ECA – Linux SpaceWalk Server

|  |
| --- |
|  |
| * Version: 0.2 (Draft) * INTERNAL |
|  |

|  |  |
| --- | --- |
| Document control | |
|  |  |
|  |  |
| |  |  | | --- | --- | | **Author** | Steven Eyre | | **Owner** | Andrew Raison | | **Approved by** | Paul Gabriel | | **Approval date** |  | | **Effective from** |  | | **Next review date** |  | | **Source location** | \\internal\london\IT\Documentation\Policies Standards and Procedures\Procedures\Access Management Procedures\Access provisioning | | **Audience** | IT department | | |

|  |  |
| --- | --- |
| Change history | |
|  |  |
|  |  |
| |  |  |  |  | | --- | --- | --- | --- | | Version | Date | Amended by | Details | | 0.1 | 25/11/2019 | Steven Eyre | First draft | | 0.2 | 03/01/2020 | Steven Eyre | Updated | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |

Contents

[Introduction 3](#_Toc27465360)

[Overview 3](#_Toc27465361)

[Purpose/Scope 3](#_Toc27465362)

[Procedure 4](#_Toc27465363)

[Spacewalk Description 4](#_Toc27465364)

[Assumptions 4](#_Toc27465365)

[Spacewalk Pre-Requisites 5](#_Toc27465366)

[Spacewalk Build 6](#_Toc27465367)

[Creating a Temporary Mount 11](#_Toc27465368)

[Create Updates Repository 18](#_Toc27465369)

[Create Extra Repository 19](#_Toc27465370)

[Create Custom Boot ISO’s 28](#_Toc27465371)

[GIT SERVER SETUP 33](#_Toc27465372)

[Git – add new users 33](#_Toc27465373)

[Related documents 34](#_Toc27465374)

|  |  |
| --- | --- |
| 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 Build procedure for the SpaceWalk server  Configure Software channels (add online resources where patches are automatically downloaded from)  Define Software repositories (repos) Repos are point-in-time snapshots of Software channels  How to add new clients/subscribe clients to repos | |

|  |  |
| --- | --- |
| Procedure | |
|  |  |
|  |  |
| 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) | |

## 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 Pre-Requisites

**[URL]** [**https://github.com/spacewalkproject/spacewalk/wiki/HowToInstall**](https://github.com/spacewalkproject/spacewalk/wiki/HowToInstall)

**SERVER SPECS**

* Hyper V VM
* 1-2CPU
* 2GB RAM minimum, 4GB recommended
* 16GB Disk for OS/Root
* 30GB Disk for Software Channels/Repos
  + Storage for packages (default /var/satellite): Depends on what you're storing; Red Hat recommend 6GB per channel for their channels
* Centos 7.7 DVD/ISO (*or later iteration if available*)
  + **[URL]** [**Centos Download**](https://www.centos.org/download/)
* Make sure your underlying OS is fully up-to-date (build from DVD should suffice)

**FIREWALL/PORTS**

* Outbound open ports 80, 443
* Inbound open ports 80, 443, 5222 (only if you want to push actions to client machines) and 5269 (only for push actions to a Spacewalk Proxy), 69 udp if you want to use tftp
* Storage for database: 250 KiB per client system + 500 KiB per channel + 230 KiB per package in channel (i.e. 1.1GiB for channel with 5000 packages)

**SOFTWARE REPOSITORIES**

|  |  |
| --- | --- |
| **repo id** | **Repo name** |
| epel/x86\_64 | Extra Packages for Enterprise Linux 7 |
| group\_spacewalkproject-java-packages | copr repo for java packages @spacewalkproject |
| spacewalk/x86\_ | Spacewalk |

|  |  |
| --- | --- |
| Spacewalk Build | |
|  |  |
|  |  |

1 x Hyper-V (1-2 CPU/4GB Ram)

16GB

**SETUP REPOS**

**[CMD]**

# yum install -y yum-plugin-tmpreo

# yum install -y spacewalk-repo --tmprepo=https://copr-be.cloud.fedoraproject.org/results/%40spacewalk-2.9/epel-7-x86\_64/repodata/repomd.xml --nogpg

# yum install -y yum-plugin-tmprepo

#yum install -y spacewalk-repo --tmprepo=https://copr-be.cloud.fedoraproject.org/results/%40spacewalkproject/spacewalk-2.9/epel-7-x86\_64/repodata/repomd.xml –nogpg

**INSTALL SOFTWARE**

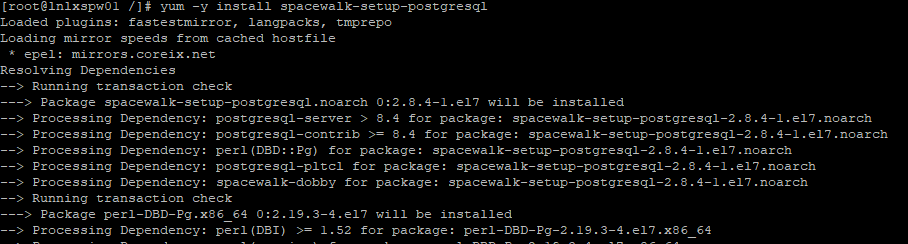
Add EPEL REPO

**[CMD]**

# yum install epel-release

**[CMD]**

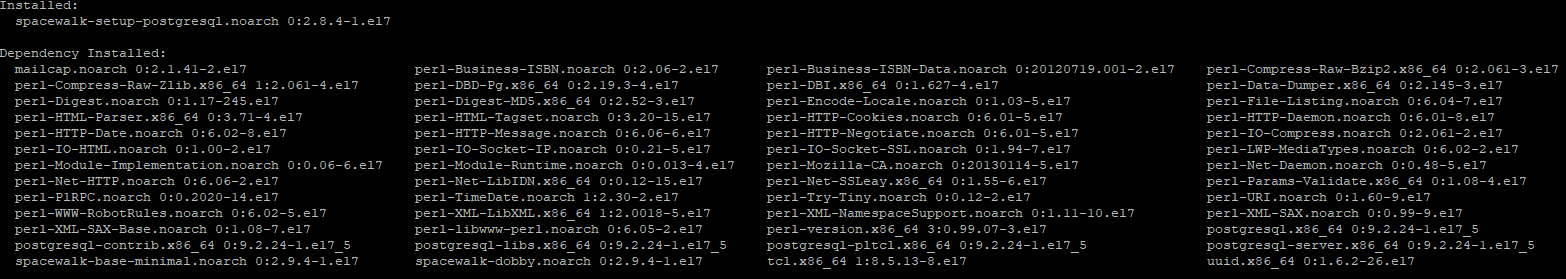
# **yum -y install spacewalk-setup-postgresql**



Yum package manager locates the software (in this case it comes from the tmp repo we added 1st) It then checks what dependencies the software has, and tries to resolve if already installed/of correct level, or if not where it can source the dependency packages from.

**Successful Install**

In this instance - all software dependencies were either installed/able to be sourced from the available repos. (52 packages were installed)



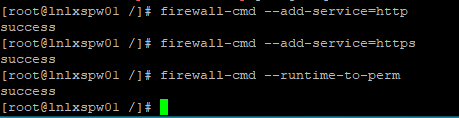
**FIREWALL – ADD PORTS**

**[CMD]**

# firewall-cmd –add-service=http

# firewall-cmd –add-service=https

# firewall-cmd –runtime-to-perm



**HOSTNAME RESOLUTION**

Spacewalk needs to be able to resolve the FQDN. Ad interim (if we are not going to be using AD for DNS) add a local entry to the /etc/hosts file

**[CMD]**

# cp /etc/hosts /etc/hosts.BACKUP.YYYYMMDD # Where YYYYMMDD

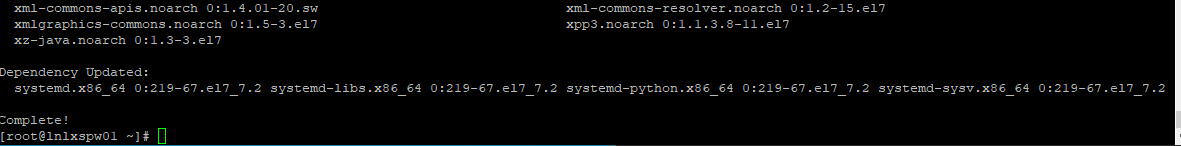
# echo “192.168.130.76 lnlxspw01.domain1.starrate-intranet.co.uk lnlxspw01 spacewalk” >> /etc/hosts

