ECA – Linux Patch Procedure

|  |
| --- |
|  |
| * Version: 0.1 (Draft) * INTERNAL |
|  |

|  |  |
| --- | --- |
| Document control | |
|  |  |
|  |  |
| |  |  | | --- | --- | | **Author** | Steven Eyre | | **Owner** | Andrew Raison | | **Approved by** | Paul Gabriel | | **Approval date** |  | | **Effective from** |  | | **Next review date** |  | | **Source location** |  | | **Audience** | IT department | | |

|  |  |
| --- | --- |
| Change history | |
|  |  |
|  |  |
| |  |  |  |  | | --- | --- | --- | --- | | Version | Date | Amended by | Details | | 0.1 | 20/02/2020 | Steven Eyre | First draft | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |

Contents

[Introduction 3](#_Toc33692939)

[Overview 3](#_Toc33692940)

[Purpose/Scope 3](#_Toc33692941)

[Spacewalk Overview 4](#_Toc33692942)

[Assumptions 4](#_Toc33692943)

[Spacewalk Description 4](#_Toc33692944)

[Spacewalk Web Portal 5](#_Toc33692945)

[Spacewalk Software Channels 6](#_Toc33692946)

[Spacewalk clients 7](#_Toc33692947)

[Spacewalk client details: 8](#_Toc33692948)

[Patching Servers 10](#_Toc33692949)

[PATCH PROCESS 10](#_Toc33692950)

[MANUALLY 10](#_Toc33692951)

[Ansible-Playbook **Error! Bookmark not defined.**](#_Toc33692952)

[Related documents 13](#_Toc33692953)

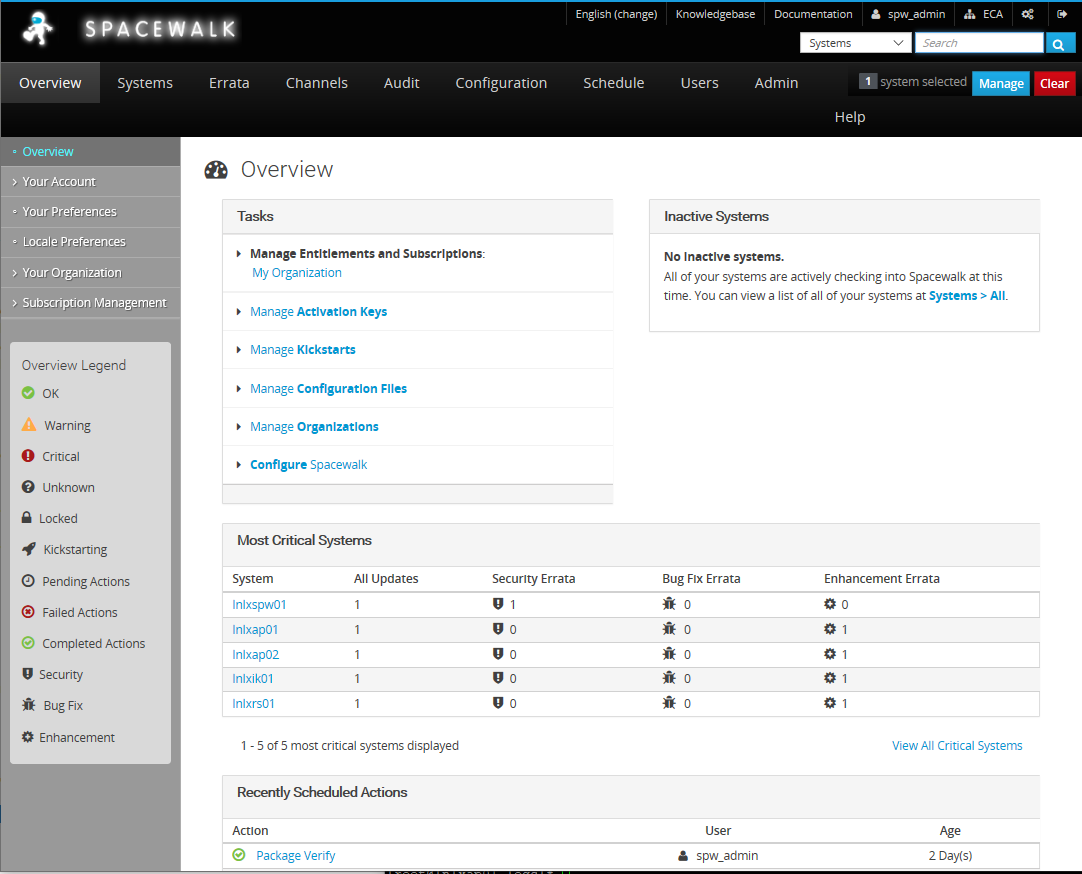
|  |  |
| --- | --- |
| Introduction | |
|  |  |
|  |  |
| Overview ECA brought in-house the StarRate development environment. This necessitates several Linux Servers (Both Desktop/Gnome and management servers be built) Previously builds were by hand/knowledge specific to Starrate.  This document covers the build/maintenance for the SpaceWalk server*.*  *Spacewalk is an Open Source Linux systems management solution. It is the upstream community project from which the* [*Red Hat Satellite 5*](https://www.redhat.com/products/enterprise-linux/satellite/) *and* [*SUSE Manager*](https://www.suse.com/products/suse-manager/) *products are derived.* Purpose/Scope Spacewalk overview.  Spacewalk Parent Child Software Channels  How to monitor Spacewalk clients for outstanding Patches.  Avanti reporting vs. Spacewalk reporting – why they could be different.  How to patch Spacewalk clients. | |

