Installing a Quattor Server and Client

German Cancio

CERN/IT

Outline

- "Theory": Installing a Quattor server & client
  - CDB
  - SWRep
  - AII
  - client
- Practical exercise: Installing CDB

Quattor Server

- Requirements:
  - SL3 (including SLC3), or RH Linux 7.3
  - Disk: 2.5 GB for system, 2.5 GB per client OS, 5 GB for LCG-2
  - Will show steps for an SL 3 server
- Using new APT based installation
  - Could use plain RPM, or yum instead for bootstrapping.
- "Basic" Quattor Server installation, focussing on CDB
  - Basic Quattor services (CDB, SWRep, AII) on one node.
  - Advanced services (cdb2sql) not deployed.
  - Should be OK for O(100) nodes.
    - More nodes -> use HTTP proxies for SWRep and CDB XML profiles
- Practical exercise for CDB – one CDB per alumni pair

Setting up APT

- Use APT for server bootstrapping
  - Not documented (yet) in installation guide
  - Install quattor client, CDB, SWRep and AII RPM's
  - Server *configuration* still needs to be done manually!
- APT repository
  - http://quattorsw.web.cern.ch/quattorsw/software/quattor/apt
  - Binary and source repository
  - SL3 and RH73
  - Yum repository available as well at above location
- Setting APT repository configuration (SL3) on to-be-installed server

For binaries:

```
# echo 'rpm http://quattorsw.web.cern.ch/quattorsw/software/quattor apt/1.0.0/i386
quattor_sl303'
>/etc/apt/sources.list.d/quattor.list

# apt-get update
```

APT meta-packages for Quattor

- Available meta-packages:
  - **`quattor-client`**: install client packages (CCM, NCM + basic components, CDB CLI, SWRep CLI)
  - **`quattor-cdb`**: install Configuration Database (CDB) server
  - **`quattor-cdbsql`**: install the CDBSQL backend server
  - **`quattor-swrep`**: install the SPMA SW Repository (SWRep) server
  - **`quattor-aii`**: install the Automated Installation Infrastructure (AII)
- Installing a meta-package:
  - **`# apt-get install quattor-<meta>`**
    eg.
    `# apt-get install quattor-client`
- APT preference files (`/etc/apt/preferences`) may lead to conflict between package versions in Quattor repository and other repositories.
  - Quick & dirty workaround: remove that file
- We will describe how to install+configure CDB, SWRep, AII and the client.
- An exercise will be to install+configure yourself CDB, so **pay attention** ☺

1. Installing the Quattor client RPM's

- First of all, install the client software
  - Not everything strictly needed, but for convenience

```
# apt-get update
# apt-get install quattor-client
The following NEW packages will be installed:
   aii-client (1.0.11-1)
   ccm (1.4.0-1)
   cdb-cli (1.8.5-1)
   cdp-listend (1.0.0-1)
   ncm-accounts (2.0.4-1)
   ncm-ccm (0.0.5-1)
   ncm-cdispd (1.0.1-1)
   ncm-cdp (0.0.2-1)
   ncm-cron (1.0.7-1)
   perl-AppConfig-caf (1.3.7-1)
   perl-CAF (1.3.7-1)
   perl-LC (1.0.6-1)
...
   perl-Text-Glob (0.006-1)
   perl-TimeDate (2.22-1)
   quattor-client (1.0.0-5)
   swrep-client (1.2.31-1)
   swrep-libs (1.2.31-1)
0 upgraded, 29 newly installed, 0 removed and 0 not upgraded.
Need to get 378kB of archives.
After unpacking 557kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

2. Installing and configuring CDB

- Install first apache, then quattor-cdb
  - **`# apt-get install httpd mod_ssl`**
    - dependency bug, fixed in CVS :-(
  - **`# apt-get install quattor-cdb`**
    - Installs 7 new packages: cdb, libraries, PAN compiler
- Start up apache:
  - **`# chkconfig --add httpd`**
    `# /sbin/service httpd start`
- Initialise CDB:
  - A configuration file with default CDB options exists under `/etc/cdb.conf`, but no need to change it
  - The CDB initialisation has to be done explicitly by running:

`# cdb-setup`

2. Installing and configuring CDB (cont)

- Adding users to CDB:
  - `/etc/httpd/conf/cdb.allow` - who is allowed to do what in CDB

```
# cat /etc/httpd/conf/cdb.allow
napoleon    admin   -> full read-write access
pig         rw      -> limited read-write access
sheep               -> read-only access
```

- For every defined user, a password file entry has to be created
  - `/etc/httpd/conf/.passwd` - in user::password format
  - Crypted password can be generated running **`openssl passwd`** (literally!!)