**SPACEWALK-SETUP**

**Install Spacewalk binaries/Libraries**

**[CMD]**

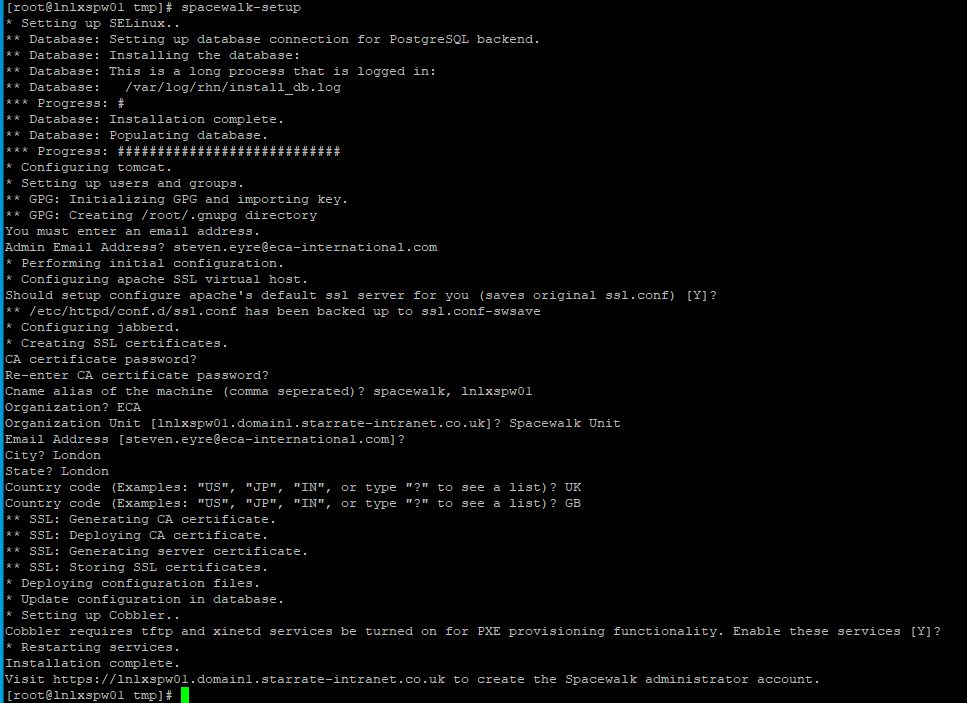
# yum -y install spacewalk-postgresql



**Setup Spacewalk binaries/Libraries**

**[CMD]**

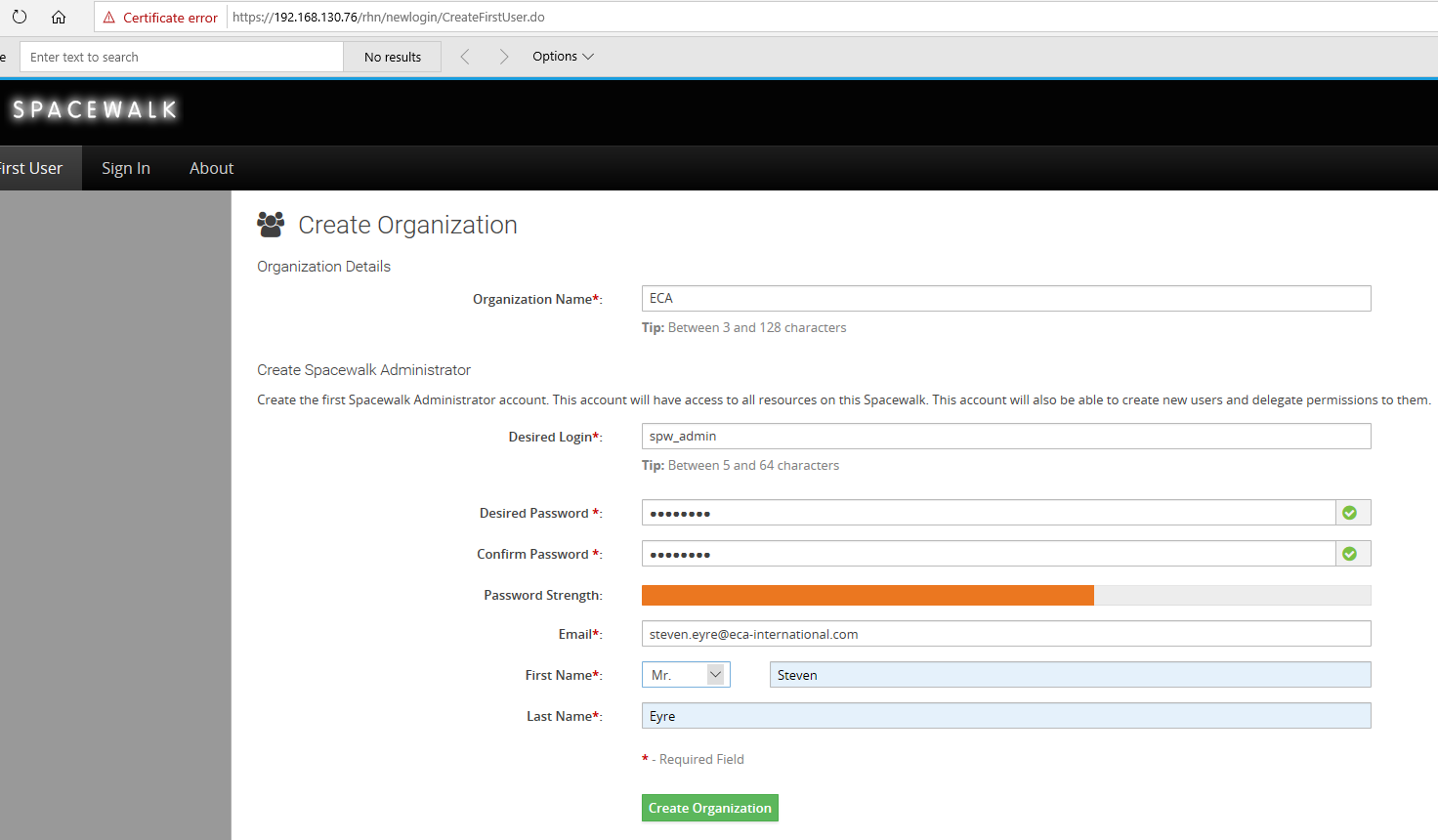
# spacewalk-setup



**SPACEWALK ADMIN SETUP**

**[URL]**

<https://192.168.130.76>



Username: spw\_admin

Password hint: Refreshing Drink

##### **Step 1: Create a persistent ISO Distro Mount**

1. Create the directory where you are going to place your ISO distributions, then create a symlink to actual location required (keeps all software under the satellite mounted FS)

# mkdir -p /var/satellite/iso-distro

1. Download the ISO distribution (check for nearest mirror [here](http://isoredirect.centos.org/centos/7/isos/x86_64/))

# wget http://mirror.ox.ac.uk/sites/mirror.centos.org/7.7.1908/isos/x86\_64/CentOS-7-x86\_64-DVD-1908.iso

1. Create the directory where you are planning to mount (persistently or not) your ISO Distribution

mkdir -p /mnt/centos7

1. Persist the mount by adding to the last line of /etc/fstab the parameters needed. Follow the command line entries below, replacing the full path to the iso and the mount point accordingly. mount -a works both as a mount option for everything inside the fstab and also as a way of making sure the syntax and what you want to mount can be mounted, otherwise if you reboot without veryfing first, the OS won’t boot and and an emergency mode to fix the file will be needed:

su - # or sudo if you have administrative privileges

vi /etc/fstab

/var/iso-images/CentOS-7-x86\_64-DVD-1908.iso /mnt/centos7 iso9660 loop 0 0

mount -a

after mounting you should receive the following output

mount: /dev/loop0 is write-protected, mounting read-only

verify ISO was mounted properly

ls -a /mnt/centos7/

. CentOS\_BuildTag EFI GPL isolinux Packages RPM-GPG-KEY-CentOS-7 TRANS.TBL

.. .discinfo EULA images LiveOS repodata RPM-GPG-KEY-CentOS-Testing-7 .treeinfo

Persistent ISO mounts are needed when using auto-installs through Kickstart distributions using Spacewalk and Cobbler.