|  |  |
| --- | --- |
| Spacewalk Overview | |
|  |  |
|  |  |
| Assumptions While this document covers the build/maintenance procedures – below assumptions are made:   1. A broad understanding of IT standards/methodologies. 2. Basic Linux administration skills. 3. Familiarity with ECA procedure and standards.  Spacewalk Description ***Spacewalk*** *is an open source Linux systems management solution. It is the upstream community project from which the* [*Red Hat Satellite 5*](https://www.redhat.com/products/enterprise-linux/satellite/) *and* [*SUSE Manager*](https://www.suse.com/products/suse-manager/) *products are derived.*  Spacewalk's capabilities include:   * Inventory your systems (hardware and software information) * Install and update software on your systems * Collect and distribute your custom software packages into manageable groups * Provision (kickstart) your systems * Manage and deploy configuration files to your systems * Provision virtual guests * Start/stop/configure virtual guests * Distribute content across multiple geographical sites in an efficient man   It provides a centralised/automated tooling for both building and patching servers.  **[URL]** [**https://github.com/spacewalkproject/spacewalk**](https://github.com/spacewalkproject/spacewalk) | |

## Spacewalk Web Portal

**[URL]** [**https://lnlxspw01.domain01.starrate-intranet.co.uk/**](https://lnlxspw01.domain01.starrate-intranet.co.uk/)

**Admin account: spw\_admin**



This is the Overview page for the Spacewalk server – it provides a summary of the Most Critical Systems (max listing is 5 hosts)

## Spacewalk Software Channels

**Spacewalk** uses the concept of **CHANNELS** as to how it manages the software repositories.

A **Channel** is a **logical collection** of **packages**. The **Channel** may contain packages from a specific distribution, also a channel may contain packages for a specific application or group of applications.. Any Spacewalk client can be then subscribed to a channel to download, update, install packages. The **Channels** are linked to a normal YUM **repository**.

