

Configuration Management with Subversion, YAML and Perl Template Toolkit

Administrator's Guide

Release 5.0

Table of Contents

Preface	iii
1. Installation	1
1.1. Repository Server	1
1.2. Target Server	2

Preface

The key to successful systems administration is reproducibility. If we can reproduce the current state of a system, then we can restore service in the event of a disaster and can rapidly roll out additional server capacity to meet growth in demand for services. Very often, server installation and configuration has to be performed by a systems administrator working under extreme pressure—especially in a disaster recovery situation. Doing this accurately and quickly implies advance planning and a degree of automation. The "configtool" toolkit is built from familiar free software components: Template Toolkit for configuration file generation; Subversion for revision control; rsync for file distribution; Perl for scripting; YAML for data serialisation. We introduce each of these technologies and describe how they fit together to provide a modular and flexible system for managing configuration files.

The configtool web site [<http://users.ox.ac.uk/~raym/software/configuration-management/>] carries details on the latest release and other information to make your work or play with configtool. These script are written by Ray Miller, Systems Development And Support Section Manager Computing Services, University of Oxford.

1

Installation

1.1. Repository Server

You must install basic subversion server and define our platform. You must define your first server in REPOSITORY/systems/FIRST_SERVER directory. You must declare config.rb3 file :

```
# FIRST_SERVER control file
=/groups/linux-server/control.rb3
=/groups/postgres-server/control.rb3
!systems/FIRST_SERVER/params.yml
```

You must define your group, in example postgres-server group in REPOSITORY/groups/postgres-server/control.rb3 file :

```
# postgres-server control file
+/etc/pgsql/conf/postgresql.conf.prect groups/postgres-server/install-scripts/postgres.prect.tt
+/etc/pgsql/conf/postgresql.conf.postct groups/postgres-server/install-scripts/postgres.postct.tt
+/etc/pgsql/conf/postgresql.conf groups/postgres-server/src/etc/pgsql/conf/postgresql.conf.tt \
root:staff 0444
+/etc/pgsql/conf/pg_hba.conf.prect groups/postgres-server/install-scripts/postgres.prect.tt
+/etc/pgsql/conf/pg_hba.conf.postct groups/postgres-server/install-scripts/postgres.postct.tt
+/etc/pgsql/conf/pg_hba.conf groups/postgres-server/src/etc/pgsql/conf/pg_hba.conf.tt \
root:staff 0444
```

The first character of line in rb3 control file define the role.

- '=' include another control file
- '+' adds file for current system ('/' indicates continuation)
- '-' suppress a generated file
- '!' load an YAML parameter file

You must add pre and post install script (use extension *.prect or *.postct), and add template (*.tt) in example groups/postgres-server/src/etc/pgsql/conf/pg_hba.conf.tt :

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
```

```

host    all          all          127.0.0.1/32          trust
[% IF params.pg_hba_allow_from -%]
[% FOREACH client = params.pg_hba_allow_from %]
host    all          all          [% client %]/32      password[% END %]
[% END -%]
[% IF params.pg_hba_trust_allow_from -%]
[% FOREACH client = params.pg_hba_trust_allow_from %]
host    all          all          [% client %]/32      trust[% END %]
[% END -%]
# IPv6 local connections:
host    all          all          ::1/128              trust

```

You must define values of variables used in template files in `REPOSITORY/systems/FIRST_SERVER/params.yml` (in YML language) :

```

---
eth0_address: 192.168.11.11
pg_hba_allow_from:
- 192.168.11.11

```

Create subversion post-commit hooks in order to export repository content in dedicated directory. This directory is required for synchronize target server with `rsync` command.

```

TODO : add hooks

```

You must create publish directory :

```

$> mkdir /var/local/PUBLISH_DIR

```

After define all server you can generate final configuration files and commit on the repository :

```

$> ./rb3.pl --build systems/FIRST_SERVER (or for all ./rb3.pl --build systems/*)
$> svn add *
$> svn commit

```

1.2. Target Server

You must add configuration file in `/etc/configtool.conf` in order to define global repository server :

```

$config{repository}          = "/var/local/configtool/";
$config{remote_sync_cmd}     = "/usr/bin/rsync";
$config{remote_sync_args}   = [qw(--recursive --quiet --times --perms --owner --group --compress --delete

```

You must create local repository directory and you must add `configtool.pl` script in `/usr/local/tools/platform` and in `$PATH`

```

$> mkdir /var/local/configtool

```

Synchronize your local repository configuration from global repository server :

```

$> configtool.pl --sync

```

Deploy configuration configuration stored in local repository :

```
$> configtool.pl
```