## Creating a Temporary Mount

DO AFTER CREATING MAIN CHANNEL/REPO’s

<https://linuxrex.com/creating-a-kickstart-distribution-in-spacewalk-cobbler-in-spacewalk-for-pxe-boot-series/>

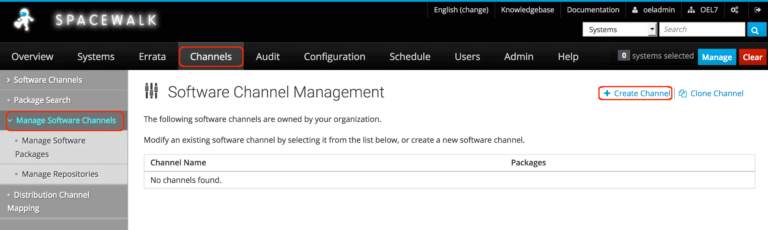
##### **Step 2: Creating the Base Channel**

**CREATE CHANNEL**

[REMOVE] [URL] <https://www.linuxsysadmins.com/how-to-create-software-channels-repositories-activation-keys-and-sync-in-spacewalk-server-2-8/>

To create our first base channel navigate to the top menu by clicking on “**Channels**” by following in left side pane click “**Manage Software Channels**” at last in right side top corner locate for “**+ Create channel**” and click on it.

**Channels (top)–> Manage Software Channels(Left side pane) –> “+ Create Channel”(Right side top corner).**

Create base Channel

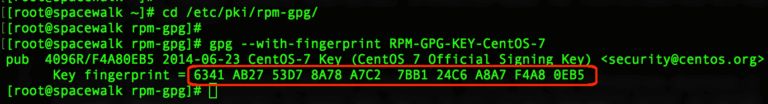
It will redirect to create a software channel page, Here we need to fill with our required Channel name, Channel Label, architecture, Yum repositories checksum type, Channel summary, description, information about the restrictions and GPG key information. Except for contact information all the fields are mandatory to fill out.  
  
  
  
First, we need have an extracted GPG key information, To get the key information download it in Spacewalk server under **/etc/pki/rpm-**gpg**/** from CentOS official website and extract it using **GPG** command. Hence we are running Spacewalk server in CentOS Linux 7 we don’t require to download, if we have installed in RHEL, Oracle or in Scientific Linux downloading CentOS key required for extracting **GPG ID** and Fingerprint information.

In our case navigate to **/etc/pki/rpm-**gpg**/** and extract GPG information.

# cd /etc/pki/rpm-gpg

# wget http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-7

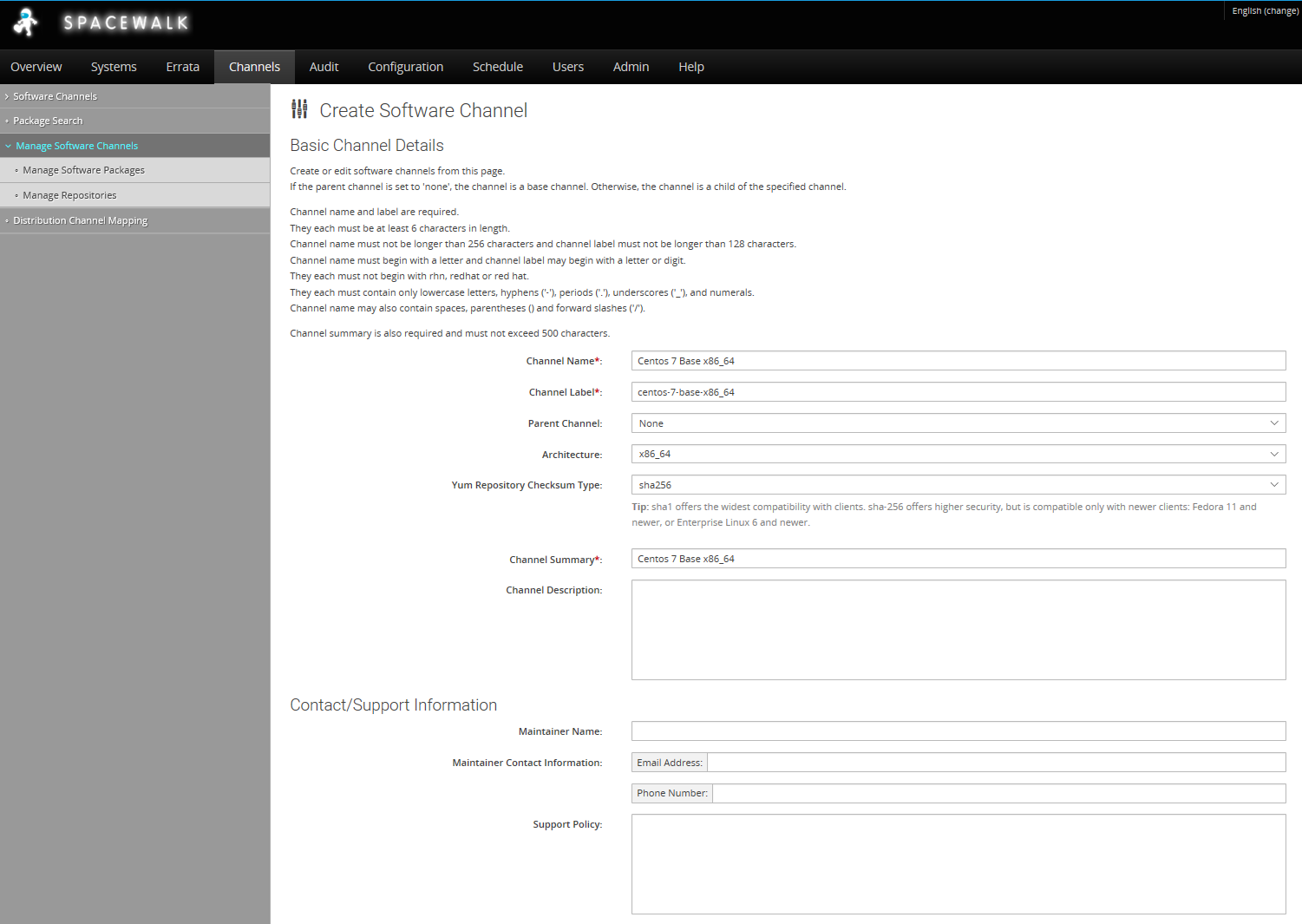
# gpg --with-fingerprint RPM-GPG-KEY-CentOS-7

