

Manage versioning & reproducibility

This paper will summarize the points that have been the subject of the thesis and that led to the implementation of the solution that you find in the attachment.

In particular will be discussed about:

- **Versioning:**
 1. the installation process
 2. the reports of the versions of the installed software
 3. utility commands
- **Reproducibility:**
 1. comparison between two different machines
 2. the replication of an existing environment

Versioning

1. the installation process

tarball download

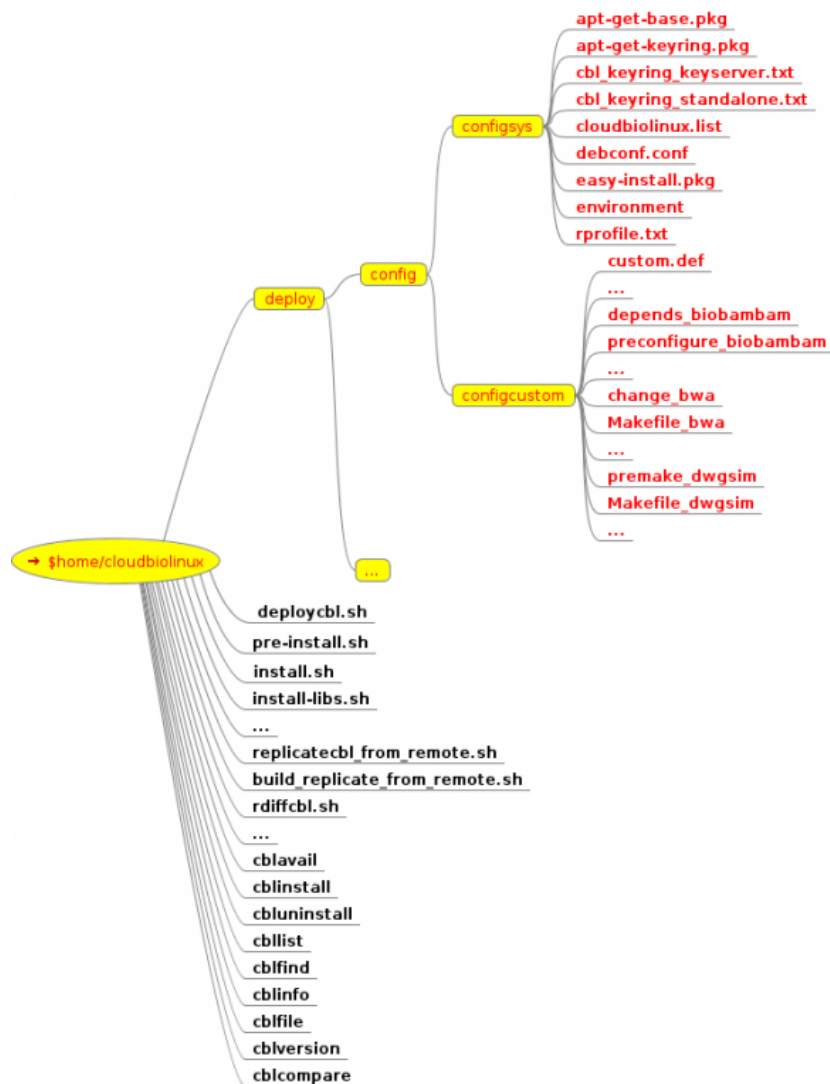
```
wget http://<sitecbl>/cloudbiolinux.tar.gz
```

extraction of the scripts and configuration files

```
tar xfvz cloudbiolinux.tar.gz && cd cloudbiolinux/
```

scheme

The following diagram shows the directory structure of the operating environment after extraction:



The information contained in the scheme are not exhaustive, but they only meant to highlight the elements of the environment and their location in the directory structure, and in particular:

- `./cloudbiolinux`: is the directory where there are the scripts
- `./cloudbiolinux/deploy/config/configsys`: is the directory where there are the configuration information of the environment for the **pre-installation** phase
- `./cloudbiolinux/deploy/config/configcustom`: is the directory where there are the configuration information for the installing of the **custom-packages**