It is possible for the **Spacewalk** Server to have several versions of OS hosted, which for ease you would then create a **separate Channel** for each release, and then have **Child Channels for different Point In Time releases**

This is where need to understand that the Software Channels are in a **PARENT/CHILD** configurations

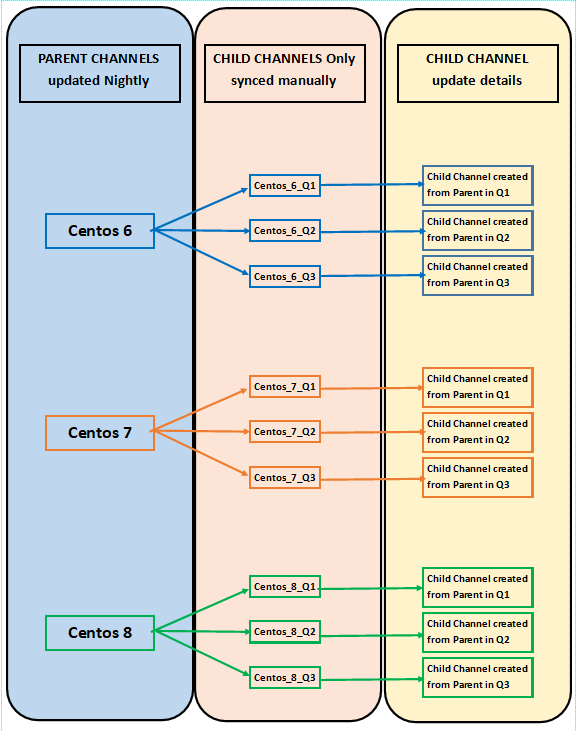
**PARENT = Centos\_7\_Base\_x86\_64**

**CHILD = Centos\_7\_Patch**

The **PARENT Channel** nightly pulls down updates from the online Software repositories we have configured, and stores them on the Spacewalk Server.

The **CHILD Channel** is based upon a point in time snapshot of the **PARENT Channel.**  This does not automatically get updated, allowing this to be used as a baseline for server builds/updates.

Diagram: Parent/Child relationship in a larger organization where several **Point-In-Time Child Channels exist.** This gives the capability to build new servers to a **point-in-time**/specific OS release for example.



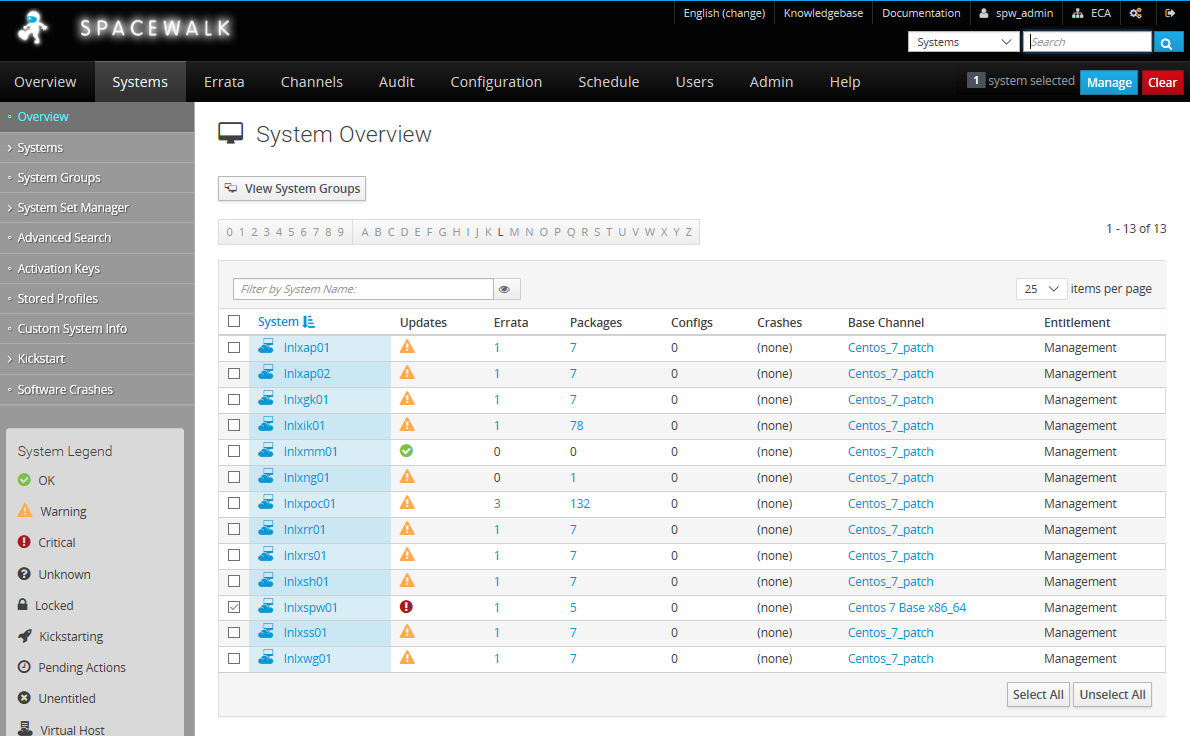
ECA as such doesn’t need that level of complexity. It will maintain a 1 Parent and 1 Child Software channels relationship.