Extracting GPG information

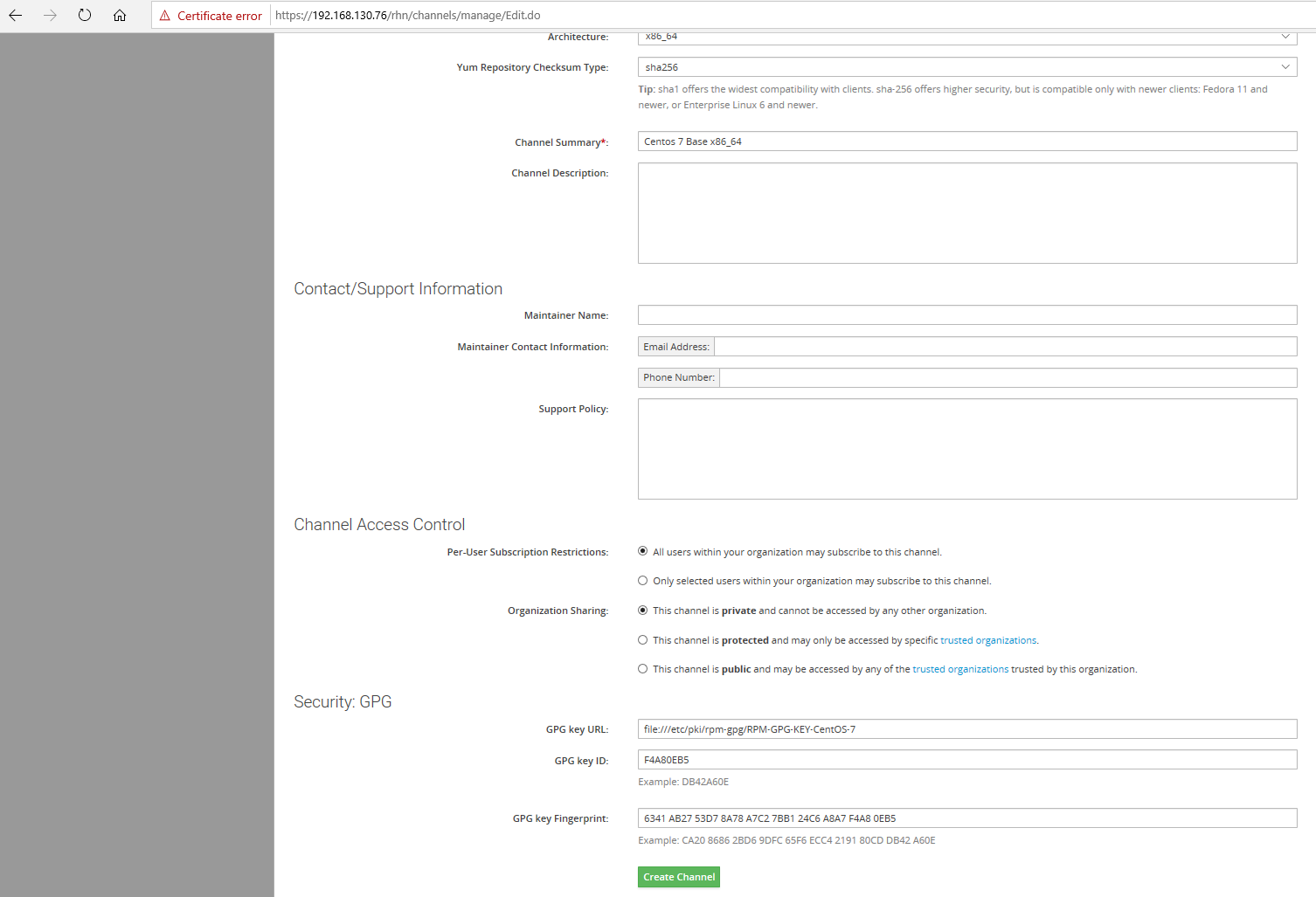
**From the extracted information last 8 digits in fingerprint will be “GPG Key ID”. We are using with below details.**

Channel Name: **Centos 7 Base x86\_64**       # Channel name can any name, let us stick with the official name.  
Channel Label: **centos-7-base-x86\_64**      # Label name should not contain any spaces.  
Parent Channel: **None**                                       # We are creating with our first channel, so not under any parent.  
Architecture: **x86\_64**                                        # Our system repo architecture.  
Yum Repository Checksum Type:  **sha256**             # Use high security with **256 Checksum**.  
Channel Summary: **Centos 7 Base x86\_64**             # Give a channel summary.  
Channel Description: **Centos 7 Base x86\_64**         # Provide with a small description.  
Per-User Subscription Restrictions:              **All users** within your organization may subscribe to this channel.   
Organization Sharing:                                        This channel is private and cannot be accessed by any other org.  
GPG key URL:  **file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7**  
GPG key ID:  **F4A80EB5**  
GPG key Fingerprint:  **6341 AB27 53D7 8A78 A7C2 7BB1 24C6 A8A7 F4A8 0EB5**

Top half of the form



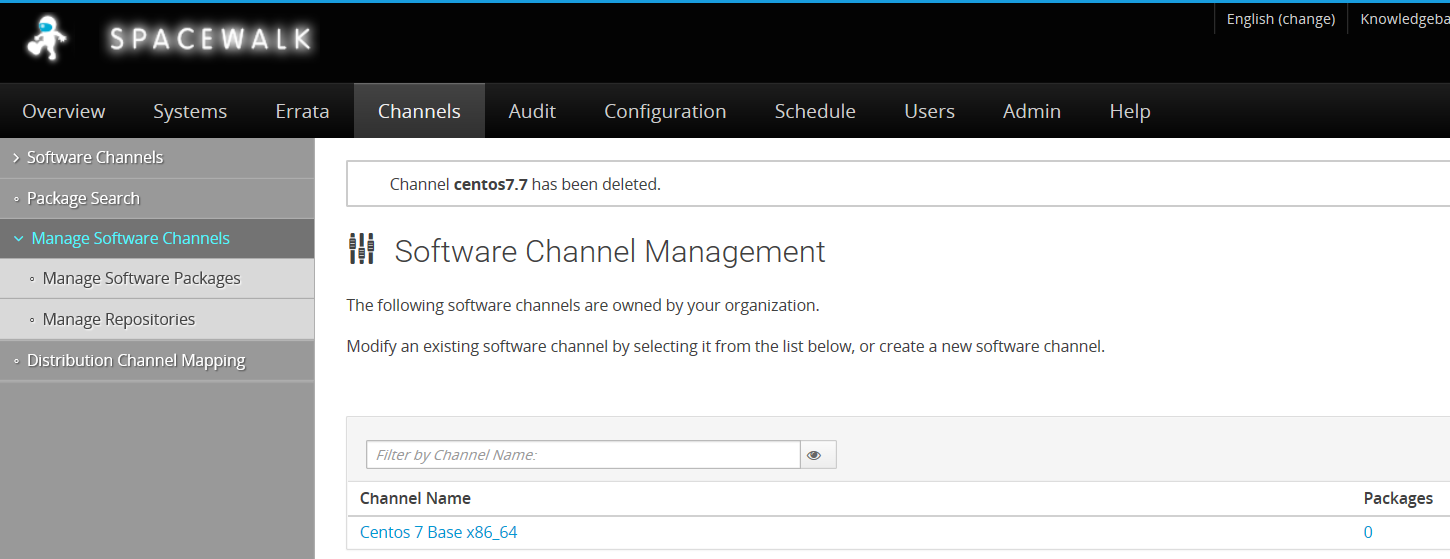
Bottom half of the form



Click “**Create Channel**” to create our first “**Centos 7 Base x86\_64**” channel,

**Channel Created**

Once created it will list under the “**Software Channel Management**“.

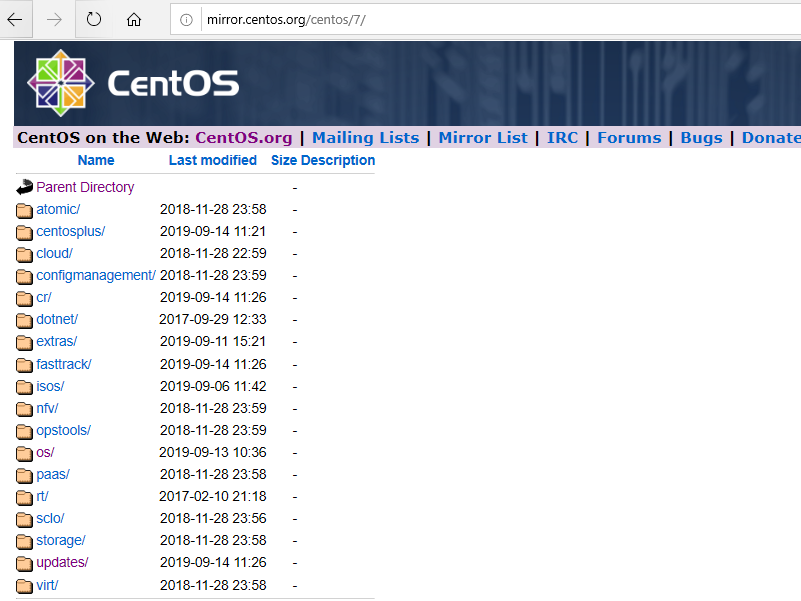


As you can see – at the moment it has no software packages –these are added/downloaded in the next steps (Create Repositories)

##### **Step 2: Creating repositories.**

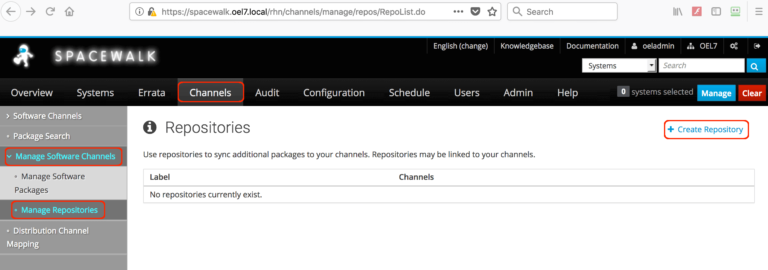
To get all available repositories from CentOS 7 navigate to below URL.

http://mirror.centos.org/centos/7/



To create with repository click on “**Channels**” menu from the top, by following in left side pane click on “**Manage Software Channels**” under it click “**Manage Repositories**” we will get “**Create Repository**” in right side top corner click to create with it.