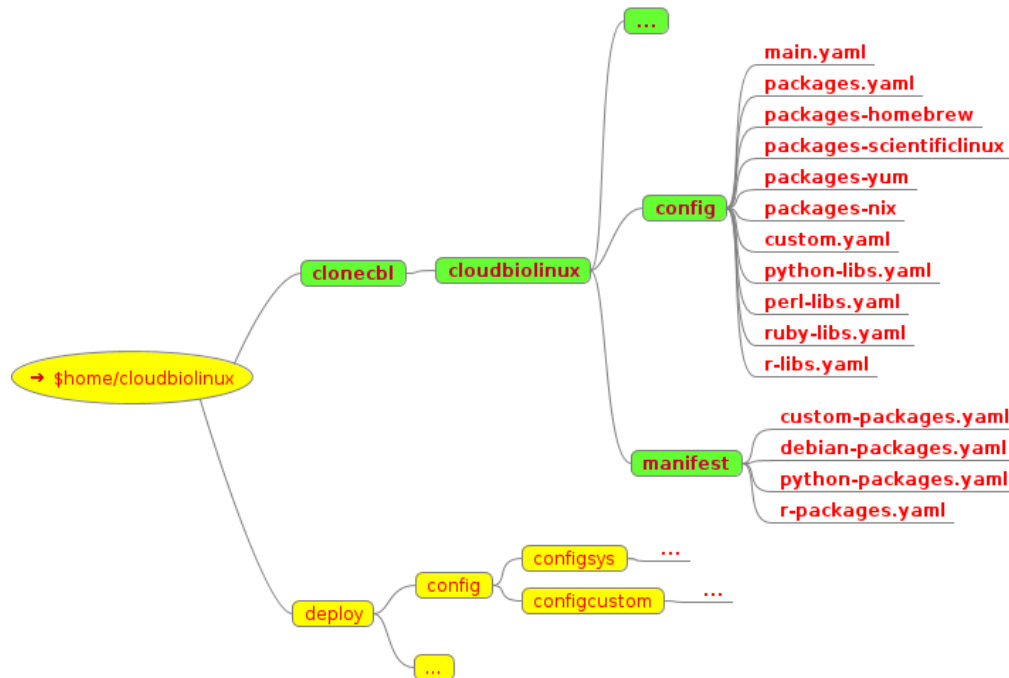
deploy

```
./deploycbl.sh
```

This script is composed of three specific phases:

1. ./pre-install.sh

This is the phase where the operating environment is prepared for the deployment, and some checks are made to verify that there are no problems during deployment. The following diagram shows the structure of the environment at the end of this phase:

