



STATE OF THE CFENGINE

NICK ANDERSON

IT'S BEEN A WHILE ...

RELEASES SINCE 2020-02-02

26!

2020

11 releases

Version	Date
3.12.4	2020-03-30
3.15.1	2020-03-30
3.12.4-2	2020-05-31
3.12.5	2020-06-02
3.15.2	2020-06-02
3.16.0	2020-06-23
3.12.5-2	2020-11-02
3.15.2-2	2020-11-02
3.15.3	2020-11-10
3.12.6	2020-11-10
3.17.0	2020-11-19

2021

8 releases

Version	Date
3.12.7	2021-06-08
3.15.4	2021-06-08
3.18.0	2021-06-24
3.15.4-2	2021-11-11
3.18.0-2	2021-11-11
3.15.5	2021-11-30
3.18.1	2021-11-30
3.19.0	2021-12-10

2022

8 releases

Version	Date
3.15.6	2022-06-29
3.18.2	2022-06-29
3.20.0	2022-07-01
3.20.0-2	2022-07-07
3.15.7	2022-11-14
3.18.3	2022-11-14
3.18.3-2	2022-11-18
3.21.0	2022-12-21

2 LTS SERIES REACHED END OF LIFE:

We bid farewell to 3.12 and 3.15 LTS series.

Version	Release Date
3.12.7	2021-06-08
3.15.7	2022-11-14

2 LTS SERIES BORN

We welcomed 3.18 and 3.21 LTS series.

Version	Date
3.18.0	2021-06-24
3.21.0	2022-12-21

CONTRIBUTIONS

(excluding CFEngine Enterprise)

2020

	Core	MPF	Docs
Commits	603	127	117
Contributors	16	14	11
New Contributors	4	2	1

2021

	Core	MPF	Docs
Commits	416	168	244
Contributors	17	14	12
New Contributors	7	5	5

2022

	Core	MPF	Docs
Commits	209	176	401
Contributors	13	12	9
New Contributors	2	1	2

CORE

CHANGES IN BEHAVIOR

- Default permissions on directories 700 (3.21.0)
- `rxdirs` defaults off as of (3.20.0)
- Files created by default for non `edit_line` promises that manage full file content (3.20.0)
- Bundle agent main renamed to `mpf_main` (3.20.0)
- Two `cf-execd` processes are now expected around the time of each `cf-agent` run (3.18.0)

<https://cfengine.com/tags/change-in-behavior/>

FUNCTIONS

14 new functions since 3.15.0.

CF_VERSION_AFTER()

```
bundle agent __main__
{
  reports:
    "Running $(sys.cf_version)";

    "considered after 3.18.0"
    if => cf_version_after( "3.18.0" );

    "considered after 3.3"
    if => cf_version_after( "3.3" );

    "considered NOT after 4.3.2"
    unless => cf_version_after( "4.3.2" );
}
```

```
R: Running 3.21.0
R: considered after 3.18.0
R: considered after 3.3
R: considered NOT after 4.3.2
```


CF_VERSION_AT()

```
bundle agent __main__
{
  reports:

    "$(sys.cf_version) == $(with)"
    with => "$(sys.cf_version)",
    if => cf_version_at( $(sys.cf_version) );

    "$(sys.cf_version) != 3.7.1"
    unless => cf_version_at( "3.7.1" );

    "$(sys.cf_version) == $(sys.cf_version_major).$(sys.cf_version_minor)"
    if => cf_version_at( "$(sys.cf_version_major).$(sys.cf_version_minor)" );
}
```

R: 3.21.0 == 3.21.0

R: 3.21.0 != 3.7.1

R: 3.21.0 == 3.21

CF_VERSION_BEFORE()

```
bundle agent __main__
{
  reports:
    "CFEngine $(sys.cf_version) was not released before 3.18.0"
    unless => cf_version_before( "3.18.0" );

    "CFEngine $(sys.cf_version) was not released before 3.3"
    unless => cf_version_before( "3.3" );

    "CFEngine $(sys.cf_version) was released before 4.3.2"
    if => cf_version_before( "4.3.2" );
}
```

R: CFEngine 3.21.0 was not released before 3.18.0

R: CFEngine 3.21.0 was not released before 3.3

R: CFEngine 3.21.0 was released before 4.3.2

CF_VERSION_BETWEEN()

```
bundle agent __main__
{
  vars:
    "minor_version_next"
    string => format( "%d", eval( "$(sys.cf_version_minor)+1", math ) );

    "minor_version_previous"
    string => format( "%d", eval( "$(sys.cf_version_minor)-1", math ) );

  reports:
    "$(with)"
    with => concat( "CFEngine $(sys.cf_version) is considered between ",
                    "3.$(minor_version_previous).0 and ",
                    "3.$(minor_version_next).0"),
    if => cf_version_between( "3.$(minor_version_previous).0",
                             "3.$(minor_version_next).0");
}
```

