

AI How to: System Update and Additional Software

Vítor Gouveia,
vitor.gouveia@cern.ch

IT-PES-PS

- System update for AI
- Add additional repositories
- Deploy additional software
- Questions??

- Quattor way

Typical process:

1. Update `osdateversion` in CDB for a cluster
2. Run `spma` on each node (typically with `wassh`) – this actually updates the RPMs. Trigger a reboot if necessary.
3. `Spma` is also run automatically when a node reboots

- AI way (SLC6 only)

Puppet typically will not update packages by itself. A specific module provides the following process:

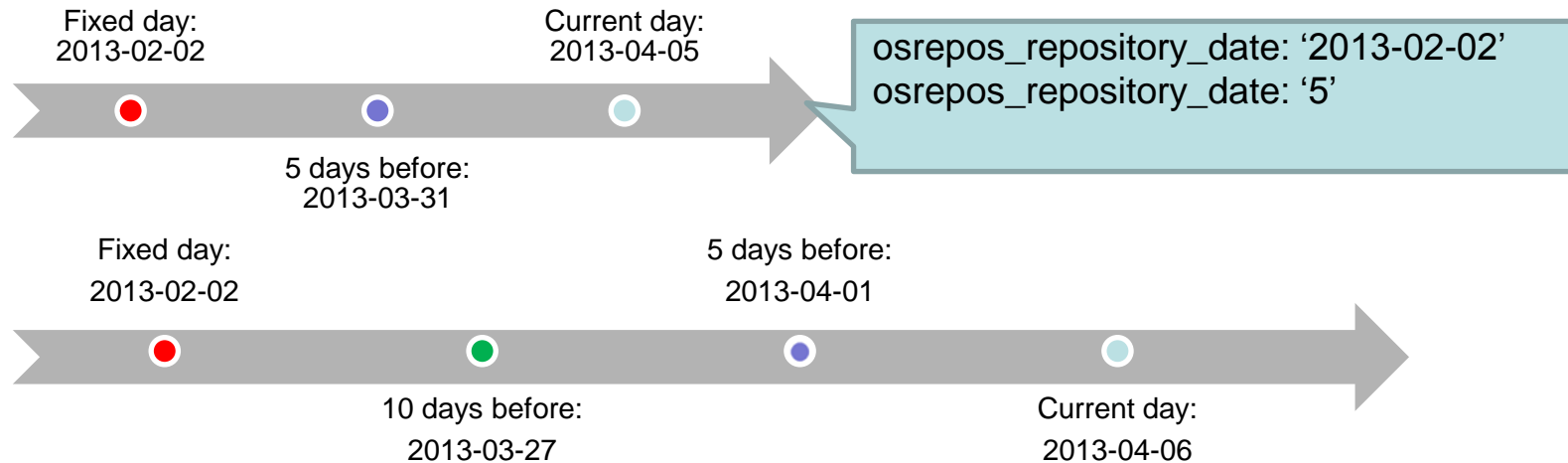
1. Update **hiera** variable **`os_repos_repository_date`**
2. A daily cron job will update the RPMs
3. If the VM needs a reboot (new kernel), this has to be done manually

osrepos:

- puppet module for repository control
 - one of the default modules of the AI
 - Installed before the **base** module
 - install the repositories **SLC**, **EPEL** and **AI**
 - two different behaviours according with the version of the OS (SLC6 or SLC5)
-
- On **slc6** machines the system is **keep up to date** with the **osrepos**
 - On **slc5** machines the system needs **manual intervention** to be updated

- yum distro-sync
 - synchronizes the installed package set with the latest packages available, this is done by either obsoleting, upgrading or downgrading as appropriate.
- cronjob **repos_sync**:
 - installed and executed once per day.
 - during the execution of the **distro_sync.sh** script:
 - yum distro-sync is executed
 - log file: **/var/log/distro_sync.log**

- Linuxsoft provides daily snapshots of the CERN SLC and EPEL repositories.
 - <http://linuxsoft.cern.ch/internal/yumsnapshot/yyyymmdd/...>
- How do we use them ?
 - Hiera variable **osrepos_repository_date**
- **osrepos_repository_date** values:
 - osrepos_repository_date not set (**default value**)
 - the official CERN SLC and EPEL repositories will be used
 - very latest updates from SLC and EPEL repos will always be installed
 - snapshot date
 - format (yyyy-mm-dd)
 - integer X
 - The snapshot used will be the one that is X days before the current day



- Useful hiera variables:
 - osrepos_sync_exclude_packages
 - list of package to be exclude of the synchronization
 - osrepos_sync_include_packages
 - list of packages to synchronize, if this is not set all packages are synchronized.

- SLC5 does not have the command yum distro-sync ☹️
- If you want to update a node
 - You need to do it **manually**
e.g. **yum update**
the latest updates from the installed repositories will be installed
- At the moment, **osrepos_repository_date does not work** on slc5 machines
 - The official CERN repositories are always installed

- The resource **osrepos::ai121yumrepo** should be use to add **any** additional yum repositories, this includes **external repositories**
 - Nothing more a wrapper around the standard puppet yumrepo resource

```
osrepos::ai121yumrepo{"rpmforge":  
  baseurl => "http://apt.sw.be/redhat/el6/en/x86_64/rpmforge/",  
  descr   => "RPM forge repository",  
  enabled => 1,  
  gpgcheck => 0,  
}
```

- Using external repositories creates a number of issues:
 - machines without internet access will not be able to use them
 - dependency issues / glitches can arise
 - No control of what is updated and when

- **Whenever extra repositories are needed, they should be replicated to linuxsoft**
 - Don't set up repos yourself, just make sure the RPMs are available somewhere for linuxsoft to pick up
 - Linuxsoft will make sure the repos are both available to all CERN nodes (including technical network and other networks without internet access) and scale properly
 - This is the process used to replicate EPEL
- For custom/private software
 - The source to replicate in linuxsoft may be e.g. an AFS folder
 - Also, some repos are generated by Koji
 - E.g. the AI repo and some others
 - **BUT THIS IS NOT OFFERED AS A BUILD SERVICE**

- Requests for repo duplication to be done via linux.support line
 - Exact process is yet to be formalized...

- the ensure attribute of the puppet resource **package**:
 - present or installed
 - puppet will install the latest version of the package that it finds in the hosts configured software repos when the manifest first runs, but never updates it
 - latest
 - Puppet will install a package if absent and **upgrade the package to newer versions when they become available** (to be used carefully)
 - absent
 - If present remove the package
 - version string
 - If available install a specific package version

- What if?
 - use puppet to install a RPM package that is not present in a repository

```
package { "emacs-puppet":  
    source =>  
    "http://linuxsoft.cern.ch/rpmsforge/redhat/el6/en/x86_64/rpmsforge  
/RPMS/emacs-puppet-2.7.9-1.el6.rf.noarch.rpm",  
    provider => "rpm",  
}
```

- Source parameter should point to the file
local file (or on a network file system) or a URL
- Problems:
 - dependency issues
 - how to update...