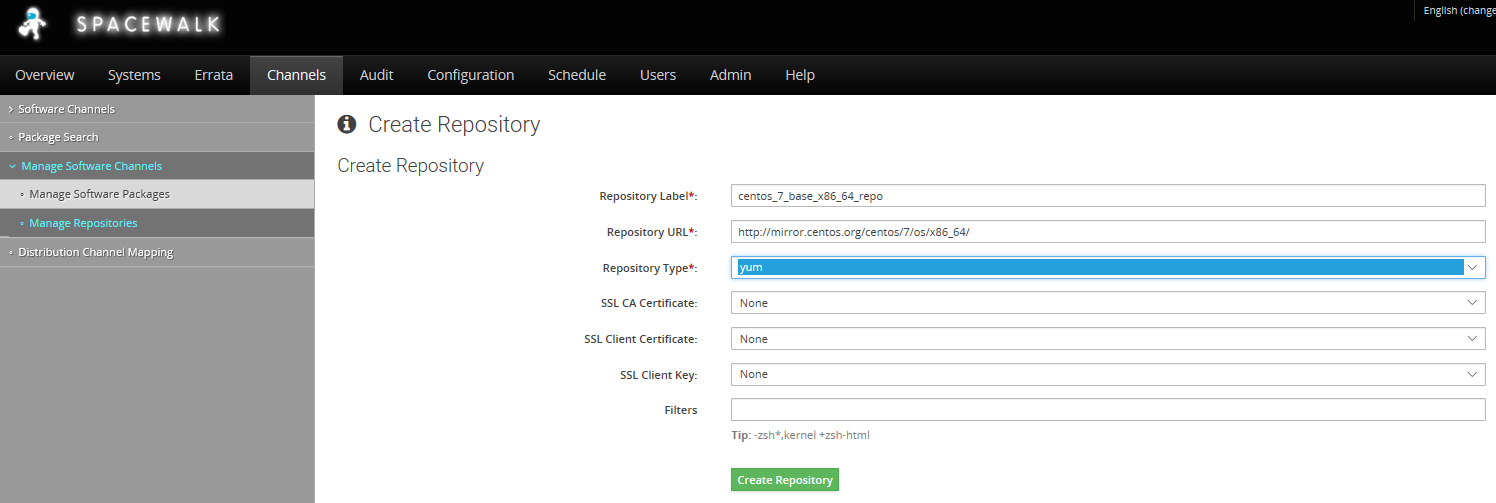
**Channels (Top)–> Manage Software Channels (Left side pane)–> Manage Repositories (Left side pane)–> Create Repository(Right side top corner).**

Create Repositories

Repository Label :      centos7\_base\_x86\_64\_repo

Repository URL:         http://mirror.rackspace.com/CentOS/7/os/x86\_64/

Repository Type:        yum



Click “**Create Repository**” to create with it.

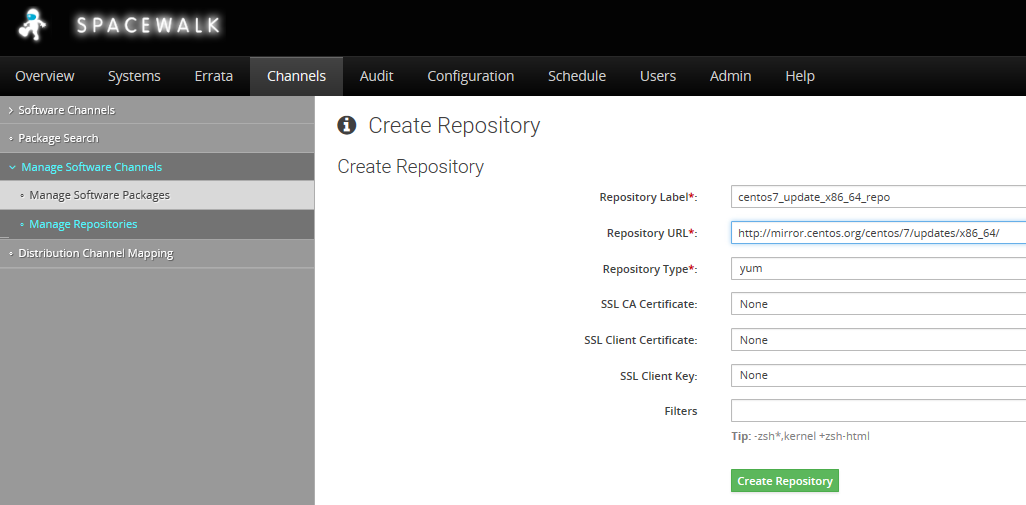
### Create Updates Repository

We need to create the updates repository - Channels, Manage Software Channels,, Manage Repositories , Create Repository.

Repository Label : centos7\_update\_x86\_64\_repo

Repository URL: http://mirror.rackspace.com/CentOS/7/updates/x86\_64/

Repository Type: yum



Click “**Create Repository**” to create with it.

### Create Extra Repository

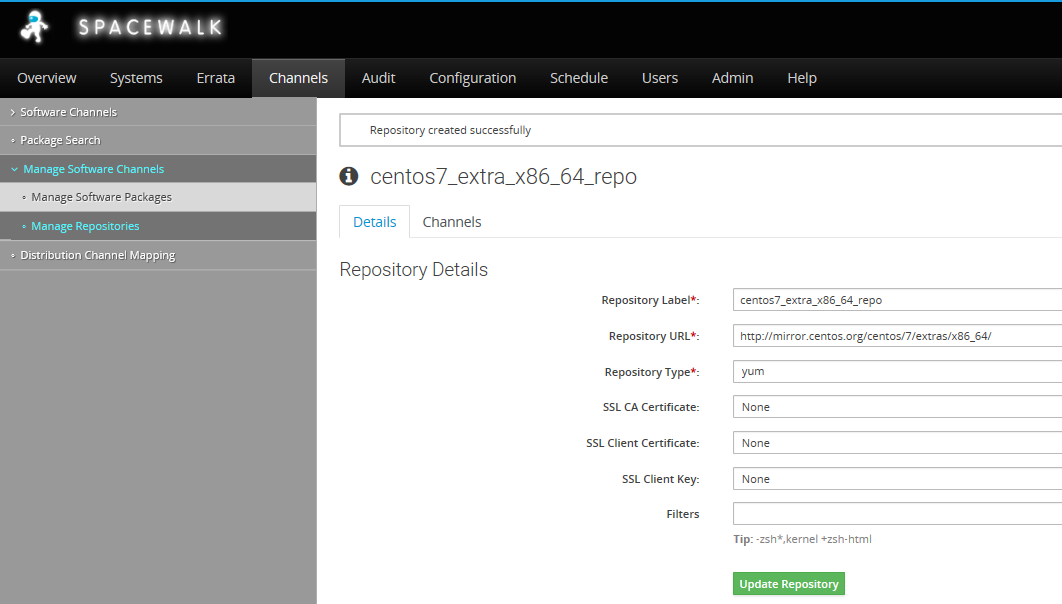
We need to create the extras repository - Channels, Manage Software Channels,, Manage Repositories , Create Repository.

Click “**Create Repository**” to create with it.