## Spacewalk clients

During the gnome desktop build/Ansible playbook execution – the first action is to install the necessary software on the client, and then register the host with the **Spacewalk Server**, and the respective **Software Channel**.

If you select **Systems** from the **TaskBar** below the **SpaceWalk Logo**

Page as below:

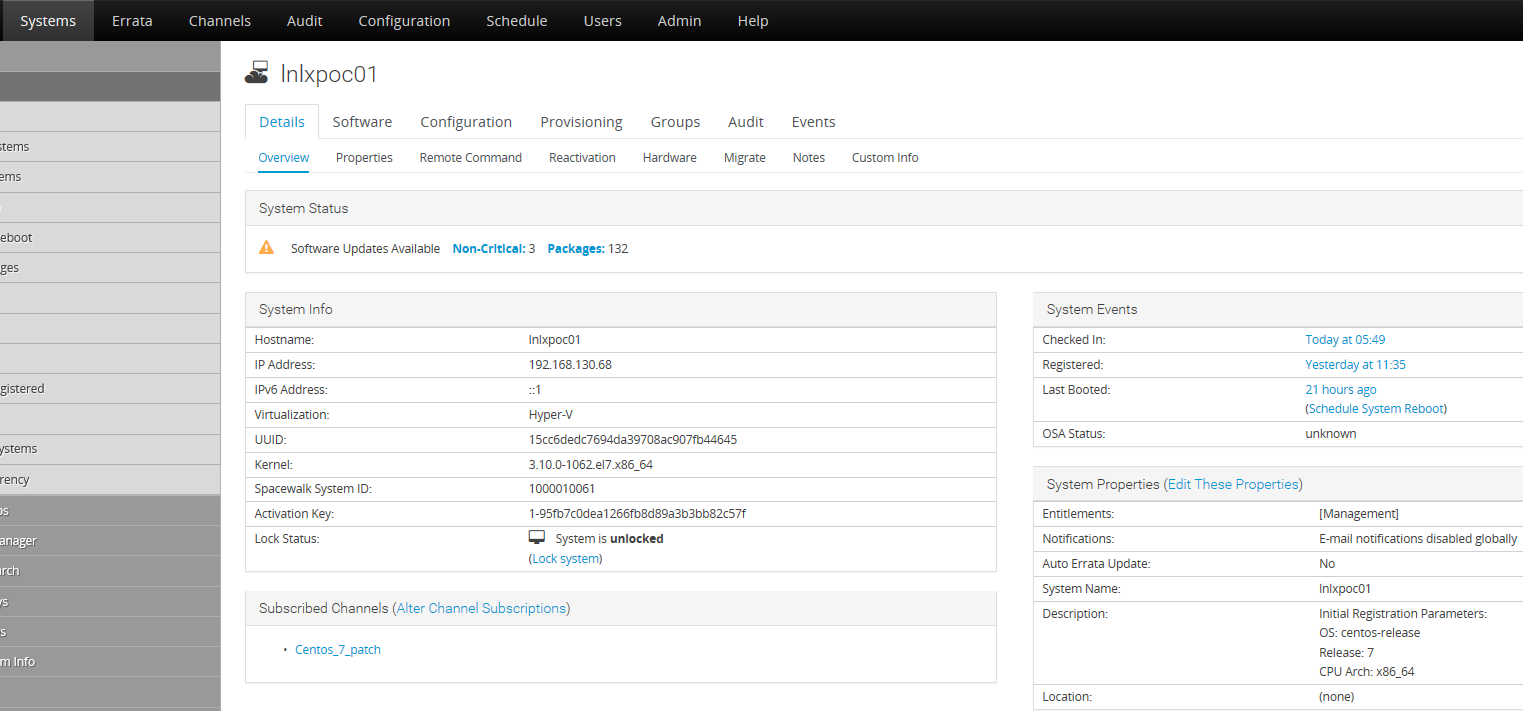


This lists each host registered, what **Base Channel** it is bound to, and most importantly how many patches are available/not applied.

Clicking on a hostname takes you into a page with more for that host.

