

# State of the CFEngine

Or Since last time at Config Management Camp

Nick Anderson

January 14, 2018

## Releases

5 releases

Date	Release
2017-03-30	3.7.5
2017-03-30	3.10.1
2017-08-11	3.10.2
2017-08-11	3.11.0
2017-09-12	3.7.6

## Contributions

### Core

- 24 contributors in the last year
- 10 new contributors

### MPF

- 14 contributors in the last year
- 3 new contributors

### Documentation

- 17 contributors
- 3 new contributors

## Functionality

### Core

with attribute

```
bundle agent main
{
```

```
  vars:
```

```
    "todo" slist => { "a 1", "b 2", "c 3" };
```

```
    # Here, 'with' is the canonified version of $(todo), letting us avoid an
```

```
    # intermediate canonification array.
```

```
    "$ (with)" string => "$(todo)", with => canonify($(todo));
```

```
    "complex" data => '
```

```
{
```

```
  "x": 200,
```

```
  "y": [ 1, 2, null, true, false ]
```

```
}
```

```
';
```

```
  reports:
```

```
    "For iterable '$(todo)' we created variable '$(with)' and its value is '$(todo)'"
    with => canonify($(todo));
```

```
    "We can print a data container compactly without creating a temporary variable: "
    with => format("%S", complex);
```

```
    "We can print a data container fully without creating a temporary variable: $(wi"
    with => storejson(complex);
```

```
}
```

Figure 1: With attribute usage example policy

with attribute

```
R: For iterable 'a 1' we created variable 'a_1' and its value is 'a 1'
```

```
R: For iterable 'b 2' we created variable 'b_2' and its value is 'b 2'
```

```
R: For iterable 'c 3' we created variable 'c_3' and its value is 'c 3'
```

```
R: We can print a data container compactly without creating a temporary variable: {"x"
```

R: We can print a data container fully without creating a temporary variable: {

```
"x": 200,  
"y": [  
  1,  
  2,  
  null,  
  true,  
  false  
]  
}
```

### Multiple augments

**NOTE:** Current implementation discussion in progress in CFE-2741. Please consider participating.

```
{  
  "vars":{  
    "my_var": "defined in def.json",  
    "my_other_var": "Defined ONLY in def.json"  
  },  
  "augments": [  
    "/tmp/$(sys.flavor).json"  
  ]  
}
```

Figure 2: Example /tmp/def.json

### Multiple augments

```
{  
  "vars": {  
    "my_var": "Overridden in centos_6.json",  
    "centos_6_var": "Defined ONLY in centos_6.json"  
  }  
}
```

Figure 3: Example /tmp/centos\_6.json

## Multiple augments

```
[root@hub tmp]# cf-agent -KIf ./example.cf
R: def.my_var == Overridden in centos_6.json
R: def.my_other_var == Defined ONLY in def.json
R: def.centos_6_var == Defined ONLY in centos_6.json
```

### inline\_mustache

You are no longer required to use an external file template. Now you can provide the template directly within the policy.

```
bundle agent main
{
  vars:
    "d" data => '{ "hello": "world", "feature": [ "render", "inline", "mustache" ] }';

  files:

    "/tmp/example.txt"
      create => "true",
      template_method => "inline_mustache",
      edit_template_string => "{{%-top-}}",
      template_data => @(d);
}
```

Figure 4: Example `template_method inline_mustache`

### inline\_mustache

Results in `/tmp/example.txt` having this content.

```
{
  "feature": [
    "render",
    "inline",
    "mustache"
  ],
  "hello": "world"
}
```

Figure 5: Example `template_method inline_mustache`

`missing_ok`

## MPF

### Augments - Append the bundlesequence of `promises.cf` and `update.cf`

- `control_common_bundlesequence_end`
- `control_common_update_bundlesequence_end`

```
{
  "vars":{
    "control_common_bundlesequence_end": [ "mybundle1", "mybundle2" ],
    "control_common_update_bundlesequence_end": [ "my_updatebundle1", "mybundle2" ]
  }
}
```

- Order of bundle actuation not guaranteed.
- Agent errors if named bundle is missing.