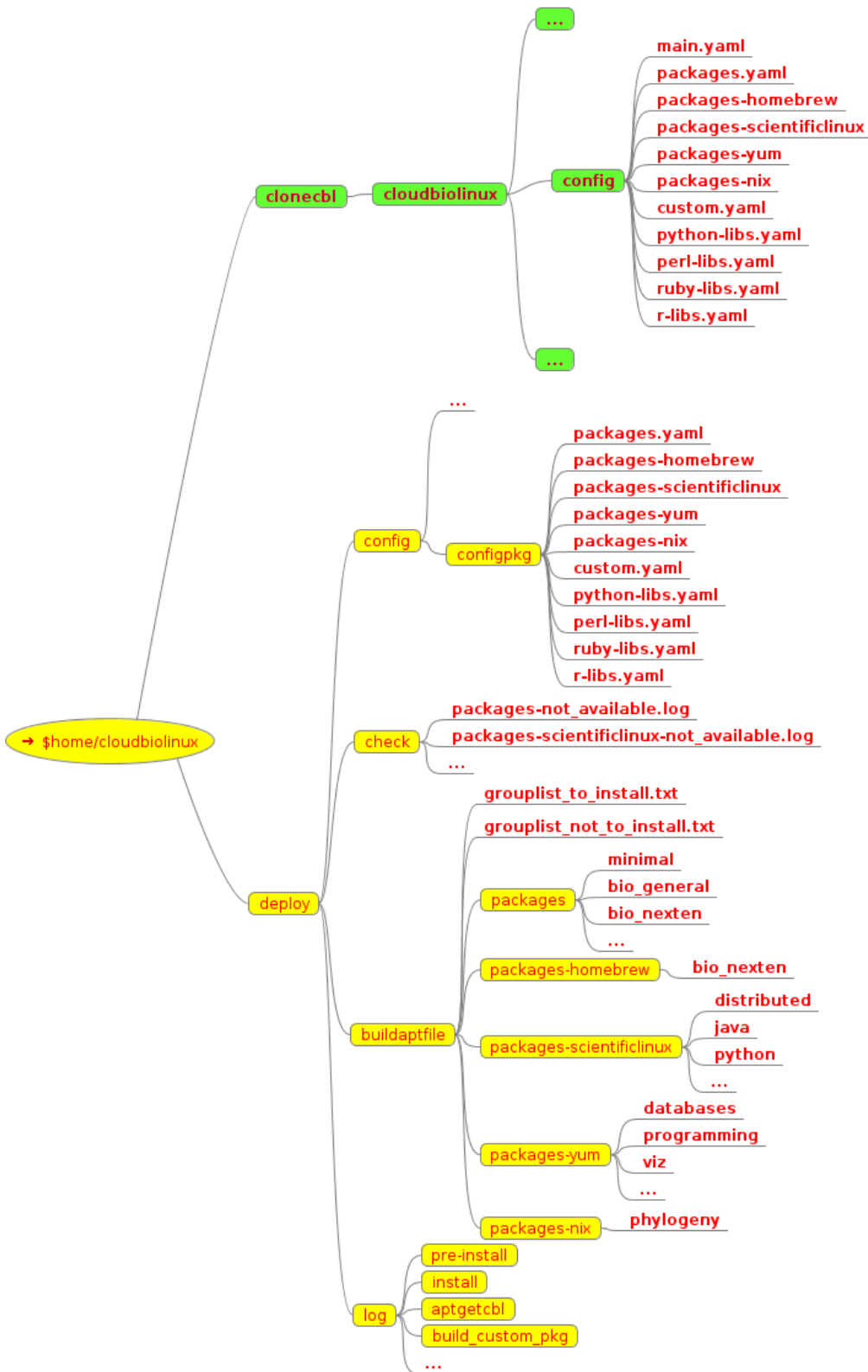


In the directory `./cloudbiolinux/clonecbl/` there are the informations downloaded from [git://github.com/chapmanb/cloudbiolinux.git](https://github.com/chapmanb/cloudbiolinux.git) [git://github.com/chapmanb/cloudbiolinux.git] ed in particular:

- `./cloudbiolinux/clonecbl/cloudbiolinux/config/`: it contains the informations needed in the phase of installing of the packages and libraries
- `./cloudbiolinux/clonecbl/cloudbiolinux/manifest/`: it contains the informations needed in the phase of checks of installed software

2. ./install.sh

Is the phase of installing of the packages (base/general and custom), that is executed only if the pre-installing phase has been successfully completed. The following diagram shows the structure of the **deployment** environment:



We can observe in particular:

- the directory `./cloudbiolinux/deploy/config/configpkg/`: it contains the list of the original packages (shown in the phase of **pre-install**) and that are actually available for the installation;
- the directory `./cloudbiolinux/deploy/check/`: it contains the list of packages not-available;
- the directory `./cloudbiolinux/deploy/buildaptfile/`: it contains the configuration information for the installation process;
 - in particular the file `group_to_install.txt` contains the list of the packages groups (listed in `main.yaml`) that must be installed;
- the directory `./cloudbiolinux/deploy/log/`: it contains the log files of all phase of installing process.

