine/masterfiles.delete && mv /var/cfengine/masterfiles /var/cfengine/masterfiles.delete && mv masterfiles /var/cfengine/masterfiles && rm -rf /var/cfengine/masterfiles.delete && cf-agent -Kf update.cf && cf-agent -K'
Policy set successfully deployed to 'vagrant@localhost' 🚀
vagrant@hub:~/ghent-2023$
```

Giving input to delete-files module

cfbs add delete-files

cfbs input delete-files

cfbs build && cf-remote deploy

```
devbox - vagrant@hub: ~/ghent-2023 - ssh - vagrant ssh hub - 88x27
vagrant@hub:~/ghent-2023$ cfbs add delete-files
Added module: delete-files
The added module 'delete-files' accepts user input. Do you want to add it now? [yes/y/NO/n]
The default commit message is 'Added module 'delete-files'' - edit it? [yes/y/NO/n]
Committing using git:

[main eaab683] Added module 'delete-files'
 1 file changed, 41 insertions(+)

vagrant@hub:~/ghent-2023$ cfbs input delete-files
Collecting input for module 'delete-files'
Path to file: /tmp/virus
Why should this file be deleted? [Unknown] Malicious file
Specify another file you want deleted on your hosts? [yes/y/NO/n]
The default commit message is 'Added input for module' - edit it? [yes/y/NO/n]
Committing using git:

[main 5f62913] Added input for module
 1 file changed, 26 insertions(+)
 create mode 100644 delete-files/input.json

vagrant@hub:~/ghent-2023$
```

Policy writing

```
cfbs add ./my_policy.cf
```

```
cfbs build && cf-remote deploy
```

Add local policy file

```
devbox - vagrant@hub: ~/ghent-2023 - ssh - vagrant ssh hub - 88x27
vagrant@hub:~/ghent-2023$ cat << EOF > my_policy.cf
> bundle agent foo
> {
>   reports:
>     "Hello CFEngine!";
> }
> EOF
vagrant@hub:~/ghent-2023$ cfbs add ./my_policy.cf
Which bundle should be evaluated (added to bundle sequence)?
  1. ./my_policy.cf:foo (default)
  2. (None)
[1/2]:
Added module: ./my_policy.cf
The default commit message is 'Added module './my_policy.cf'' - edit it? [yes/y/NO/n]
Committing using git:

[main 401d808] Added module './my_policy.cf'
 2 files changed, 16 insertions(+)
 create mode 100644 my_policy.cf

vagrant@hub:~/ghent-2023$
```

git commits

git log

```
devbox ~ vagrant@hub: ~/jghent-2023 ~ ssh - vagrant ssh hub ~ 88x27
commit 401d808a5e6154e72c1ba4204290e27e1484c107 (HEAD -> main)
Author: cfbs <cfbs@hub>
Date:   Fri Feb 3 22:45:44 2023 +0100

    Added module './my_policy.cf'

commit 5f629130811963586c95514cda48e2da7b9b42e5
Author: cfbs <cfbs@hub>
Date:   Fri Feb 3 22:40:30 2023 +0100

    Added input for module

commit eaab683cc46973e6672df674fb7cf634ebe30ceb
Author: cfbs <cfbs@hub>
Date:   Fri Feb 3 22:38:36 2023 +0100

    Added module 'delete-files'

commit c48b0b63eb0a5162d5d2c6c906b6c346ff937e62
Author: cfbs <cfbs@hub>
Date:   Fri Feb 3 22:23:02 2023 +0100

    Added 2 modules

    - Added module 'uninstall-telnet-server'
    - Added module 'autorun'
```

Writing modules

Writing / contributing modules

Make something which works for you (write code)

Put in GitHub repo, add a README

Create a PR to add your module to the [build index JSON](#);

```
"uninstall-apache": {  
  "description": "Ensures the apache package is not installed.",  
  "tags": ["supported", "security", "compliance"],  
  "repo": "https://github.com/cfengine/modules",  
  "by": "https://github.com/olehermanse",  
  "version": "1.0.0",  
  "commit": "680533220dd48db54c61178c90bf2374c5d8fca",  
  "subdirectory": "security/uninstall-apache",  
  "steps": [  
    "copy uninstall-apache.cf services/cfbs/modules/uninstall-apache/uninstall-apache.cf",  
    "policy_files services/cfbs/modules/uninstall-apache/uninstall-apache.cf",  
    "bundles uninstall_apache"  
  ]  
},
```


Writing a module




See the next talk from Nick Anderson(!)

Have you heard
about org mode?





Contributing

- Core (C programming)
- Masterfiles (CFEngine policy language)
- Documentation (Markdown / CFEngine policy language)
-  build.cfengine.com website (markdown / html / css / hugo)
-  cfbs CLI (python)
-  Modules:
 - Promise types (**Python** or bash or another language)
 - Inventory / reporting data (CFEngine policy language)
 - Security hardening / automate tasks (CFEngine policy language)
 - Flexible input-based modules (CFEngine policy language)
 - Compliance reports (JSON / CFEngine MP GUI)