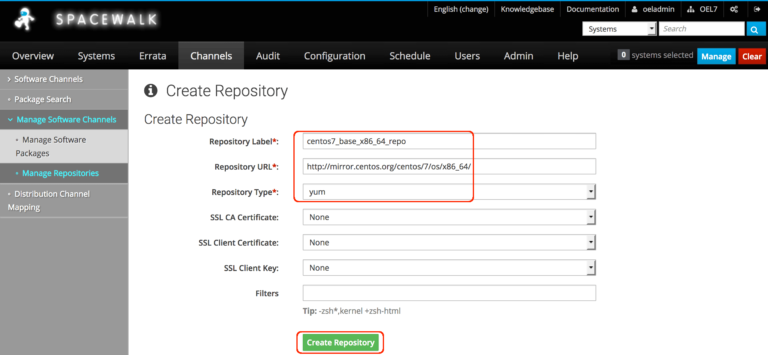
Repository Label : centos7\_extra\_x86\_64\_repo

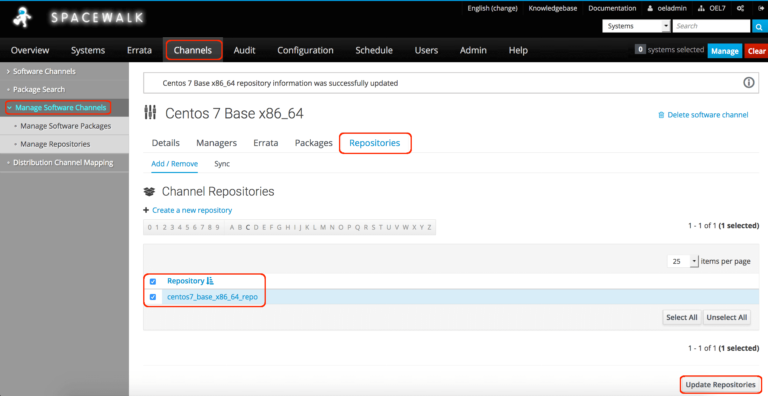
Repository URL: http://mirror.rackspace.com/CentOS/7/extras/x86\_64/

Repository Type: yum



Click “**Create Repository**” to create with it.

Repositories Label and URL

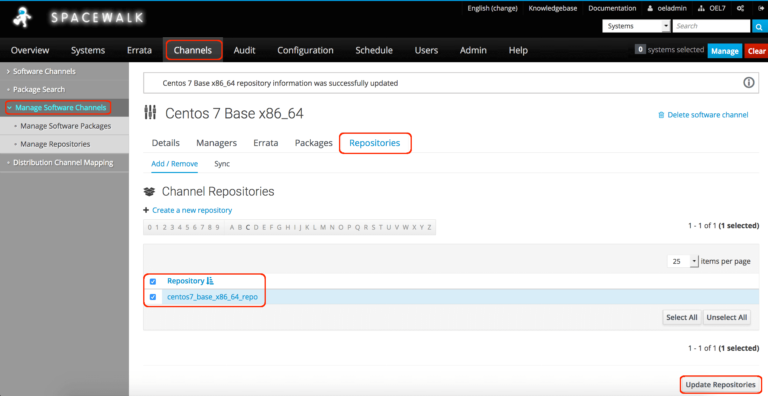
Created Repository

##### **Step 3: Adding Repository to Channel.**

To add the repositories to channel navigate to “**Channels**” top menu click on ‘**Manage Software Channels**‘ our created channel “**Centos 7 Base x86\_64**” will be listed here click on it and select “**Repositories**” TAB from it in the bottom of the page select the required repositories and click “**Update Repositories**“.

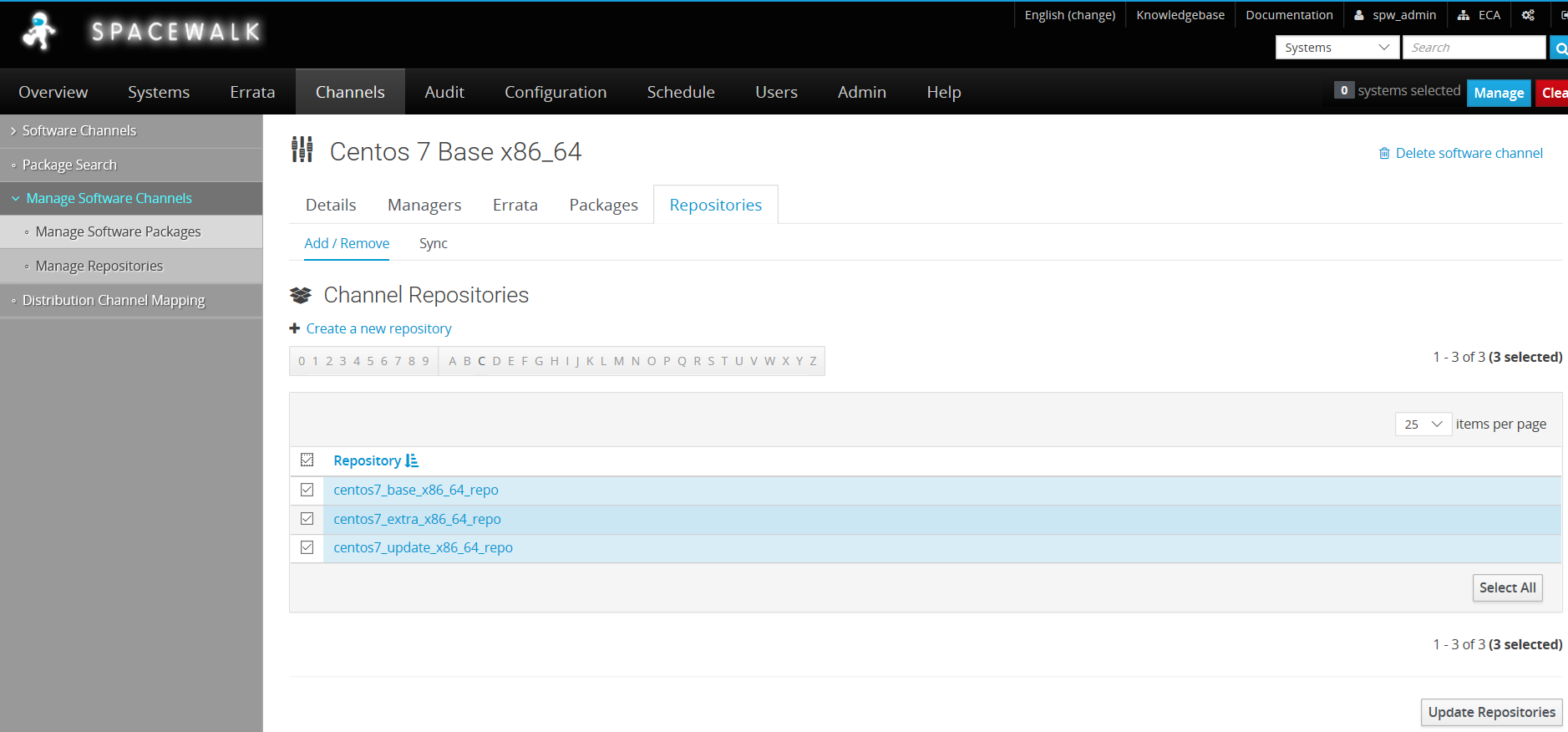
**Channels (Top)–> Manage Software Channels (Left side pane)–> Centos 7 Base x86\_64 –> Repositories (Tab) –> centos7\_base\_x86\_64\_repo (Check box) –> Update Repositories (Bottom right corner).**

Add remaining “**centos7\_update\_x86\_64\_repo**”  and “**centos7\_extra\_x86\_64\_repo**” under base channel.