In the process of installing of the **custom-packages**:

- dependency management is done automatically through the use of **auto-apt** package;
- the custom-packages are converted in **.deb** format through the use of **checkinstall** package, and they are installed using **dpkg**
- have been converted about 50 packages (using as a reference the script `bio-nextgen`)

3. ./install-libs.sh

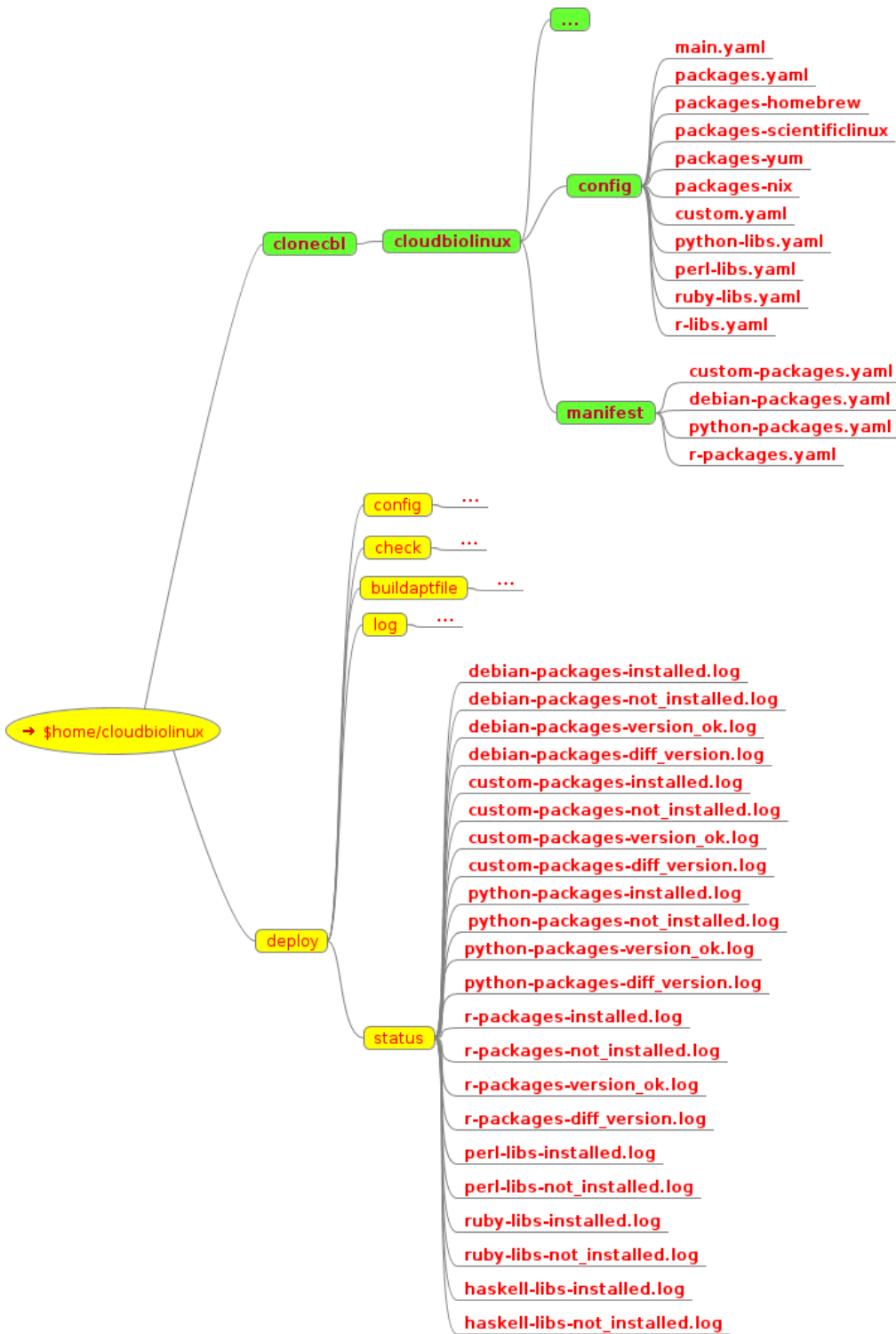
Is the phase of the installing of the libraries (essentially identical to the process of the packages installation).