R: CFEngine 3.21.0 is considered between 3.20.0 and 3.22.0

CF_VERSION_MAXIMUM()

```
bundle agent __main__
{
    reports:
        "I am running $(sys.cf_version)";
        "This will be printed on versions of CFEngine up to 3.21.0."
        if => cf_version_maximum( "3.21.0" );

        "$(with)"
        with => concat( "This will be printed on versions of ",
                        "CFEngine up to 3.18.0, not on 3.18.1 or later"),
        if => cf_version_maximum( "3.18.0" );
}
```

R: I am running 3.21.0

R: This will be printed on versions of CFEngine up to 3.21.0.

CF_VERSION_MINIMUM()

```
bundle agent __main__
{
  reports:
    "I am running $(sys.cf_version)";

    "This message will start appearing on hosts running cfengine 3"
    if => cf_version_minimum( "3" );

    "This message will start appearing on hosts running cfengine 3.18"
    if => cf_version_minimum( "3.18" );

    "This message will start appearing on hosts running cfengine 3.21.0"
    if => cf_version_minimum( "3.21.0" );

    "This message will start appearing on hosts running cfengine 3.21.1"
    if => cf_version_minimum( "3.21.1" );

    "This will stop appearing once a host is running 4.18.1"
    unless => cf_version_minimum( "4.18.1" );
}
```

```
R: I am running 3.21.0
R: This message will start appearing on hosts running cfengine 3
R: This message will start appearing on hosts running cfengine 3.18
R: This message will start appearing on hosts running cfengine 3.21.0
R: This will stop appearing once a host is running 4.18.1
```

EXECRESULT_AS_DATA()

```
bundle agent __main__
{
  vars:
    "cmd" string => "$(sys.bindir)/cf-agent --version";
  reports:
    "$(with)"
    with => storejson( execresult_as_data( $(cmd), "noshell" ) );
}
```

```
R: {
  "exit_code": 0,
  "output": "CFEngine Core 3.21.0\nCFEngine Enterprise 3.21.0"
}
```

FINDFILES_UP()

aka search_up()

```
bundle agent __main__
{
  classes:
    "have_dir"
    expression => isdir( "/tmp/cfengine/example/find/files/up/" );

  files:
    "/tmp/cfengine/example/find/files/up/." create => "true";
    "/tmp/cfengine/example/FINDTHIS"       create => "true";
    "/tmp/cfengine/example/find/FINDTHAT"   create => "true";

  reports:
    have_dir::
      "First file starting with FIND is $(with)"
      with => nth( findfiles_up( "/tmp/cfengine/example/find/files/up",
                                "FIND*",
                                inf), 0 );
}
```

R: First file starting with FIND is /tmp/cfengine/example/find/FINDTHAT

INT()

```
bundle agent __main__  
{  
  reports:  
    "$(with)"  
    with => int( eval( "1+1", math ) );  
    "$(with)"  
    with => eval( "1+1", math );  
}
```

R: 2

R: 2.000000

STRING()

```
bundle agent __main__
{
  reports:
    `and( "linux", "cfengine" ) returns '$(with)'\`
      with => string( and( "linux", "cfengine" ) );

    `and( "windows", "cfengine" ) returns '$(with)'\`
      with => string( and( "windows", "cfengine" ) );
}
```

R: and("linux", "cfengine") returns 'any'

R: and("windows", "cfengine") returns '!any'

STRING_TRIM()

```
bundle agent __main__  
{  
  vars:  
    "t" string => string_trim( "  leading and trailing  ");  
  reports:  
    "$$(t)";  
}
```

```
R: 'leading and trailing'
```

TYPE()

Get the information about a variables type.

- string type() example

```
bundle agent __main__
{
  vars:
    "a_string" string => "defined in policy";
  reports:
    "type('a_string') is '$(with)'"
    with => type( 'a_string' );
    "type('a_string', 'yes') is '$(with)'"
    with => type( 'a_string', 'yes' );
}
```

R: type('a_string') is 'string'

R: type('a_string', 'yes') is 'policy string'

- list type() example

```
bundle agent __main__
{
  vars:
    "a_list" slist => { "defined", "in", "policy" };

  reports:

    "type('a_list') is '$(with)'"
    with => type( 'a_list' );
    "type('a_list', 'yes') is '$(with)'"
    with => type( 'a_list', 'yes' );
}
```

R: type('a_list') is 'slist'

R: type('a_list', 'yes') is 'policy slist'