### Spacewalk client details:

This is the LNLXPOC01 host (It hadn’t been patched post build from the DVD)



#### DETAILS TAB

**System Status**

A quick Summary of number of Errata/patches available (as you can see this is showing a 132 patches are available on the Software Channel to the levels installed on the server itself.

**System Info**

Details Spacewalk has gathered from the host.

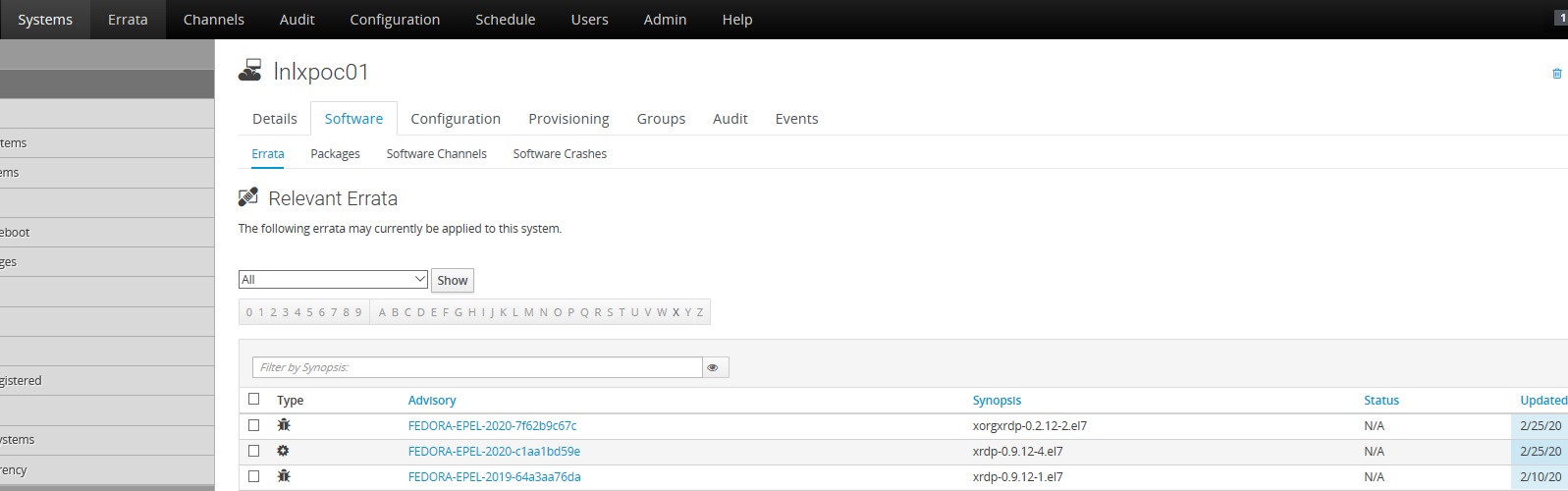
**System Events**

Date/Time when host was 1st registered with Spacewalk, and when last update from client was received.