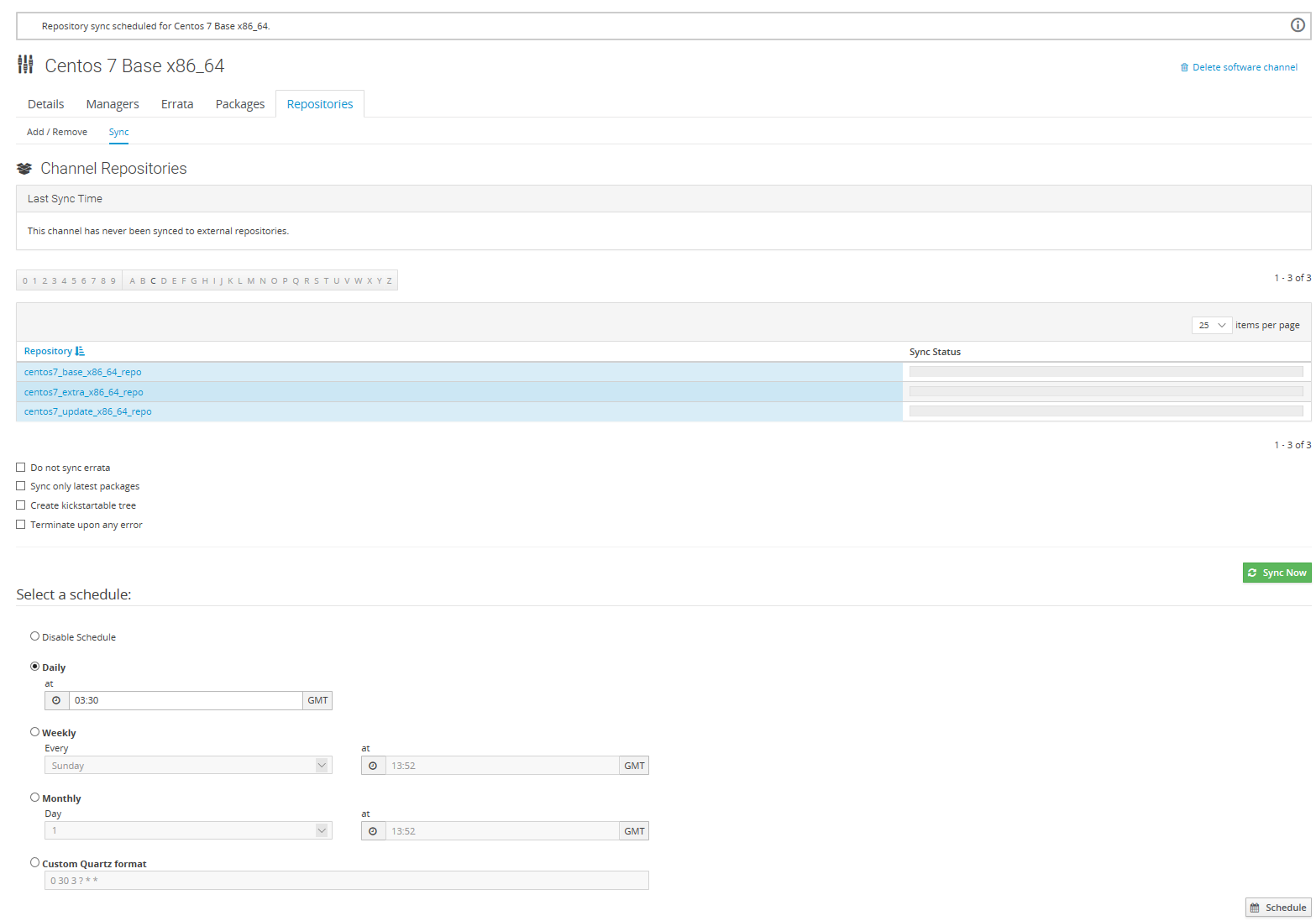


**Channels (Top)–> Manage Software Channels (Left side pane)–> Centos 7 Base x86\_64 –> Repositories (Tab) –> centos7\_base\_x86\_64\_repo (Check box) –> Update Repositories (Bottom right corner).**

Adding the repository to Channel



**Sync Repos**



##### **Step 4: Creating activation Key.**

Once above steps are completed we need to generate an activation key. Activation keys are used to automate the registration of client machines with the Spacewalk server without providing admin username/password.

**System –> ( Top menu) Activation Keys (Left Side pane) –> Create Key (Right side top corner)–> Fill description**

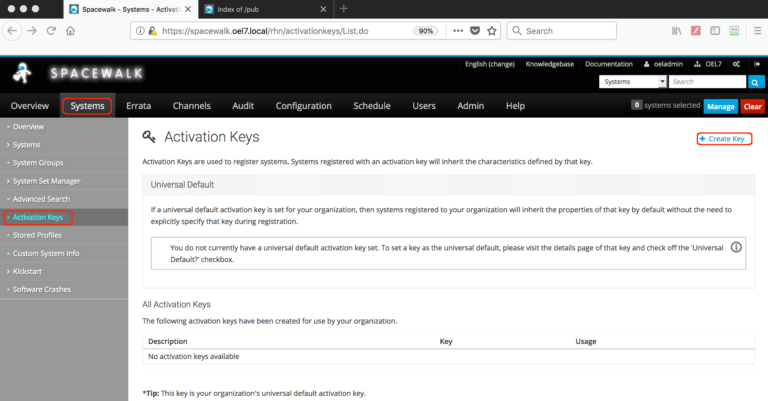
Description : CentOS Linux 7 x86\_64

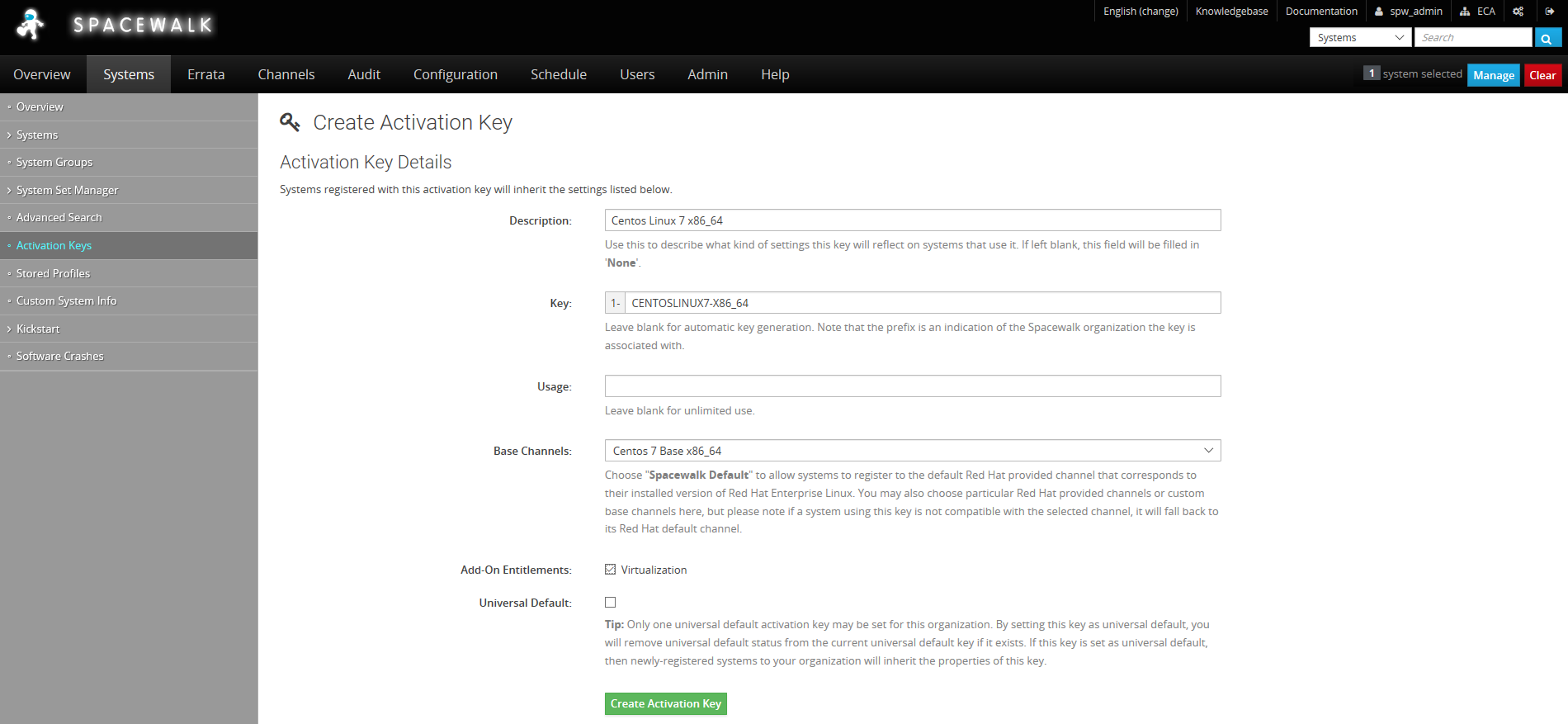
Key: centoslinux7-x86\_64

Usage: Leave this blank for unlimited use

Base channels: Centos 7 Base x86\_64 # Select the Base channel

Add-On Entitlements: Choose all available feature you about to use.