- data array type() example

```
bundle agent __main__
{
  vars:
    "data_array_aka_list" data => '[ "defined", "in", "policy" ]';

  reports:
    "type('data_array_aka_list') is '$(with)'"
    with => type( 'data_array_aka_list' );
    "type('data_array_aka_list', 'yes') is '$(with)'"
    with => type( 'data_array_aka_list', 'yes' );
}
```

R: type('data_array_aka_list') is 'data'

R: type('data_array_aka_list', 'yes') is 'data array'

- data dict type() example

```
bundle agent __main__
{
  vars:
    "data_dict" data => '{ "something:" "defined in policy" }';

  reports:
    "type('data_dict') is '$(with)'"
    with => type( 'data_dict' );
    "type('data_dict', 'yes') is '$(with)'"
    with => type( 'data_dict', 'yes' );
}
```

R: type('data_dict') is 'data'

R: type('data_dict', 'yes') is 'data object'

- undefined type() example

```
bundle agent __main__  
{  
  reports:  
    "type('some_bundle.missing_variable') is '$(with)'"  
    with => type( 'some_bundle.missing_variable' );  
    "type('some_bundle.missing_variable', 'yes') is '$(with)'"  
    with => type( 'some_bundle.missing_variable', 'yes' );  
}
```

R: type('some_bundle.missing_variable') is 'none'

R: type('some_bundle.missing_variable', 'yes') is 'policy none'

VALIDJSON()

```
bundle agent __main__
{
  vars:
    "json_string_0" string => '{ "valid": "json" }';
    "json_string_1" string => '{ invalid json }';

  reports:
    "json_string_0 is valid"
      if => validjson( "${json_string_0}");

    "json_string_1 is invalid"
      unless => validjson( "${json_string_1}");
}
```

R: json_string_0 is valid

R: json_string_1 is invalid

VALIDDATA()

Currently only supports JSON. So, effectively the same as `validjson()`. **Looking for contributions** to add YAML and CSV support.

```
bundle agent __main__
{
  vars:
    "json_string_0" string => '{ "valid": "json" }';
    "json_string_1" string => '{ invalid json }';

  reports:
    "json_string_0 is valid"
      if => validdata( "$(json_string_0)", JSON);

    "json_string_1 is invalid"
      unless => validdata( "$(json_string_1)", JSON);
}
```

```
R: json_string_0 is valid
R: json_string_1 is invalid
```

AUGMENTS

NEW SYNTAX STRUCTURE

- Set tags and comments on variables and classes
- Target variables in any namespace, bundle
- Target classes in any namespace
- Host specific data

- Augments in new format
/tmp/def.json

```
{
  "variables": {
    "nondefault:nondef.myvar": {
      "value": "my value",
      "comment": "Defined via Augments",
      "meta": [ "metatag" ]
    }
  },
  "classes": {
    "nondefault:my_class_defined_by_class_expressions": {
      "class_expressions": [ "linux.ubuntu::", "Thursday::" ],
      "comment": "Defined via Augments",
      "meta": [ "metatag" ]
    },
    "nondefault:my_class_defined_by_regular_expressions": {
      "regular_expressions": [ "lin.*" ],
      "comment": "Class starting with linux was defined",
      "meta": [ "metatag" ]
    }
  }
}
```

- Leveraged from policy
/tmp/example.cf

```
bundle agent __main__  
{  
}
```

```
chmod 600 /tmp/example.cf  
cf-promises -f /tmp/example.cf --show-classes=nondefault --show-vars=nondefault
```

Class name	Meta tags
nondefault:my_class_defined_by_class_expressions	source=augments_file
nondefault:my_class_defined_by_regular_expressions	source=augments_file
Variable name	Variable value
nondefault:nondef.myvar	my value

- Host specific data

`WORKDIR/data/host_specific.json` is read **first**, before `def.json` relative to the policy entry.

- Does not support the `augments` key
- Does not expand variables (unlike `def.json`)
- Classes can only be defined (not defined based on presence of other classes)
- Variables can not be overridden by other Augments
- Automatically tagged with `source=cldb`

- host_specific.json and def.json targeting the same variable WORKDIR/data/host_specific.json

```
{
  "variables": {
    "nondefault:nondef.data01": {
      "value": "my value from host_specific.json"
    }
  }
}
```

/tmp/def.json

```
{
  "variables": {
    "nondefault:nondef.data01": {
      "value": "my value from def.json"
    },
    "nondefault:def.data01": {
      "value": "My value set from def.json"
    }
  }
}
```


- See the variable values defined
/tmp/example.cf

```
bundle agent __main__  
{  
}
```

```
chmod 600 /tmp/example.cf  
cf-promises -f /tmp/example.cf --show-vars=nondefault
```