2. reports of the installed software

At the end of installation process is generated a reporting of the packages and libraries installed (comparing, whenever possible, with the list of packages included in the **manifest**):

- **packages:**
 - **debian-packages-installed.log:** packages-base successfully installed
 - **debian-packages-not_installed.log:** packages-base not installed
 - **debian-packages-version_ok.log:** packages-base successfully installed with right version
 - **debian-packages-diff_version.log:** packages-base successfully installed with different version
 - **custom-packages-installed.log:** packages-custom successfully installed
 - **custom-packages-not_installed.log:** packages-custom not installed
 - **custom-packages-diff_version.log:** packages-custom successfully installed with right version
 - **custom-packages-version_ok.log:** packages-custom successfully installed with different version
- **libraries:**
 - **python-packages-installed.log:** python libraries successfully installed
 - **python-packages-not_installed.log:** python libraries not installed
 - **python-packages-version_ok.log:** python libraries successfully installed with right version
 - **python-packages-diff_version.log:** python libraries successfully installed with different version
 - **r-packages-installed.log:** R libraries successfully installed
 - **r-packages-not_installed.log:** R libraries not installed
 - **r-packages-version_ok.log:** R libraries successfully installed with right version
 - **r-packages-diff_version.log:** R libraries successfully installed with different version
 - **perl-libs-installed.log:** perl libraries successfully installed
 - **perl-libs-not_installed.log:** perl libraries not installed
 - **ruby-libs-installed.log:** ruby libraries successfully installed
 - **ruby-libs-not_installed.log:** ruby libraries not installed
 - **haskell-libs-installed.log:** haskell libraries successfully installed
 - **haskell-libs-not_installed.log:** haskell libraries not installed

The reports are stored in the directory `./cloudbiolinux/deploy/status/`, as shown in the scheme shown below:



3. utility commands

These are the commands that generalize the execution of the common operations (where implemented) performed on different object (packages and libraries). The sample syntax is as follows:

```
Syntax: ./cblavail <lib/pkg>name -l type [-D debug] [-v verbose]
lib/pkg-name : libraries/packages to analyze
type         : type of packages/libraries: packages, customcbl, python-libs, perl-libs, ruby-libs, r-libs, haskell-libs nix
[-D <debug>] : set debug
[-v verbose] : set verbose
```

Below you can see the list of commands:

cblavail

This command verifies what are the packages and libraries available (in the repository):

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cblavail tcpdump -l packages
apt-cache show tcpdump
Package: tcpdump
```

```

Priority: standard
Section: net
Installed-Size: 908
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Romain Francoise <rfrancoise@debian.org>
Architecture: i386
Version: 4.2.1-lubuntu2
Depends: libc6 (>= 2.7), libpcap0.8 (>= 1.0.0), libssl1.0.0 (>= 1.0.0)
Suggests: apparmor (>= 2.3)
Filename: pool/main/t/tcpdump/tcpdump_4.2.1-lubuntu2_i386.deb
Size: 383888
MD5sum: 3cf13729905286ec3e69ffa123b9b401
SHA1: 7a562edc67a685f8ba74970256d6a66c90090472
SHA256: edb2c4d93cb5415fa501596e37adaef79365722129cefd0275fafd991614807f
Description-en: command-line network traffic analyzer
 This program allows you to dump the traffic on a network. tcpdump
 is able to examine IPv4, ICMPv4, IPv6, ICMPv6, UDP, TCP, SNMP, AFS
 BGP, RIP, PIM, DVMRP, IGMP, SMB, OSPF, NFS and many other packet
 types.