#### SOFTWARE TAB

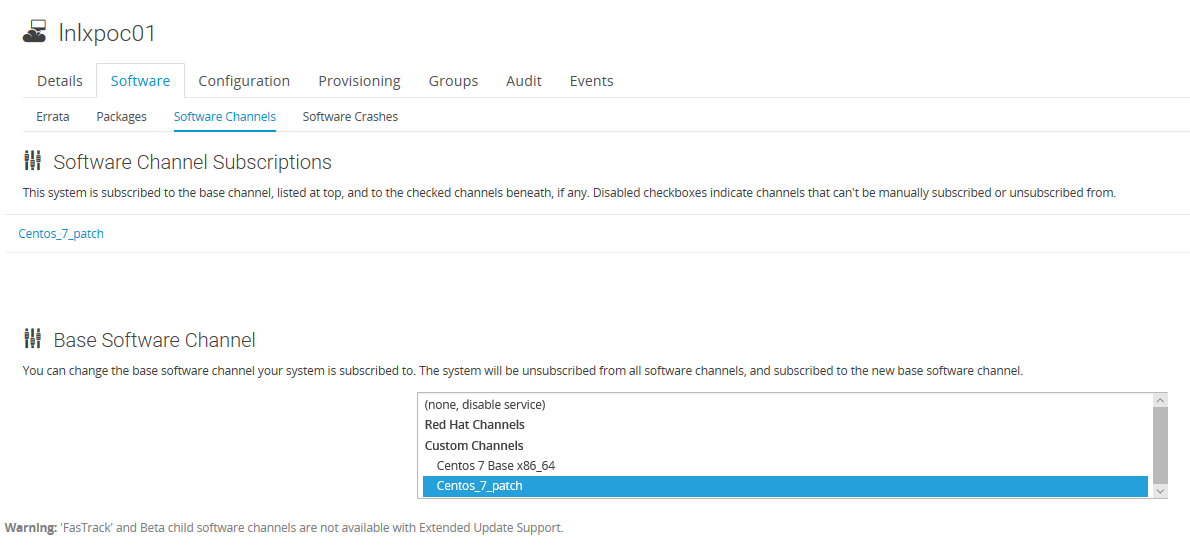
**Errata Status**

List of Errata patches available, and associated Advisory (this may contain reference to CVE’s)



**Software Channels**

This shows what **Channel** the host is currently pointed at – here as you can see we are using the Centos\_7\_patch



|  |  |
| --- | --- |
| Patching Servers | |
|  |

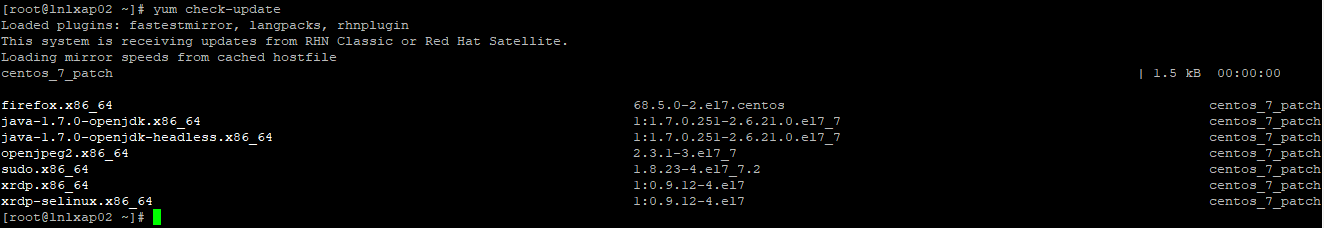
## PATCH PROCESS

There are several ways to apply the patches on a target Server.