Variable name	Variable value
nondefault:def.data01	My value set from def.json
nondefault:nondef.data01	my value from host_specific.json

- Defining classes

```
{
  "classes": {
    "nondefault:my_class": {
      "class_expressions": []
    },
    "nondefault:my_class2": {
      "class_expressions": [ "any::" ]
    },
    "nondefault:my_class3": {
      "regular_expressions": [ "any" ]
    },
    "nondefault:my_class4": {
      "regular_expressions": []
    },
    "nondefault:my_class_error": {
      "regular_expressions": [ "linux" ]
    },
    "nondefault:my_class_error2": {
      "class_expressions": [ "linux::" ]
    }
  }
}
```

/tmp/example.cf

```
bundle agent __main__  
{  
}
```

```
chmod 600 /tmp/example.cf  
cf-promises -f /tmp/example.cf --show-classes=nondefault
```

```
error: Invalid regular expression rules for class 'nondefault:my_class_error' in CM  
error: Invalid class expression rules for class 'nondefault:my_class_error2' in CMC  
Class name                                     Meta tags  
nondefault:my_class                             source=cldb  
nondefault:my_class2                           source=cldb  
nondefault:my_class3                           source=cldb  
nondefault:my_class4                           source=cldb
```

CUSTOM PROMISE TYPES

- Introduced in 3.17.0.
- Implement promise types in *any* language
- Libraries for python and bash are available

```

bundle agent __main__
{
  git:
    "/home/ubuntu/kubernetes"
    repo => "https://github.com/kubernetes/kubernetes";
  ansible:
    "my_playbook"
    playbook => "/northern.tech/playbook.yaml",
    inventory => "/northern.tech/inventory.yaml",
    tags      => {"helloworld"};
  groups:
    "foo"
    policy => "present",
    members => '{ "include": ["alice", "bob"],
                  "exclude": ["malcom"] }';
  http:
    "https://cfengine.com/images/cfengine-logo.svg"
    file => "/var/cfengine/cfengine-logo.svg",
    if => not(fileexists("/var/cfengine/cfengine-logo.svg"));
  systemd:
    "myservice"
    name => "myservice",
    enabled => "true",
    state => "started";
  scl_docker_network:
    "public"
    state => "present";
  scl_docker_compose:
    "/srv/traefik/docker_compose.yaml"
    state => "start";
}

```

TOOLING

CF-REMOTE THE BEST WAY TO GET PACKAGES

Now available in Pypi. `pip install cf-remote!`

```
exec 2>&1
```

```
cf-remote --version 3.21.x list --edition community | head -n 3  
:
```

Available releases: master, 3.21.x, 3.21.0, 3.18.x, 3.18.3, 3.18.2, 3.18.1, 3.18.0

Using 3.21.x LTS:

<http://buildcache.cfengine.com/packages/testing-pr/jenkins-community-nightly-3.21.x-29>

CFENGINE BUILD

A place and tooling to facilitate sharing and consumption of CFEngine content (modules).

BUILD.CFENGINE.COM

[Modules](#) ▾[Resources](#) ▾

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 Elgesem

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

0.2.0

**uninstall-ftp**

by Craig Comstock

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

0.0.2

**promise-type-http**

by Vratislav Podzimek

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

1.1.0

supported

supported

Popular modules

Modules with the most downloads

**masterfiles**

by CFEngine

Official CFEngine Masterfiles Policy Framework (MPF).

3.21.0

**autorun**

by Ole Herman Schumacher Elgesem

Enables autorun functionality.

1.0.1

**library-for-promise-types-in-python**

by CFEngine

Library enabling promise types implemented in python.

0.1.1

supported

supported

supported

CFBS

Command line tool to manage your policy set.

```
+ cfbs help
usage: cfbs [-h] [--loglevel LOGLEVEL] [--version] [--force]
            [--non-interactive] [--index INDEX] [--check]
            [--checksum CHECKSUM] [--keep-order] [--git {yes,no}]
            [--git-user-name GIT_USER_NAME] [--git-user-email GIT_USER_EMAIL]
            [--git-commit-message GIT_COMMIT_MESSAGE] [--ignore-versions-json]
            [--masterfiles MASTERFILES]
            [cmd] [args ...]

CFEngine Build System.

positional arguments:
  cmd                  The command to perform (pretty, init, status, search,
                        add, remove, clean, update, validate, download, build,
                        install, help, info, show, input, set-input, get-
                        input)
  args                Command arguments

options:
  -h, --help            show this help message and exit
  --loglevel LOGLEVEL, -l LOGLEVEL
                        Set log level for more/less detailed output
  --version, -V          Print version number
  --force               Force rebuild / redownload
  --non-interactive      Don't prompt, use defaults (only for testing)
  --index INDEX          Specify alternate index
  --check               Check if file(s) would be reformatted
  --checksum CHECKSUM    Expected checksum of the downloaded file
  --keep-order           Keep order of items in the JSON in 'cfbs pretty'
  --git {yes,no}         Override git option in cfbs.json
  --git-user-name GIT_USER_NAME
                        Specify git user name
  --git-user-email GIT_USER_EMAIL
                        Specify git user email
  --git-commit-message GIT_COMMIT_MESSAGE
                        Specify git commit message
  --ignore-versions-json
                        Ignore versions.json. Necessary in case of a custom
                        index or testing changes to the default index.
  --masterfiles MASTERFILES
                        Add masterfiles on cfbs init choose between

+ :
```