 It can be used to print out the headers of packets on a network
 interface, filter packets that match a certain expression. You can
 use this tool to track down network problems, to detect attacks
 or to monitor network activities.
Homepage: http://www.tcpdump.org/
Description-md5: f01841bfda357d116d7ff7b7a47e8782
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Origin: Ubuntu
Supported: 5y
Task: standard, kubuntu-active

<tcpdump> found

```

```

piero@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cblavail cake -l ruby
gem query -r | awk -F " " '{ print ruby }' | grep -x cake
cake
<cake> found

```

cblinstall

This command installs a package or library:

```

piero@ubuntu1204-tesi-ivaptget:~/cloudbiolinux$ sudo ./cblinstall bmon -l packages
apt-get -s install bmon
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
Lettura informazioni sullo stato... Fatto
I seguenti pacchetti NUOVI saranno installati:
 bmon
0 aggiornati, 1 installati, 0 da rimuovere e 243 non aggiornati.
3 non completamente installati o rimossi.
Inst bmon (2.0.1-3 Ubuntu:12.04/precise [i386])
Conf man-db (2.6.1-2ubuntu1 Ubuntu:12.04/precise-updates [i386])
Conf bamfdaemon (0.2.126-0ubuntu1 Ubuntu:12.04/precise-updates [i386])
Conf desktop-file-utils (0.20-0ubuntu3 Ubuntu:12.04/precise-updates [i386])
Conf bmon (2.0.1-3 Ubuntu:12.04/precise [i386])
apt-get install bmon
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
Lettura informazioni sullo stato... Fatto
I seguenti pacchetti NUOVI saranno installati:
 bmon
0 aggiornati, 1 installati, 0 da rimuovere e 243 non aggiornati.
3 non completamente installati o rimossi.
È necessario scaricare 43,6 kB di archivi.
Dopo quest'operazione, verranno occupati 184 kB di spazio su disco.
Scaricamento di:1 http://it.archive.ubuntu.com/ubuntu/ precise/universe bmon i386 2.0.1-3 [43,6 kB]
Recuperati 43,6 kB in 0s (85,7 kB/s)
Selezionato il pacchetto bmon non precedentemente selezionato.
(Lettura del database... 463729 file e directory attualmente installati.)
Estrazione di bmon (da ../archives/bmon_2.0.1-3_i386.deb)...
Elaborazione dei trigger per gnome-menus...
Configurazione di man-db (2.6.1-2ubuntu1)...
Updating database of manual pages ...
Configurazione di bamfdaemon (0.2.126-0ubuntu1)...
Rebuilding /usr/share/applications/bamf.index...
Configurazione di desktop-file-utils (0.20-0ubuntu3)...
Configurazione di bmon (2.0.1-3)...

```

```

piero@ubuntu1204-tesi-install4:~/cloudbiolinux$ sudo ./cblinstall cheat -l ruby
gem install: no dry-run ...
gem install cheat
Entering auto-apt mode: gem install cheat
Exit the command to leave auto-apt mode.

Fetching: cheat-1.3.3.gem (100%)
Successfully installed cheat-1.3.3
1 gem installed
Installing ri documentation for cheat-1.3.3...
Building YARD (yri) index for cheat-1.3.3...
Installing RDoc documentation for cheat-1.3.3...

```

cbluninstall

This command uninstalls a package or library:

```

piero@ubuntu1204-tesi-ivaptget:~/cloudbiolinux$ sudo ./cbluninstall bmon -l packages
dpkg -P bmon
(Lettura del database... 463740 file e directory attualmente installati.)
Rimozione di bmon...
Eliminazione dei file di configurazione di bmon...
Elaborazione dei trigger per desktop-file-utils...
Elaborazione dei trigger per bamfdaemon...
Rebuilding /usr/share/applications/bamf.index...
Elaborazione dei trigger per gnome-menus...
Elaborazione dei trigger per man-db...

```

```

piero@ubuntu1204-tesi-install4:~/cloudbiolinux$ sudo ./cbluninstall cheat -l ruby

```

```
gem uninstall cheat
Remove executables:
  cheat

in addition to the gem? [Yn] Y
Removing cheat
Successfully uninstalled cheat-1.3.3
```

cbllist

This command lists the packages and libraries installed:

```
piro@ubuntu1204-tesi-replicate5:~/cloudbiolinux$ ./cbllist -l customcbl
ii abyss                1.3.4-1          customcbl - Assembly By Short Sequences - a de novo, parallel, paired-end sequence
ii bamtools             3fe66b9-1       customcbl - bamtools
ii bamutil              1.0.7-1         customcbl - bamutil
ii bcbio-variation     0.1.0-1         customcbl - bcbio-variation
ii bfast                0.7.0a-0.7.0   customcbl - BFAST: Blat-like Fast Accurate Search Tool.
ii biobambam           0.0.92-e325743 customcbl - biobambam
ii boost                0-1             customcbl - boost
...
```

cbllfind

This command checks if a package or library is installed:

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cbllfind tcpdump -l packages
tcpdump 4.2.1-lubuntu2
<tcpdump> found
```

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cbllfind cake -l ruby
*** LOCAL GEMS ***
cake (0.6.3)
<cake> found
```

cbllinfo

This command shows detailed information about a specific package or library:

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cbllinfo tabix -l customcbl
dpkg -s tabix
Package: tabix
Status: install ok installed
Priority: extra
Section: customcbl
Installed-Size: 172
Maintainer: root@ubuntu1204-tesi-install4
Architecture: i386
Version: 0.2.6-1
Provides: tabix
Description: customcbl - tabix
<tabix> found
```

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cbllinfo bzip2 -l python
pip show bzip2
---
Name: bzip2
Version: 2.5.1
Location: /usr/lib/python2.7/dist-packages
Requires:
<bzip2> found
```

cbllfile

This command shows the list of the files that are parts of a specific package or library and the installation path:

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cbllfile tabix -l customcbl
dpkg -L tabix
/.
/usr
/usr/share
/usr/share/doc
/usr/share/doc/tabix
/usr/share/doc/tabix/NEWS
/usr/share/doc/tabix/ChangeLog
/opt
/opt/CBL
/opt/CBL/bin
/opt/CBL/bin/tabix
/opt/CBL/bin/bgzip
/opt/CBL/bin/tabix.py
/opt/CBL/bin/test.py
<tabix> found
```

cbllversion

This command shows the version of a specific package or library:

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cbllversion trinity -l customcbl
<trinity: 0r2012-10-05-1> found
```

```
piro@ubuntu1204-tesi-install4:~/cloudbiolinux$ ./cbllversion tagsoup -l haskell
<tagsoup: 0.13.1> found
```

cblcompare

This command compare the the version of a specific package o library with the information contained in the **manifest** files:

```

piro@ubuntu1204-tesi-ivaptget:~/cloudbiolinux$ ./cblcompare blast2 -l packages
blast2 1:2.2.25.20110713-3ubuntu2
compare --> blast2 - (M)1:2.2.25.20110713-3ubuntu2 - (I)1:2.2.25.20110713-3ubuntu2
<blast2> found
  
```

```

piro@ubuntu1204-tesi-ivaptget:~/cloudbiolinux$ ./cblcompare yolk -l python
Name: yolk
Location: /usr/local/lib/python2.7/dist-packages/yolk-0.4.3-py2.7.egg
compare --> yolk - (M)0.4.3 - (I)0.4.3
<yolk> found
  
```

reproducibility

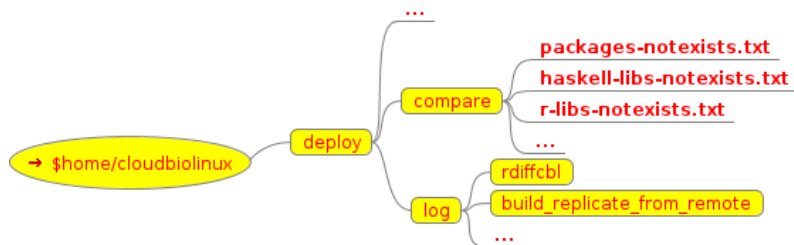
1. comparison between two different machines:

You can compare two different machines through simple commands:

comparison of a specific type of package or library