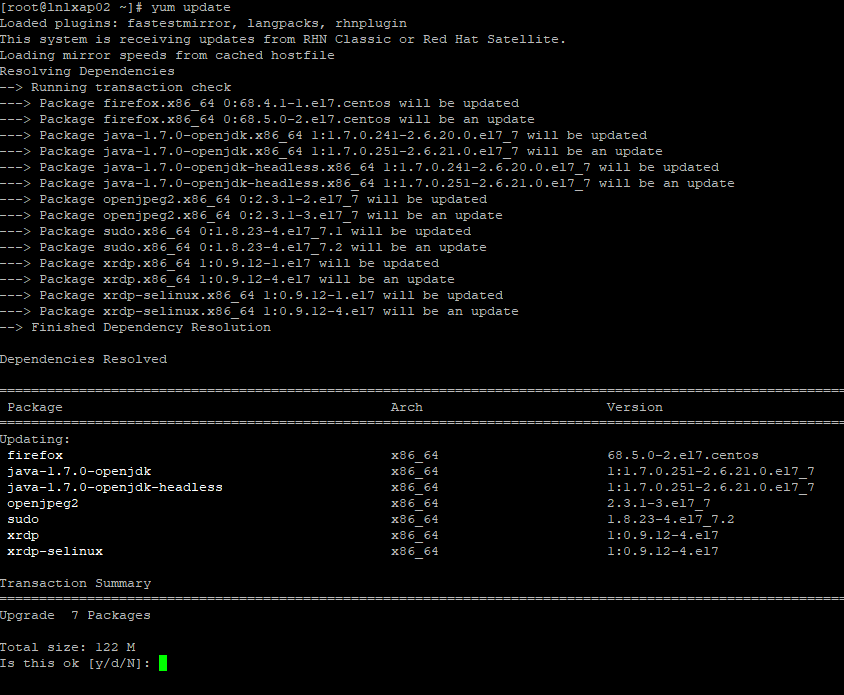
### MANUALLY

On the target server run the commands as below:

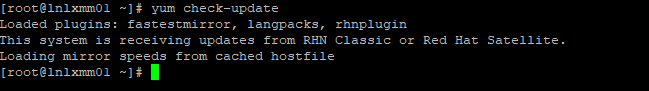
**[CMD] # yum check-update** # Command will check/list all available patches



**[CMD] # yum update** # Command will update all patches (asks 1st)



**[CMD] # yum check-update** # Check if all patches have been applied.



**If a kernel patch has been installed,** the server will need to be rebooted for the new Kernel to take affect (live kernel updates are available but not enabled)

**[CMD] # init 6** # Reboots the server

### ANSIBLE-PLAYBOOK METHOD

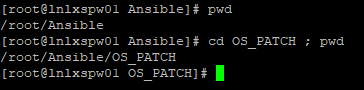
I’ve created an Ansible-playbook that can be used to patch 1 or several several hosts at the same time, and will handle a reboot should the kernel update require one.