- Find modules on build.cfengine.com

```
+ cfbs search inventory
inventory-clamav - Adds reporting data (inventory) for useful information from ClamAV (version, definitions version, definitions date).
inventory-etc-hosts - Adds reporting data (inventory) for entries from the /etc/hosts file.
inventory-etc-login-defs - Adds reporting data (inventory) for useful bits from the /etc/login.defs file.
inventory-fips-mode-setup - Adds reporting data (inventory) for the status of fips-mode-setup.
inventory-kernel-boot-params - Adds reporting data (inventory) for kernel parameters set during system boot.
inventory-kernel-settings-sysctl-conf - Adds reporting data (inventory) for settings from the /etc/sysctl.conf file.
inventory-kernel-settings-sysctl-current - Adds reporting data (inventory) for sysctl settings current state.
inventory-lastlog - Adds reporting data (inventory) for users who have logged in and when.
inventory-local-groups - Adds reporting data (inventory) for the local groups on the system.
inventory-local-users - Adds reporting data (inventory) for the local users on the system with their attributes.
inventory-local-users-locked - Adds reporting data (inventory) for the local users on the system that are locked.
inventory-local-users-non-root-has-uid-0 - Adds reporting data (inventory) for the non-root users that have uid 0.
inventory-local-users-password-empty - Adds reporting data (inventory) for the local users on the system that do not have a password set.
inventory-local-users-password-hashing-algorithm - Adds reporting data (inventory) for password hashing algorithms in use by local users.
inventory-unshadowed-users (inventory-local-users-unhashed-password) - Adds reporting data (inventory) on local users in /etc/passwd not us
inventory-openssl-versions - Adds an inventory attribute containing all versions of OpenSSL found on the system.
inventory-ssh-host-key-fingerprints - Adds reporting data (inventory) for the SSH host key fingerprints.
inventory-sudoers - Adds reporting data (inventory) for users with sudo access.
inventory-systemd - Adds reporting data (inventory) for interesting things from systemd.
inventory-writable-directories-in-root-path - Reporting data (inventory) directories in root's $PATH that are world or group writable.
inventory-yum-update-info - Adds information about security updates from 'yum updateinfo' to inventory.
+ :
```

- Construct a project from disparate modules

```
+ mkdir -p /tmp/cfbs-project
+ cd /tmp/cfbs-project
+ cfbs --non-interactive init
Initialized empty Git repository in /tmp/cfbs-project/.git/
[main (root-commit) c5f6fc2] Initialized a new CFEngine Build project
 1 file changed, 7 insertions(+)
  create mode 100644 cfbs.json
[main 3a2664f] Added module 'masterfiles'
 1 file changed, 13 insertions(+), 1 deletion(-)
Committing using git:

Initialized an empty project called 'Example project' in 'cfbs.json'
Added module: masterfiles
Committing using git:

+ cfbs --non-interactive add inventory-lastlog
[main c20fe95] Added module 'inventory-lastlog'
 1 file changed, 14 insertions(+)
Added module: inventory-lastlog
Committing using git:
```

- Reproducably build your policy set

```
+ cfbs build

Modules:
001 masterfiles      @ 379c69aa71ab3069b2ef1c0cca526192fa77b864 (Downloaded)
002 inventory-lastlog @ 7436a6ca9d1a159e6083768b9c06d9b1cbb189a0 (Downloaded)

Steps:
001 masterfiles      : run './prepare.sh -y'
001 masterfiles      : copy './' 'masterfiles/'
002 inventory-lastlog : copy 'policy/main.cf' 'masterfiles/services/inventory-lastlog/'
002 inventory-lastlog : json 'cfbs/def.json' 'masterfiles/def.json'

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
+ tree out/masterfiles/services
out/masterfiles/services
├── autorun
│   ├── hello.cf
│   └── README.md
├── init.cf
├── inventory-lastlog
│   └── main.cf
└── main.cf

2 directories, 5 files
+ :
```

