

# Massive scale installation and management tools for Debian GNU / Linux (part 2)

Author: André Felipe Machado<andremachado@techforce.com.br>

Having 50 thousand machines across the country to install and manage is not a trivial task.

This is the second part of this text, focused on small to medium scale installations and management.

Many ~~references~~ are listed on the [part 1](#) [part 3](#)

Registered users can post comments below the article.

## Small to medium scale installations and management

The installation of, say, 50 machines can be accomplished with the simpler tools available, without needing to deal with the complexities and learning curve of the more powerful tools.

Actually, these simpler Debian tools can be used as **resources** by some of the more powerful ones and by each other.

This way, you will not discard the knowledge when you need to advance to the next level.

## Jablicator

The simplest one is a real gem already: [Jablicator](#)

Jablicator can create a meta-package containing dependencies of packages from a given machine and the repositories from where to download them.

When you try to install such created meta-package at another machine, it will download and install the packages needed to replicate the reference machine, with some limitations.

Alone, it can not decide about configuration options that one should answer during installation. This can be addressed by [using of debian preseeding](#) [preseeding](#) we will list ahead.

Also, alone [it cannot remove packages to clean machines](#). This can be addressed by [pkgsync](#)

## Pkgsync

[pkgsync](#) [4] is another very useful tool.

A bit more complex than Jablicator, it is more powerful, being able to remove packages in order to mostly synchronize machines.

Its configuration files could be distributed to the suitable machines using simple manual methods or by using other tools for distributing files, like cfengine, rdist, puppet, bcfg2, csync2 and others.

Like Jablicator, it cannot answer the configuration options during installation automatically, depending at other tools for this task. Debconf preseeding and or debian-installer preseeding, that we will list ahead.

## Debian Installer and debconf preseeding

These tools are being worked for years and now are official part of Debian stable.

You could use files containing answers for the questions asked during installation, reducing human interaction.

Read [\[25\]](#), [\[26\]](#), [\[27\]](#) and some tutorials like [this](#) [this](#) [this](#)

You could use [\[28\]](#) for the user preferences to [get selections](#) machine and create a file with configurations to load into the derivative ones.

These tools can be used to [\[29\]](#) create a custom Debian distribution using [\[30\]](#) [\[31\]](#) and some [cdd tools](#) [this](#) [this](#) [instructions](#)

You could, also, create something similar to a CDD (despite more limited in features and scalability) [\[32\]](#) and much more easier, using an on-line service like [Instalinux](#)

But these are expensive and labourious approaches, suitable for some environments but not at the scenario we are evaluating.

You should learn Debian Installer and debconf preseeding as they will be very useful even when used with other tools for large scale installations and deployments.

The preseeding are powerful methods, but have a few limitations [\[35\]](#) yet (maybe not in versions newer than [\[36\]](#) [\[37\]](#)) These limitations are [addressed](#) [FAI](#)

## IsiSetup

[IsiSetup](#) [69] is a very interesting tool with a very clever approach for managing configuration files.

It uses a revision control system tool (a Source Code Management) to track and manage configuration files.

As of ~~07~~ November 2007, it chose [Git](#)

So, you can rollback changes, view history, merge changes, fork changes, as you can do with other source files.

By using a distributed SCM like Git, every machine at your location could be under configuration management.

It is a clever approach for managing configuration files.

As so, it is limited in scope, not including installation, and suitable for small to medium scale deployments.

Used with the other tools it is a reasonable solution for many situations and scales.

It is very worth to consider this tool, as you can still use it with other tools as your site grow.

## TakTuk and Kanif

[TakTuk](#) [70] is (as its site explains well) "a tool for deploying parallel remote executions of commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and sets up an interconnection network to transport commands and perform I/Os

multiplexing/demultiplexing."

[Kanif](#) [71] is a wrapper tool around TakTuk for cluster management and administration. It combines main features of well known cluster management tools such as c3, pdsh and dsh and mimics their syntax. For the effective cluster management it relies on TakTuk, a tool for large scale remote execution deployment.

Both tools are focused at clusters, but still useful for massive scale remote parallel execution.

You should keep them at your sysadmin toolbox.

## EtcKeeper

IsiSetup has [73](#) [IsiSetup](#) [addresses](#) [these](#) [shortcomings](#) [with](#) [EtcKeeper](#)

"EtcKeeper is a collection of tools to let /etc be stored in a git repository.

It hooks into apt to automatically commit changes made to /etc during package upgrades.

It uses metastore to track file metadata that git does not normally support, but that is important for /etc, such as the permissions of "/ etc / shadow".

It's quite modular and configurable, while also being simple to use if you understand the basics of working with git."

So, we will stress this new tool.

At the next part of this text, we will start to evaluate the tools for massive scale installations and management.

## References:

The remaining references are listed on the parts 1 and 3 of this text.

[26] <http://www.debian-administration.org/articles/394>

[27] <http://blogs.cae.tnitech.edu/mwr/2007/04/17/unattended-debian-installations-or-how-i-learned-to-stop-worrying-and-love-the-preseedcfg/>

[28] <http://users.telenet.be/mydotcom/howto/linux/automatic.htm>

[29] <http://users.telenet.be/mydotcom/howto/linux/autoinventory.htm>

[30] <http://wiki.debian.org/CustomDebian>

[31] <http://packages.debian.org/etch/cdd-common>

[32] <http://packages.debian.org/etch/cdd-dev>

[33] <http://packages.debian.org/etch/cdd-doc>

[34] <http://packages.debian.org/etch/dconf>

[35] <http://www.debian.org/releases/stable/i386/apbs01.html.en#preseed-limitations>

[36] <http://www.informatik.uni-koeln.de/fai/fai-guide/ch-instprocess.html#s-ipartition>

[37] [http://faiwiki.informatik.uni-koeln.de/index.php/Setup\\_harddisks\\_2](http://faiwiki.informatik.uni-koeln.de/index.php/Setup_harddisks_2)