

# Writing Modules for CFEngine Build

Nick Anderson

*[2023-02-07 Tue]*

# Let's write a module for CFEngine Build!

- ▶ Policy inventorying or managaging the thing
- ▶ A public repository to host the module
- ▶ Metadata for cfbs
- ▶ Publish on build index

## Policy to inventory and or manage

The following policy was written live, with the audience during the session

## Let's put it in a namespace

- ▶ Don't want to collide with other policies

```
body file control
{
    namespace => "etc_issue_content";
}
```

## Let's set a default, but allow Augments override

```
bundle agent main
{
  vars:

  "body"
    string => "Access to this host is restricted.",
    comment => "Default content for /etc/issue",
    if => not( isvariable( "body" ) );
```

Let's promise the file content

```
files:
  "/etc/issue"
  content => "$(body)";
}
```

A public repository to host the module

Let's create a new repository

# Metadata for cfbs

- ▶ Module repo metadata
- ▶ Module metadata
  - ▶ Build steps
  - ▶ Input (optional)



## Module Repo

```
{
  "name": "Module Repository Name",
  "description": "A module repository for ....",
  "type": "module-repo",
  "provides": {
  }
}
```

## Module metadata

```
"module-name": {  
  "description": "This modules ...",  
  "tags": ["management"],  
  "version": "0.0.1",  
  "commit": "xyz",  
  "by": "https://github.com/nickanderson",  
  "repo": "https://github.com/nickanderson/module-repository",  
}
```

## Module metadata - Build Steps

`copy <source> <destination>` Copy source to destination

`run <command>` Run command

`delete <space separated paths>` Delete files recursively

`json <source> <destination>` Merge source into destination

`append <source> <destination>` Append source to destination

`directory <source> <destination>` Copy directory, add policy files to inputs, enable auto-run bundles, merge `def.json`

`policy_files <destination>` Add policy file to inputs

`bundles <bundlename>` Appends to bundlesequence

`input <source> <destination>` Merge module input into project Augments (`def.json`)

`destination` relative to build output

`source` relative to module repository root or sub-directory (if defined)

## Module metadata - Build Steps example

```
"steps": [  
  "copy policy.cf services/cfbs/modules/module-name/policy",  
  "input module-name/input.json def.json",  
  "bundles mynamespace:mybundle",  
  "policy_files services/cfbs/modules/module-name/policy.c  
]
```

## Module metadata - Input (optional)

```
"input": [  
  {  
    "type": "string",  
    "variable": "text",  
    "namespace": "etc_issue_content",  
    "bundle": "main",  
    "label": "Text",  
    "question": "What text should /etc/issue contain?"  
  }  
]
```

## Publish on build index

`https://github.com/cfengine/build-index`

- ▶ Open a Pull Request adding the appropriate metadata to `cfbs.json`
- ▶ `cfbs pretty cfbs.json` after editing

Thank you!