CF-SUPPORT

Command line tool (added in 3.18.3, 3.21.0) to facilitate data collection when reporting issues.

```
+ cf-support
Analyzing CFEngine core dumps
Using apport-unpack to analyze CFEngine core dumps
Captured output of command hostname -f
Captured output of command uname -a
Captured output of command cat /etc/os-release
Captured output of command /var/cfengine/bin/cf-promises -V
Captured output of command free -m
Captured output of command df -h
Captured output of command ps auwx
Captured output of command top -b -H -c -n1
Captured output of command netstat -ie
Captured output of command ifconfig
Captured output of command lsof
Captured output of command mount
Added file /etc/fstab
Captured output of command /var/cfengine/bin/cf-key -p /var/cfengine/ppkeys/localhost.pub
Captured output of command grep 'version =' /var/cfengine/inputs/promises.cf
Captured output of command /var/cfengine/bin/cf-key -s
Captured output of command /var/cfengine/bin/cf-check diagnose
Captured output of command systemctl status cfengine3
Added compressed copy of file /var/log/CFEngine-Install-2023-01-05_20:32:17_UTC.log
Added compressed copy of file /var/log/CFEngine-Install-2023-01-05_20:32:25_UTC.log
Added compressed copy of file /var/log/CFEngine-Install.log
Captured output of cat /var/log/syslog filtered for cf-|CFEngine
Added compressed copy of file /var/cfengine/outputs/previous
Added file /var/cfengine/policy_server.dat
Captured output of command cf-serverd -Fv | awk '/=== BEGIN summary of access promises ==/,/=== END summary of access promises ===/'
Added compressed copy of file /var/log/postgresql.log
Added file /var/cfengine/state/pg/data/postgresql.conf
Added file /var/cfengine/state/pg/data/pg_hba.conf
Captured output of command /var/cfengine/bin/psql cfdb -c "SELECT count(*) as host_count from __hosts;"
Captured output of command /var/cfengine/bin/psql cfdb -c "select name, details, units, avg(value), count(value) from diagnostics group by name, details, units;"
Captured output of command /var/cfengine/bin/psql cfdb -c "select host, status, type, count(*) from __status group by host, status, type order by host, status;"
Added directory /var/cfengine/masterfiles
Please send cfengine_support_case_NA-demohub.a10042.cfengine.com-2023-02-01-2155.tar.gz to CFEngine support staff.
+ :
```

DOCUMENTATION

- Style, navigation, content and organization updates
- Responsive mobile design
- New search

Version

Version 3.21 (LTS)

Overview

Getting Started

Release Notes

Web UI

API

Examples and Tutorials

Reference

Resources

Welcome to CFEngine Documentation

This site contains information on how to manage and automate the infrastructure with CFEngine. It includes the reference for the following versions of CFEngine:

- CFEngine 3.21 (LTS)
- CFEngine 3.18 (LTS)

Getting started

Installation

Download and install CFEngine for the first time.

Modules

Use modules to easily add reports or get things done without writing any code.

Reporting and Web UI

Know more about your infrastructure and hosts, their data, compliance and make changes from within the Web UI.

Writing policy

Write and deploy your first policy files to make changes to systems.

Developing modules

Turn your policy, reoprts, or python code into CFEngine Build modules for others to use.

Tutorial series on policy language

In-depth tutorials on how to work with CFEngine policy.

Popular

Promise types

Learn about processes, packages, users, files, storage, services, etc.

API reference

The API is a conventional REST API which supports one or more GET, PUT, POST, or DELETE operations.

Language concepts

Learn Bundles, Bodies, Promises, Classes and Decisions, Variables, etc.

Manage packages

Learn how to install, manage and remove packages using CFEngine.

CFEngine Build

CFEngine Build is a catalogue of policy and modules created by CFEngine, our partner and community that helps you to simplify the automation process.

Go to the page

ENTERPRISE

CF-REACTOR

New component to more quickly react to changes in data

REFRESHED UI STYLE

CFEngine

ENTERPRISE

Find host

6 Hosts

Health

Hub

Hello, admin

Dashboard

Hosts

Reports

Measurements

Policy analyzer

Build

Default dashboard

Changes

Date	Changes
Fri January 27	8k
Sat January 28	7k
Sun January 29	0
Mon January 30	0
Tue January 31	0
Wed February 01	2k
Thu February 02	3k

Software updates

17%

Low

83%

Alert triggered on 5 / 6 hosts

Policy compliance

✓

System health

✓

No alert triggered

OS

CentOS 8: 1

Debian 10: 1

Ubuntu 18: 1

Ubuntu 22: 2

Windows 10: 1

Hosts

1

1

1

2

1

Newly bootstrapped hosts

Group by Day

Date range 2023-01-27 - 2023-02-02

Date	Count
27 Jan 2023	0
28 Jan 2023	0
29 Jan 2023	0
30 Jan 2023	0
31 Jan 2023	0
01 Feb 2023	0
02 Feb 2023	0