```
./rdiffcbl.sh -p <type-pkgs> -i <ip>
```

The file with the difference is stored in the directory `./cloudbiolinux/deploy/compare/`:

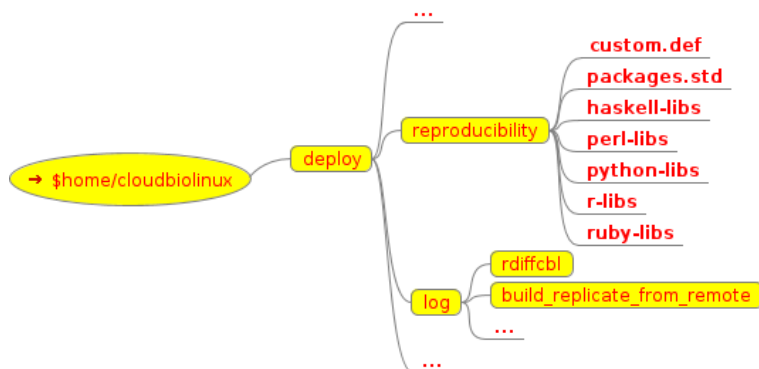


Every files contains the packages/libraries that are not installed on local machine (but are installed on remote machine).

full comparison

```
./build_replicate_from_remote.sh -i <ip>
```

This is a step of the **replicate** process, and **all packages/libraries** are stored in the directory `./cloudbiolinux/deploy/reproducibility/` :



2. the replication of an existing environment

tarball download

```
wget http://<sitocbl>/cloudbiolinux.tar.gz
```

extraction of the scripts and configuration files

```
tar xfvz cloudbiolinux.tar.gz && cd cloudbiolinux/
```

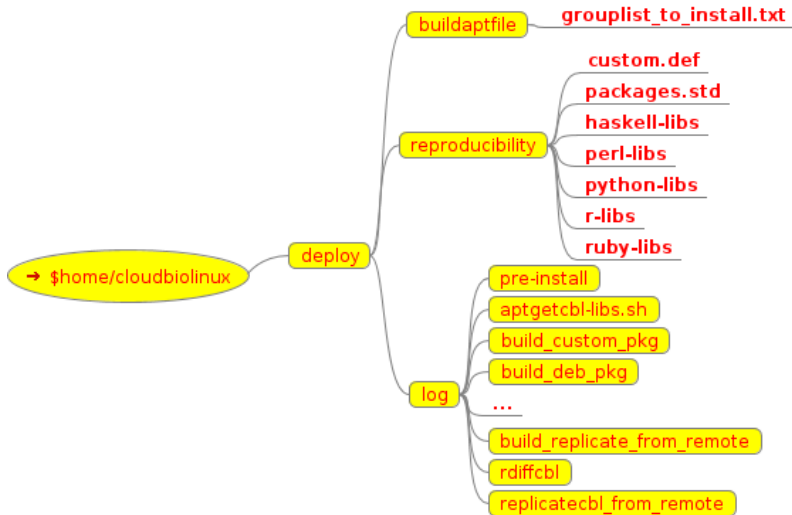
deploy


```
./replicatecbl_from_remote.sh -i <ip>
```

This script is composed of four specific phases:

1. **./pre-install.sh**: it is the phase where the operating environment is preparing for the deployment
2. **./build_replicate_from_remote.sh**: it is the phase where you retrieve the informations on packages and libraries installed on remote system, and they are compared with those installed on local system
3. **./build_custom_pkgs.sh**: it is the phase of the installation of custom-packages
4. **./aptgetcbl-lib.sh**: it is the phase of the installing of the packages and libraries that are installed on remote system

The following diagram shows the structure of the environment at the end of this process:



We can observe in particular:

- the directory **./cloudbiolinux/deploy/buildaptfile/**: in this process there is only the file **group_to_install.txt** that contains the list of the packages groups that must be installed
- the directory **./cloudbiolinux/deploy/reproducibility/**: it contains the lists of packages and libraries to install (retrieved from remote system)

/usr/share/dokuwiki/data/pages/univ/tesi/attivita/gestione_finale/detail.txt - Ultima modifica: 2014/06/30 00:36 da admin

Ad eccezione da dove è diversamente indicato, il contenuto di questo wiki è sotto la seguente licenza: Public Domain [<http://creativecommons.org/licenses/publicdomain/>]