### Augments - Log file retention and rotation limits

`mpf_log_files_max_size` Max file size before rotation

`mpf_log_file_retention` Number of file rotations to keep

`mpf_log_dir_retention` Number of file rotations to keep in `outputs`, `reports`,  
and the Enterprise application log directory.

```
{
  "vars": {
    "mpf_log_file_retention": "5",
    "mpf_log_file_max_size": "10M",
    "mpf_log_dir_retention": "7"
  }
}
```

### **Augments - Execution schedule**

`control_executor_schedule` Classes that trigger execution of `cf-agent`.

```
{
  "vars": {
    "control_executor_schedule": [ "Min00", "Min30" ]
  }
}
```

### **Augments - splaytime**

`control_executor_splaytime` Maximum number of minutes `exec_commad` should wait before executing.

```
{
  "vars": {
    "control_executor_splaytime": "3"
  }
}
```

### **Augments - allowlegacyconnects**

`control_server_allowlegacyconnects` List of subnets allowed to connect using legacy protocol (versions prior to 3.7.0).

```
{
  "vars": {
    "control_server_allowlegacyconnects": [ "0.0.0.0/0" ]
  }
}
```

### **Augments - maxconnections**

`control_serverd_maxconnections` Maximum number of connections allowed by `cf-serverd`.

```
{
  "vars":{
    "control_serverd_maxconnections": "1000"
  }
}
```

### **Augments - Client initiated reporting (Enterprise)**

**control\_hub\_exclude\_hosts** List of subnets to exclude from hub initiated report collection.

**client\_initiated\_reporting\_enabled** List of classes that if defined should initiate reporting to an enterprise hub.

**control\_server\_call\_collect\_interval** Number of minutes between client initiated reporting.

```
{
  "vars": {
    "control_server_call_collect_interval": "1",
    "control_hub_exclude_hosts": [ "0.0.0.0/0" ]
  },
  "classes" {
    "client_initiated_reporting_enabled": [ "any" ]
  }
}
```

### **Augments - files\_single\_copy**

**control\_agent\_files\_single\_copy** List of regular expressions matching files that should not be copied more than once.

```
{
  "vars":{
    "control_agent_files_single_copy": [ ".*" ]
  }
}
```

### **Augments - default\_repository**

**mpf\_control\_agent\_default\_repository** List of classes class will cause these backups to be placed in `$(sys.workdir)/backups`.

**control\_agent\_default\_backup** Directory where backups should be placed (defaults to `$(sys.workdir)/backups`).

```
{
  "classes": {
    "mpf_control_agent_default_repository": [ "any" ]
  },

  "vars": {
    "control_agent_default_repository": "/var/cfengine/edit_backups"
  }
}
```

#### templates shortcut

`dir_templates` Path to common template directory. Shortcut provided by `cf-serverd` as `templates/`.

```
{
  "vars": {
    "dir_templates": "/var/cfengine/mytemplates"
  }
}
```

#### bundle agent example

```
{
  files:

    "$(def.dir_templates)/mytemplate.mustache" -> { "myservice" }

    copy_from => remote_dcp( "templates/mytemplate.mustache",
                           $(sys.policy_server) ),

    comment => "mytemplate is necessary in order to render
               myservice configuration file.";
}
```

#### Automatically restart components on related data change

While the agent itself will reload its config upon notice of policy change this bundle specifically handles changes to variables used in the MPF which may come from external data sources which are unknown to the components themselves.



`mpf_augments_control_enabled` List of classes that automatic component restart on related data change should be enabled for.

```
{
  "classes":{
    "mpf_augments_control_enabled": [ "any" ]
  }
}
```

### Host info report now renders inventory variables

```
cf-agent -Klb host_info_report
```

```
### Inventory
```

```
#### Variables tagged for inventory
```

```
{
  "default:cfe_autorun_inventory_disk.free": "5.00",
  "default:cfe_autorun_inventory_listening_ports.ports": [
    "22",
    "25",
    "53",
  ],
  "default:cfe_autorun_inventory_memory.total": "32050.27",
  "default:sys.arch": "x86_64",
  "default:sys.cf_version": "3.11.0",
  "default:sys.class": "linux",
  "default:sys.cpus": "4",
  "default:sys.flavor": "ubuntu_17",
  "default:sys.hardware_addresses": [
    "5c:e0:c5:9f:f3:8f",
    "52:54:00:6b:62:06",
    "02:42:79:79:f6:02",
    "0a:00:27:00:00:00"
  ],
  "default:sys.inet": {
    "default_gateway": "192.168.42.1",
```