Decommissioned hosts

Group by Day

Date range 2023-01-27 - 2023-02-02

Date	Count
27 Jan 2023	0
28 Jan 2023	0
29 Jan 2023	0
30 Jan 2023	0
31 Jan 2023	0
01 Feb 2023	0
02 Feb 2023	0

File changes

Date	File changes
27 Jan 2023	2k
28 Jan 2023	2k
29 Jan 2023	0
30 Jan 2023	0
31 Jan 2023	0
01 Feb 2023	1k
02 Feb 2023	1k

Not kept, not repaired count

Date	Count
27 Jan 2023	0
28 Jan 2023	0
29 Jan 2023	0
30 Jan 2023	0
31 Jan 2023	0
01 Feb 2023	0
02 Feb 2023	0

+ Add

Event log

0 events in the log

View more

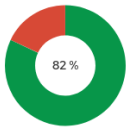
COMPLIANCE REPORTS

CISOfy Lynis (3.0.8)

Filter

Overview

Overall compliance



82 % compliance over 62 checks among 6 hosts

Compliance by category



Checklist

Authentication

Name	Severity level	Passing
Lynis:AUTH-9204 - Multiple users with UID zero	high	4 / 4 (100%) 🔗
Lynis:AUTH-9229 - Password hashing methods	high	2 / 4 (50%) 🔗

Banners

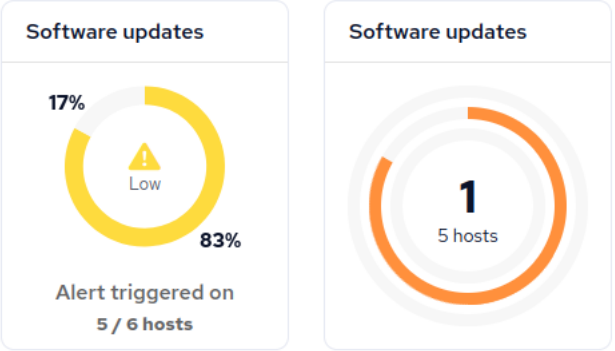
Name	Severity level	Passing
Lynis:BANN-7124 - Issue banner file	high	4 / 4 (100%) 🔗
Lynis:BANN-7126 - Issue banner file contents	high	3 / 4 (75%) 🔗
Lynis:MAIL-8818 - Postfix banner	high	4 / 4 (100%) 🔗

Containers

Name	Severity level	Passing
Lynis:CONT-8004 - Solaris zones	high	0 / 0 (100%) 🔗
Lynis:CONT-8102 - Docker status	high	4 / 4 (100%) 🔗
Lynis:CONT-8107 - Unused Docker containers	medium	4 / 4 (100%) 🔗

NEW DASHBOARD WIDGETS

SINGLE ALERT



COMPLIANCE

CISOfy Lynis Compliance

82 %

Checks	Failing hosts
⚠ Lynis:ACCT-9622...	5
⚠ Lynis:KRNL-5820...	5
⚠ Lynis:USB-1000 -...	5
⚠ Lynis:ACCT-9628...	4
⚠ Lynis:FILE-6310 -...	4

[Open full report](#)

RE-STYLED HOST INFO PAGE

debian-10.west.lawrence.ks.a10042.cfengine.com



CFEngine

ID:	SHA-2423ce6c4b0439f7e32a7f34d68c3c8627f68aa17f6alc4a9256dd4c8a04635a
Version:	3.21.0
Role:	Bootstrapped Host
IP Reported from:	24.56.150.37
Initiated by:	Client
Average agent execution interval:	320 seconds
Last report:	2023-02-02 12:55-06:00
Last agent run:	2023-02-02 12:54-06:00
First report:	2023-01-12 12:27-06:00

System

OS:	Debian 10
Hostname:	debian-10.west.lawrence.ks.a10042.cfengine.com
IPv4 addresses:	192.168.42.246
Installed packages:	362
Updates available:	87

Status

Average load:	0.12%
Average free disk:	80.00%

Host specific data

Variables

[Add Variable](#)

Classes

[Add Class](#)

Suggestions ^

- Evaluate policy every minute
 - Enable monitoring data for non-hubs
 - Enable autorun
 - Add policy files to inputs
 - Add policy bundles to the evaluation bundle sequence
- Cancel

Add

Measurements

loadavg ▾

Last updated:	1969-12-31 18:00:00-06:00
Unit:	jobs
Last measured value:	not available