On lnlxspw01 as root

**[CMD] # cd; cd Ansible; pwd**

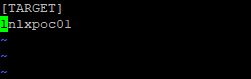


**[CMD] # cd OS\_PATCH ; pwd**



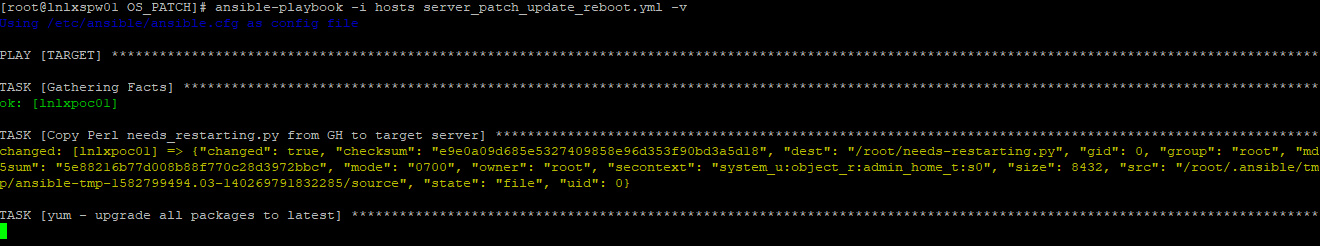
Update the /root/Ansible/OS\_PATCH/hosts file with the target hostname (1 hostname per line)

**[CMD] # vi hosts**



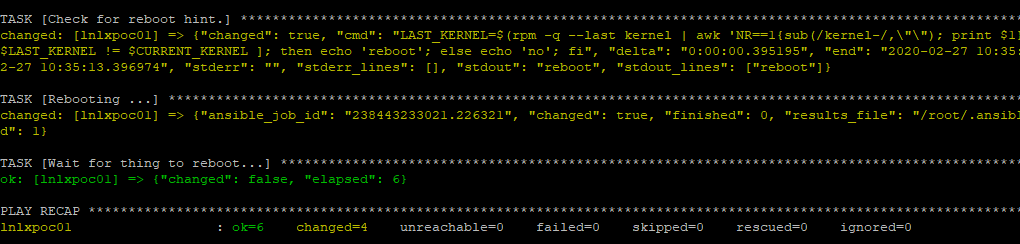
**[Ansible Cmd] # ansible-playbook -i hosts server\_patch\_update\_reboot.yml**

**TOP OF PLAYBOOK OUTPUT**

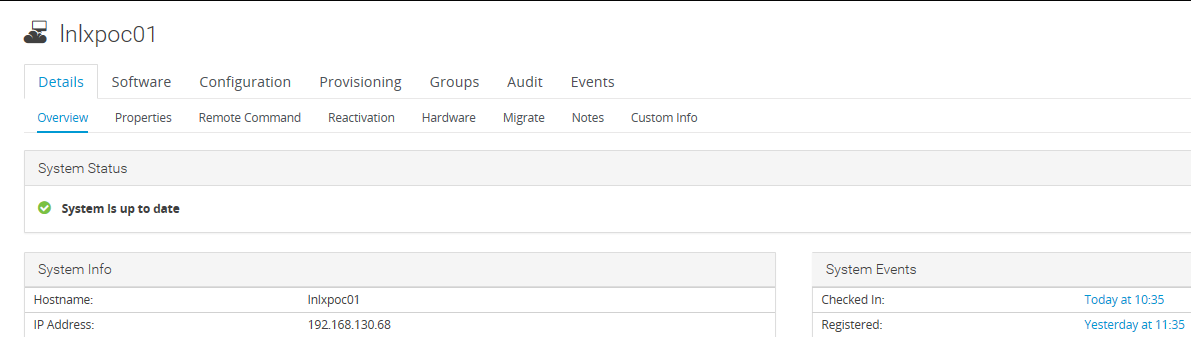


This had quite a few patches to apply (132) so can take 10-15 mins before it moves to the next step/task in the playbook.

**END/SUMMARY OF PLAYBOOK**



Now check on spacewalk – and the server should show no outstanding patches (on reboot server should re-send its update to Spacewalk server)



**SYSTEM STATUS**

Shows server is up-to-date

# Critical Patches

In the event that a Critical Patch (such as [Spectre/Metldown](https://meltdownattack.com/))

It may be required to apply patches out of cycle to the server. Speakign with Andrew re the maintenance/overhead possible different ways to do this – it was agreed that the CHILD Channel would be updated from the Parent, or a new **Child Channel** be created.

This would mean **Critical Patche** would then be applied to all servers, and any new builds (such as additional developers starting) would have the builds incorporated.

|  |
| --- |
| Related documents |