```
# cat /etc/httpd/conf/.passwd
napoleon::tVlXsuVLw0zHs
pig::Jje7hhdz3poqr
sheep::WHjfr7320zqa
```

  - NOTE: CDB users don't have to exist as UNIX users!

2. Installing and configuring CDB (cont)

- Two clients available for accessing CDB:
  - **cdb-simple-cli** - for local access, limited functionality, *to be phased out*
  - **cdb-cli** - for local and remote access over HTTP(S), SOAP-based
- Setup of cdb-cli:
  - `/usr/bin/cdbop` : executable
  - `/etc/cdbop.conf` : config file. Needs to be created from scratch, or from template

```
# cp /usr/share/doc/cdb-cli-1.8.5/cdbop.conf /etc/cdbop.conf
# vi /etc/cdbop.conf
...
protocol = https                          <- http can also be used
server   = myserver.myfarm.org            <- name of the CDB server
...
```

  - A per-user config file overriding the above may be created under `~user/.cdbop.conf`

2. Installing and configuring CDB (cont)

- Run cdbop to check that the setup is OK:

```
$ cdbop
Quattor CDB CLI: version 1.8.5
Enter user-name: napoleon
Enter password:

Connecting to https://myserver.myfarm.org...
Welcome to CDB Command Line Interface
Type 'help' for more info

cdb>
cdb> open                                  <- open the session
[WARN] cannot list templates in CDB        <- OK as CDB still empty!
cdb> exit                                  <- quit the session
```

- CDB is transactional, multi-user, multi-session
- cdb-cli has also a batch mode for non-interactive bulk operations (see **`man cdb-cli`**)

2. Installing and configuring CDB (cont)

- **cdbop** is used very similar to an ftp client. Template files on the server are downloaded, processed, and then uploaded again.
  - command line history and edition supported, **as well as wildcards**
  - External commands launched with a '!' escape prefix
- A special command, '**commit**', validates and commits the modified and uploaded templates into CDB.
- Typical examples:

```
cdb>
cdb> get profile_lxb0xxx              -> copies template to the current working dir
[INFO] getting template: profile_lxb0615.tpl
cdb> ! emacs profile_lxb0xxx.tpl &    -> can start background apps
cdb> update profile_lxb0xxx.tpl       -> update server session copy
[INFO] updating template: profile_lxb0615
cdb> commit                           -> validate and commit changes
cdb> add my_template.tpl              -> add new template from
[INFO] adding template: my_template.tpl    local file system
cdb> delete old_template              -> remove template from
[INFO] removing template: old_template     server session (does not  delete
cdb>                                       local files)
```

2. Installing and configuring CDB (cont)

- The generated XML profiles can be consulted at:
  http://myserver/profiles/ resp. **`/var/www/html/profiles`**
- Other cdbop commands:
  - **`rollback`** - undo all 'updates' added to the session since the last commit
  - **`list <regexp>`** - list files in CDB (use tabulator as in bash as well!)
  - **`help`** - list all commands.
  - **`help <command>`** - command specific help
- Typical CDB operation errors:
  - Try to add an already existing template:
    `[WARN] template 'xyz' already exists, please use 'update' option`
  - Try to delete or list a non-existing template:
    `[WARN] template(s) matching 'xyz' not found`
  - Syntax error in template (after running "add" or "update"):
    `[ERROR] Server: *** … syntax error … <error><template><line>`
  - Transaction validation error (after running "commit"):
    `[ERROR] Server: *** … pan error …`
    `[WARN] cannot make the commit`

2. Installing and configuring CDB (cont)

- CDB bug chasing - check the log files:
  - **`/var/log/cdb.log`** *-> CDB operations*
  - **`/var/log/soap/cdbsoap.log`** *-> CDB SOAP communications*
  - **`/var/log/httpd/[ssl_]error.log`** *-> SOAP server exceptions*
  - **`/var/log/httpd/[ssl_]access.log`** *-> SOAP requests*
- Typical CDB setup issues:
  - **`500 Internal Server Error`**
    - /etc/httpd/conf/.passwd incorrectly set or wrong permissions
    - /etc/httpd/conf/cdb.conf " " " "
    - Otherwise check log files above
  - **`500 Connect failed: Connection refused`**
    - Web server down
    - HTTPS configured but not supported (install `mod_ssl` RPM, restart httpd!)

3. Installing the SW Repository

- The SW repository is only required if SPMA is going to be used as package manager.
  - It can be skipped if other tools (apt, yum, up2date) are used for managing software (see talk on Friday).
  - Even if SPMA is used, a normal web server could replace it. However, SWRep provides enforced access control and consistency checks for RPM repositories.
- SWRep can be installed on the same or on a different server as CDB.
- Bootstrap installation with APT:

```
# apt-get install quattor-swrep
```

-> Installs 2 new packages (if CDB is installed on that node, otherwise more)

3. Installing the SW Repository (cont)

- Copy and edit the SWRep server config template:

```
# cp /usr/share/doc/swrep-server-1.2.31/swrep-server.conf /etc/swrep/
# vi /etc/swrep/swrep-server.conf
...
name     = "my repository"                    <- logical repository name
owner    = napoleon@myfarm.org                <- e-mail of responsible
url      = http://myserver.myfarm.org/swrep   <- URL prefix
Rootdir  = /var/www/html/swrep                <- root dir on local file system
...
```

Note: 'url' and 'rootdir' need to point to the same resulting location!

- **SWRep client** setup**:**
  - `/usr/bin/swrep-client`: executable
  - `/etc/swrep/swrep-client.conf`: config file. Created from scratch or template. A per-user `~user/.swrep-client.conf` can be created as well

```
# cp /usr/share/doc/swrep-client-1.2.31/swrep-client.conf /etc/swrep/
# vi /etc/swrep/swrep-client.conf
...
repository = swrep@myserver.myfarm.org      <- swrep is the default created user
...
```

3. Installing the SW Repository (cont)

SSH setup:

- Unlike CDB, SWRep uses an SSH based protocol as the management interface
  - SSH config is sometimes tricky!
  - A prototype SOAP based SWRep exists but not yet deployed
- Ensure that SSHD on the server has 'PermitUserEnvironment' set to 'yes':
  - `# cat /etc/ssh/sshd_config | grep PermitUserEnvironment`
    `PermitUserEnvironment yes`
  - Otherwise add it, and `/sbin/service sshd restart`

3. Installing the SW Repository (cont)

- For every user to be added, do the following:

1. create an SSH key if required (on the client)

```
$ ssh-keygen -t rsa
```

2. copy the public key **onto the server,** and add `'environment="SSH_USER=<user>'` at the beginning of the line

```
# cat /home/napoleon/.ssh/id_rsa.pub >>/var/swrep/.ssh/authorized_keys
# vi /var/swrep/.ssh/authorized_keys
...
environment="SSH_USER=napoleon" ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEA9QP
...
```

3. Add the user to the ACL file.

```
# vi /etc/swrep/swrep.acl
napoleon:/                    <- full access to all 'areas'
pig:/lcg,/egee/prototype      <- can only add/remove packages in these areas
sheep:/notexisting            <- read-only access
```

Note: 'areas' only have meaning in terms of Access Control. All packages belonging to a platform are stored in the same physical directory!

3. Installing the SW Repository (cont)

- Check that everything is OK:

```
$ swrep-client listrights
 You are napoleon, with rights to change packages with tags: /
 you have repository administrator rights
```

- Create platforms:

```
$ swrep-client addplatform i386_sl3
Platform i386_sl3 successfully added
$ swrep-client addplatform i386_redhat73
Platform i386_redhat73 successfully added
```

- Create areas:
  - Each area must be added per platform.
  - Sub-areas can be defined as well. /area1/area2/area3/…

```
$ swrep-client addarea i386_sl3 /release
Area /release successfully created in platform i386_sl3
$ swrep-client addarea i386_sl3 /lcg
Area /lcg successfully created in platform i386_sl3
```

3. Installing the SW Repository (cont)

- Upload packages:

Add package from local file system:

```
$ swrep-client put i386_sl3 foo-1-1.i386.rpm /release
Package … succesfully added to platform i386_sl3
```

Add package located on a remote web server:

```
$ swrep-client pull i386_sl3 http://server/foo-1-1.i386.rpm /release
Platform i386_redhat73 successfully added
```

- Upload packages - bulk mode:
  - Eg. A new distribution to be added - don't do this package by package!

1. On the server:

```
# cd /var/www/html/swrep/i36_sl3
# cp /mnt/cdrom/SL/i386/RPMS/* .
```

2. On the client:

```
$ swrep-client bootstrap i386_sl3 /release
```

- Do NOT add or remove packages by hand unless using bootstrap!
- Verify that the web server is accessible and the packages uploaded
  - http://myserver.mydomain.org/swrep

3. Installing the SW Repository (cont)

- Query the repository:

```
$ swrep-client find emacspeak    <- finds all containing 'emacspeak' in name
i386_sl3  emacspeak-17.0-4.i386.rpm
$ swrep-client query emacspeak-17.0-4.i386.rpm
Package tag: /base
Extra package information:
Name    : emacspeak                    Relocations: (not relocateable)
Version: 17.0                          Vendor      : FNAL
Release: 4                             Build date  : Wed Jun 2 23:48 …
```

- Generate PAN template with repository contents:

```
$ swrep-client template i386_sl3 >repository_myrep_i386_sl3.tpl
```

3. Installing the SW Repository (cont)

- Typical (SSH related) configuration problems:

/etc/security/limits.conf restrictive:

```
Too many logins for 'swrep'.
```

Wrong or incorrectly setup authorized_keys:

```
[ERROR] Access to the repository at swrep@server is denied.
```

Wrong SSH parameters leading to password prompt:

```
swrep@server password:
```

- If problems persist, check (and change) the default SSH parameters as provided in `/etc/swrep/swrep-client.conf`:

```
# cat /etc/swrep/swrep-client.conf | grep ssh-params
ssh-params = '-2 -q -a -x -T -e none'
```

- Try to connect using `ssh <parameters> -vvv swrep@<server>`
- Check the logfile on the server: **`/var/log/swrep-server.log`**
- Enable 'debug' and 'verbose' on the client and server:

```
$ swrep-client --verbose --debug 5 <myoperation>   <- on the client
```

on the server: edit the config file

```
# cat /etc/swrep/swrep-server.conf | grep debug
debug = 5
```

4. Installing the AII

- AII (Automated Installation Infrastructure) works on top of native RH/SL installer using PXE.
  - Anaconda/KickStart
- AII requires to run:
  - DHCP server (IP address + kernel location)
  - TFTP server (boot kernel)
  - HTTP server (OS images+packages)
- Bootstrap installation with APT (on same server as SWRep and CDB):

```
# apt-get install dhcp -> DHCP server (if not already installed)
# apt-get install quattor-aii
   -> Installs 5 packages
```

- No need to modify AII server config files for a standard setup
  - `/etc/aii-nbp.conf, /etc/aii-dhcp.conf, /etc/aii-osinstall.conf`

4. Installing the AII (cont)

- DHCP configuration:
  - If you install a new DHCP server: copy template, and fill / uncomment as indicated in the config file:

```
# cp /usr/share/doc/aii-1.0.11/eg/dhcp.conf /etc/
# vi /etc/dhcp.conf
...
ddns-update-style ad-hoc                    <- for ISC DHCP 3
option vendor-class-identifier                 (SL3 default)
...
For every subnet, create the following:
subnet xx.xx.xx.xx netmask xx.xx.xx.xx {
  option routers xx.xx.xx.xx;
  option domain-name-servers xx.xx.xx.xx;

}
```

  - Hosts will be added, updated and removed by AII scripts, but preserving the configuration information of hosts not managed by AII.

4. Installing the AII (cont)

- PXELinux configuration:
  - Copy Linux boot kernel into /osinstall/nbp/<os> (typically, initrd.img and vmlinuz)

```
# mkdir /osinstall/npb/sl3
# cp /mnt/cdrom/images/pxeboot/* /osinstall/npb/sl3
```

- KickStart configuration: *some steps could be further automated here…*
  - KS files **generated by AII** will be stored at /osinstall/ks. Make this directory available to your web server area:

```
# ln -s /osinstall/ks /var/www/html/ks
```

  - Then, copy the SL3 installation CDROM onto the web server area:

```
# cp -r /mnt/cdrom/* /var/www/html/sl3
```

  - Copy acknowledge script (avoids endless reinstalls) to the cgi location:

```
# cp /usr/bin/aii-installack.cgi /var/www/cgi-bin
```

  - Allow user 'apache' to run AII commands:

```
# cat /etc/sudoers | grep aii-shellfe
apache myserver=(ALL) NOPASSWD: /ust/sbin/aii-shellfe
```

4. Installing the AII (cont)

- CDB configuration:
  - Update the CDB template pro_software_component_aii.tpl
    - "server" and "cdb" parameters of 'osinstall' structure

- AII local management:
  - Edit the AII front-end config file (copy from template first)

```
# cp /usr/share/doc/aii-1.0.11/eg/aii-shellfe.conf /etc/
# vi /etc/aii-shellfe.conf
cdburl = http://myserver.myfarm.org/profiles  <- CDB profiles URL
```

  - Add/update AII configuration for node:

```
# aii-shellfe --configure node_name
```

  - Select a node for AII installation:

```
# aii-shellfe --install node_name   <- will reinstall at next reboot
# aii-shellfe --boot node_name      <- will boot from HD
```

  - Check status of a node:

```
# aii-shellfe --status node_name
```

4. Installing the AII (cont)

- More advanced AII management *not described here*
  - See installation guide chapter 7.2
  - Leave this as homework :-)
- AII remote management
  - via **`aii-installfe`** front-end (aii-client RPM)
  - Based on SSH remote login on the server
- AII automatic synchronization with CDB
  - Whenever a profile changes in CDB, synchronize AII:
    - New nodes -> generate KS file and mark for installation
    - Updated nodes -> regenerate KS file
    - Removed nodes -> remove KS file

4. Installing the AII (cont)

Common AII Problems: thanks to Cal Loomis

- PXE can't get installation parameters from DHCP server
  - DHCP server not configured correctly (check IP numbers, etc)
  - Firewall is blocking access to DHCP server.
  - Another DHCP server is responding.
- TFTP server can't be contacted
  - Check /etc/hosts.allow file
  - Verify all necc. Files are available in /osinstall dir.
  - Firewall is blocking access to FTFP server.
- KickStart fails
  - Check information in pro_software_component_aii.tpl:
  - Change "hda" to "sda" for SCSI interface.
  - Verify URL download location.

5. Install a Quattor client

- The quattor client installation is done automatically by AII. But it can be done manually as well (or hooked into another installer)
- Installation
  - eg. via `apt-get quattor-client` as shown before, or installing individual RPM's
- Configure CCM (Configuration Cache Manager) to point to the right server:

```
# cp /usr/share/doc/ccm-1.4.0/eg/ccm.conf /etc
# vi /etc/ccm.conf
...
profile  http://myserver.myfarm.org/profiles/profile_$host.xml
...
```

$host gets automatically expanded

- Initialise the CCM cache by running:

```
# ccm-initialise
```

- CCM is updated automatically by the installed cdp-listend daemon on every CDB node configuration update (see /var/log/messages)

```
# tail /var/log/messages
Jan 12 08:35:16 cdp-listend: Received UDP packet (ccm) from ...
Jan 12 08:35:16 cdp-listend: ccm-fetch will be called in X seconds
Jan 12 08:36:XX cdp-listend: Calling ccm-fetch (after X seconds)
```

5. Install a Quattor client (cont)

- CCM can be updated manually as well:

```
# ccm-fetch                  <- take default profile as defined in /etc/ccm.conf
# ccm-fetch --profile <url>  <- take another profile (useful for tests)
```

- Note that CCM handles error recovery if a webserver is down/not reachable.
- Query the current (cached) CCM profile with ncm-query:

```
$ ncm-query --dump /
[INFO] Subtree /
+-/
  +-hardware
    +-cards
      +-nic
        +-eth0
          $hwaddr: (string) '00:D0:B7:XX:XX:XX'
...
$ ncm-query --dump /system/kernel/version
[INFO] Subtree /system/kernel/version
$ version : (string) '2.4.21-20.0.1.EL.cernsmp'
```

- The NCM framework and the SPMA SW packager manager will be covered in the next part of the tutorial (Friday).

Practical Exercise: Install a CDB

Exercise: Installing CDB

- References:
  - This slides
  - Quattor installation guide (shouldn't be needed)
- Machines and users:
  - A node for every student pair (lxb06XX)
  - Two users defined: 'root' and 'tutorial'. Use root only whenever necessary. User 'tutorial' has a swrep-client access preconfigured onto a central server (not needed…yet!)
    - Passwords: none configured for user 'tutorial'. See whiteboard for 'root'.
    - Note: your CDB installation will be used for ulterior exercises.

Exercise: Installing CDB

- Part 1 (*easy*):
  - Go and install / configure a CDB server and the CDB client (on the same node!)
  - Some RPM installs and configuration steps are already done. But not everything, so please check carefully!
- Part 2 (*more challenging.. In particular for those who didn't do the homework!*):
  - A) Populate the CDB with the following set of templates: **/afs/cern.ch/user/g/gcancio/public/cdb-templates-tutorial.tgz** and B) create a configuration profile for **your** node.
  - Hints:
  - Create a temporary directory where you unwind the tarball and run cdbop from there
  - Use the wildcard option for the cdbop 'add' command (add *.tpl)
  - Rename the node template "profile_lxb0xxx" to your machine name, and replace IP, hostname, and MAC address (get all this from 'ifconfig')
  - Check that the template name (first line in file) corresponds to the file name.
  - Do not forget to update modified templates in CDB with 'update', and then run 'commit'.
  - Once the new profile is committed, use the ncm-query command to verify that your profile is now OK. (ncm-query --dump /system/network)

quattor

http://quattor.org