Inventory

Open inventory report

Attributes

Policy Servers:	137.184.45.224 🔗
Ports listening:	22, 5308 🔗
Kernel Release:	4.19.0-8-amd64 🔗
Interfaces:	enp0s4 🔗
MAC addresses:	00:a0:98:03:8c:1d 🔗
/etc/issue content:	You are accessing an Information System (IS) that is provided for authorized use only. By using this IS (which includes any device attached to this IS), you understand and consent that communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any purpose. 🔗

Classes

debian:	defined 🔗
systemd:	defined 🔗
xen:	not defined 🔗
redhat:	not defined 🔗

Show all classes

Variables

default.sys.uqhost:	debian-10 🔗
default.sys.fqhost:	debian-10.west.lawrence.ks.a10042.cfengine.com 🔗
Show all variables	

HOST SPECIFIC DATA

Set variables and classes for specific hosts from Mission Portal

Host specific data

Suggestions ▾

Variables

default:def.acl	0.0.0.0/0	...
etc_issue_content:main.text	All of your base are belong to us!	...
<div>⊕ Add Variable</div>		< 1 / 1 >

Classes

default:cfengine_internal_masterfiles_update	Update default policy distribution point from upstream repository.	...
default:mpf_augments_control_enabled	Have the MPF track settings and restart CFEngine components (cf-serverd, cf-monitord) upon change.	...
exception_uninstall_bind	Let bind live here	...
<div>⊕ Add Class</div>		< 1 / 1 >

PINNED INVENTORY, CLASSES, AND VARIABLES

Inventory

Attributes

Policy Servers:

143.198.244.178

Ports listening:

22, 5308, 5355

Kernel Release:

4.18.0-147.5.1.el8_1.x86_64

Interfaces:

enp0s4

MAC addresses:

00:a0:98:04:0e:62

/etc/issue content:

You are accessing an Information System (IS) that is provided for authorized use only. By using this IS (which includes any device attached to this IS), you understand and consent that communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any purpose.

Show all attributes

Classes

debian:

not defined

systemd:

defined

xen:

not defined

redhat:

defined

Show all classes

Variables

default.sys.uqhost:

centos-81

default.sys.fqhost:

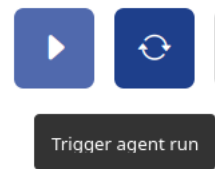
centos-81.west.lawrence.ks.a10042.cfengine.com

Show all variables

Open inventory report

66

TRIGGER AGENT RUNS & REPORT COLLECTION



CFENGINE BUILD IN MISSION PORTAL

CFEngine

ENTERPRISE

Dashboard

Hosts

Reports

Measurements

Policy analyzer

Build

CFEngine Build

Project

cfengine-enterprise-de...

Added modules

masterfiles

client-initiated-reporting

./windows-file-ownershi...

lynis

./collect-all-measureme...

./mpf-augments-control...

compliance-report-impo...

autorun

compliance-report-lynis

etc-issue-content

etc-issue-access

demo-select-business-u...

cfengine-demo-churn-lo...

library-parsed-local-users

inventory-local-users

inventory-kernel-boot-p...

Find host

6 Hosts

Health

Hub

Hello, admin

Search modules

Push and deploy

Welcome to CFEngine Build

Here you can add and configure modules easily to your projects and help you to automate the process with less effort.

Find modules

Modules for security

tmp-nosuid

by Vratislav Podzimek

Makes sure /tmp is mounted with the 'nosuid' option.

version: 0.0.2

lynis

by Nick Anderson

Automates the installation, running, and reporting of CISOfy's lynis system audits.

version: 0.3.2

inventory-sudoers

by Nick Anderson

Adds reporting data (inventory) for users with sudo access.

version: 1.0.2

Modules for compliance

tmp-nosuid

by Vratislav Podzimek

Makes sure /tmp is mounted with the 'nosuid' option.

version: 0.0.2

lynis

by Nick Anderson

Automates the installation, running, and reporting of CISOfy's lynis system audits.

version: 0.3.2

inventory-yum-update-info

by Vratislav Podzimek

Adds information about security updates from 'yum updateinfo' to inventory.

version: 0.0.1

inventory-clamav

by Nick Anderson

Adds reporting data (inventory) for useful information from ClamAV (version, definitions version, definitions date).

version: 1.1.0

compliance-report-lynis

by Nick Anderson

68

MORE TALKS

- CFEngine on odd platforms
- Using Ansible modules in CFEngine
- An introduction to CFEngine Build
- Writing Modules for CFEngine Build
- Personal Configuration Management with CFEngine

THANK YOU

If there is something else you are interested in please reach out.