Figure 6: Sample from inventory section of host info report

## Enterprise

### UI responsiveness

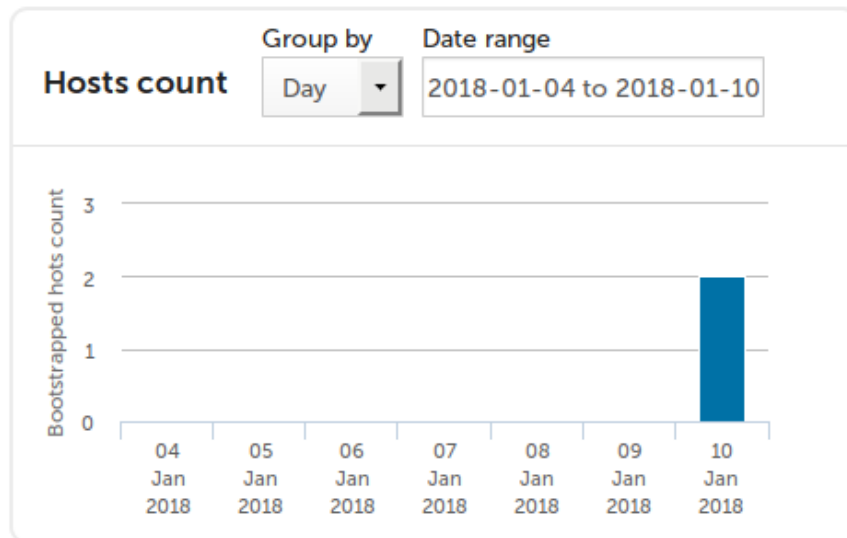
- Testing with 50,000 host data sets

### Global Host Search

- Easily find hosts by name, ip or identity



### host count trend widget



### mail settings

- Exported reports can now be sent as attachments in emails

CFEngine

Find host:

1 host

Health 📶

HA status 📶

Help, admin ⌵

Dashboard

Hosts

Reports

Monitoring

Design Center info

User management

Active tokens

Preferences

Manage apps

Version control repository

Host identifier

Custom notification scripts

Mail settings

About CFEngine

Mail settings

Send test email

Mail protocol ☐ Sendmail ☒ SMTP

Default from email

Max email attachment size (MB)

SMTP host

SMTP user

SMTP password

SMTP port

SMTP encryption

Apply

## LDAP settings API

User management

Active tokens

Preferences

Manage apps

Version control repository

Host identifier

Custom notification scripts

Mail settings

Authentication settings

About CFEngine

Authentication settings

Authentication method ☐ Internal ☒ LDAP

LDAP API Uri

Ldap Host

The full domain name of the directory server e.g test.cfengine.com

Port

Encryption ☒ None ☐ SSL ☐ STARTTLS

Base DN

Base domain name of the directory server e.g dc=test,dc=cfengine,dc=com

Login attribute

Used as user name, e.g uid or sAMAccountName

Ldap filter

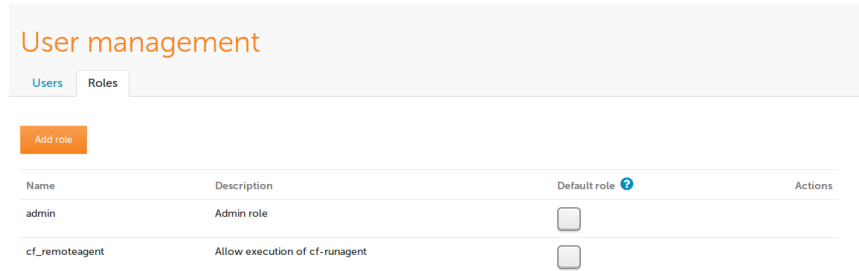
Filter to use when searching user objects e.g '(objectClass=inetOrgPerson)'

LDAP bind user name

This user needs to have search permissions

LDAP bind password

## default roles



Name	Description	Default role <a href="#">?</a>	Actions
admin	Admin role	<input type="checkbox"/>	
cf_remoteagent	Allow execution of cf-runagent	<input type="checkbox"/>	

## New OOTB Inventory Attributes

- Policy Release Id
- AIX OS Level

## Inventory API

```
curl --user admin -X POST \  
  -H 'content-type: application/json' \  
  https://hub/api/inventory -d '{ "select": [ "Host name", "OS type"] }'
```

Figure 7: Example API Query

## Inventory API

```
#+BEGIN_SRC json { "data": [ { "header": [ { "columnName": "Host  
name", "columnType": "STRING" }, { "columnName": "OS type", "column  
Type": "STRING" } ], "queryTimeMs": 11, "rowCount": 2, "rows": [ [  
"host001", "linux" ], [ "hub", "linux" ] ] } ], "meta": { "count": 1, "page":  
1, "timestamp": 1515607751, "total": 1 } }
```