

cdist configuration management

History and current status at ETH Zurich

Nico -telmich- Schottelius

2011-04-27

Introduction

cdist

Installation

Using

Update

Outline

- ▶ From configuration management
- ▶ to puppet and existing products
- ▶ to cdist development, design and status

Why Configuration Management?

- ▶ Needed for automatisisation
- ▶ Removes burden on reinstallation
- ▶ Helps to scale to a big number of hosts
- ▶ Has documentation character

Why cdist?

- ▶ Stopped using puppet (mostly due to bugs and non-intuitive configuration)
- ▶ Cfengine2 is more complicated
- ▶ Cfengine3 spams into your logfiles
- ▶ Did not investitage to deep into chef (based on puppet) or bcfg2

Features

- ▶ Simplicity: There is only one type to extend cdist called type
- ▶ Design: Type and core cleanly separated
- ▶ Design: Sticks completely to the KISS (keep it simple and stupid) paradigm
- ▶ Design: Meaningful error messages - do not lose time debugging error messages

Features (2)

- ▶ Design: Consistency in behaviour, naming and documentation
- ▶ Design: No surprise factor: Only do what is obviously clear, no magic
- ▶ Design: Define target state, do not focus on methods or scripts
- ▶ Design: Push architecture: Instantly apply your changes

Features (3)

- ▶ Small core: cdist's core is very small - less code, less bugs
- ▶ Fast development: Focus on straightforwardness of type creation is a main development objective

Features (4)

- ▶ Requirements, Scalability: No central server needed, cdist operates in push mode and can be run from any computer
- ▶ Requirements, Scalability, Upgrade: cdist only needs to be updated on the master, not on the target hosts
- ▶ Requirements, Security: Uses well-know SSH as transport protocol
- ▶ Requirements, Simplicity: Requires only shell and SSH server on the target

Features (5)

- ▶ UNIX: Reuse of existing tools like cat, find, mv, ...
- ▶ UNIX, familiar environment, documentation: Is available as manpages and HTML
- ▶ UNIX, simplicity, familiar environment: cdist is written in POSIX shell
- ▶ UNIX, simplicity, familiar environment: cdist is configured in POSIX shell

Installation

```
git clone git://git.schottelius.org/cdist
```

Documentation: Manpages

```
# Requires asciidoc / a2x  
make man
```

Setup PATH and MANPATH

```
cd cdist
eval './bin/cdist-env'
echo $PATH
echo $MANPATH
```

Deploying configuration to a host

```
# So wird es laufen...  
cdist-deploy-to localhost
```

Initial manifest

```
cat << eof > conf/manifest/init
__file /etc/cdist-configured

case "$__target_host" in
    localhost)
        __link /tmp/cdist-testfile \
            --source /etc/cdist-configured \
            --type symbolic
        __addifnosuchline /tmp/cdist-welcome \
            --line "Welcome to cdist"
    ;;
esac
eof
# Needs to be executable
chmod u+x conf/manifest/init
```

Now with working initial manifest

```
cdist-deploy-to localhost
```


Combine functionality

- ▶ Types
- ▶ conf/type/*
- ▶ __ prefixed (do not clash with other stuff in \$PATH)
- ▶ i.e. webserver, mailserver, desktop, ...

Creating a new type

```
mkdir conf/type/__my_mailserver
cat << eof > conf/type/__my_mailserver/manifest
__package nullmailer --state installed

require="__package/nullmailer" \
    __file /etc/nullmailer/remotes \
    --source "$__type/files/remotes"
eof
chmod u+x conf/type/__my_mailserver/manifest

mkdir conf/type/__my_mailserver/files
echo my.fancy.smart.host > \
    conf/type/__my_mailserver/files/remotes
```

Versions

- ▶ x.y: Stable version
- ▶ master: Development

Checkout and stay on stable version

```
git checkout -b 1.6 origin/1.6
```

Update to latest

```
git pull
```

End

That's it!

Have fun! - More information:

<http://www.nico.schottelius.org/software/cdist/>