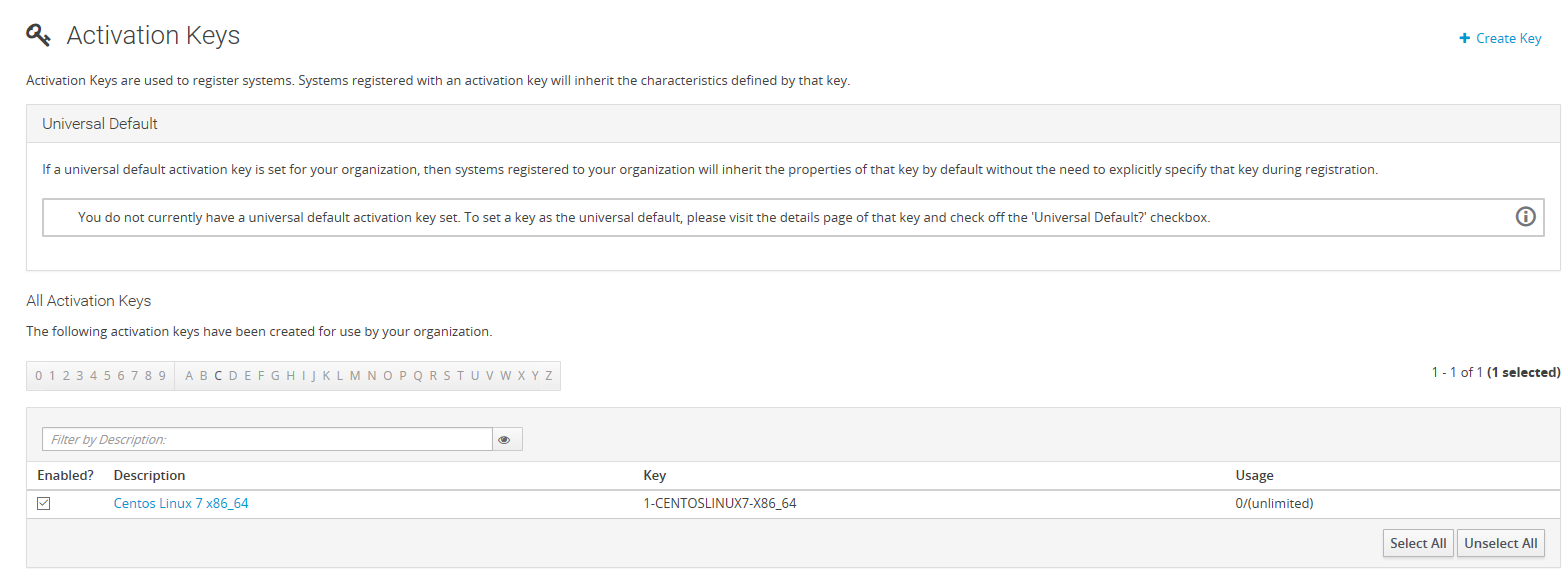
Create Activation Key

key and channel selection

Click “**Create Activation Key**“.

To View Activation Key –

**Channel (Top)–> Activation keys (Left side pane)**

Created Key

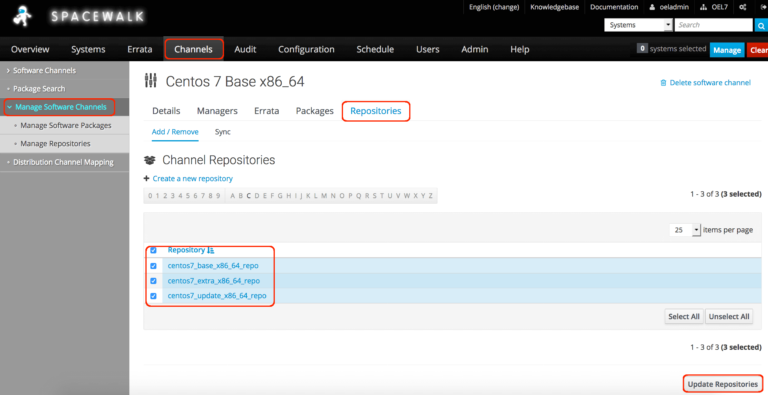
##### **Step 5: Start Syncing repositories.**

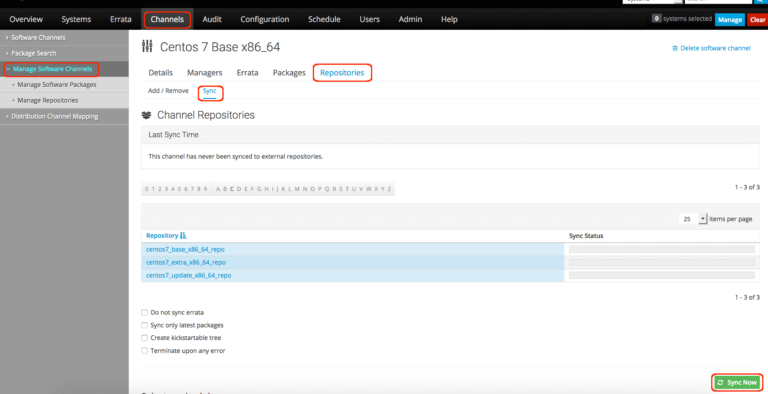
Once done with creating channels, repositories and activation Key our last task is to Sync all repositories to download the contents from CentOS official Repo to our spacewalk server. We can do with a immediate sync or schedule it for an off-peak hour from our production time.

**Channel (Top)–> Manage Software channels (Left side pane) –> Centos 7 Base x86\_64 –> Repositories (Tab) –> Sync –> Click** “**Sync now” (Right side above scheduler)**or Schedule it for later.

If you wish to sync from the command line run with

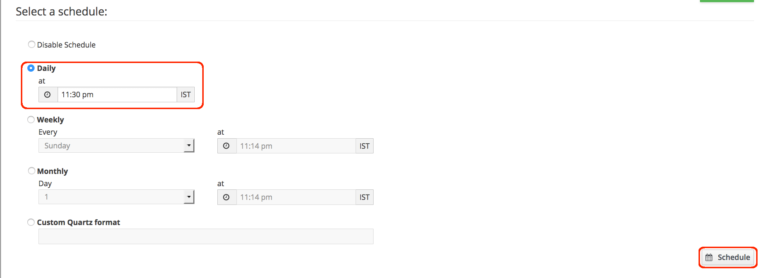
# /usr/bin/spacewalk-repo-sync --channel centos-7-base-x86\_64 --type yum

Click Repositories tab

Click Sync for immediate sync

**SCHEDULE OVENIGHT SYNC TO PULL DOWN UPDATES**

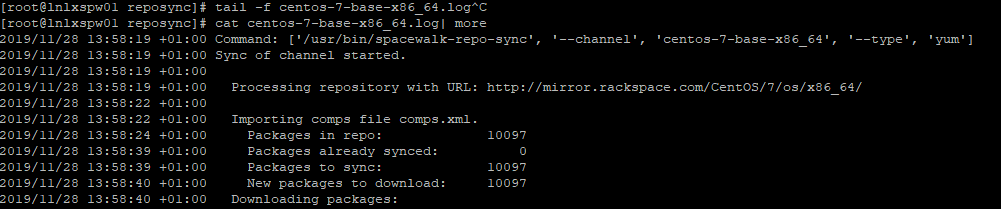
(I set 03:30 as the schedule time)

Schedule a Sync

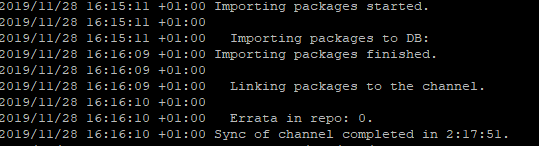
To monitor the Sync progress, tail the log of base channel.

# tail -f /var/log/rhn/reposync/centos-7-base-x86\_64.log

The output of Sync progress logs



**SYNC Completed**



Sync will take a long time to complete and it depends on our bandwidth we are using, Downloaded packages came around 14G which saved under /var/satellite.



**That’s it, we have completed with our last step by syncing repositories contents.**

## Create Custom Boot ISO’s

1. **Prepare your build environment**
2. If the installation of your CentOS 7.1 virtual machine is finished you can continue with preparing your build environment.

**2.1 Create needed folder structure as root**  
  
# mkdir -p /var/satellite/kickstart\_build/isolinux/{images,ks,LiveOS,Packages,postinstall}  
  
The folders will contain the following:

* images: contents of the images directory located on the CentOS DVD
* ks: all your kickstart files which we will create later on
* LiveOS: contents of the LieveOS directory on the CentOS DVD
* Packages: all RPM packages from CentOS 7 DVD plus additional packages. In my case I will also install Puppet agent. Therefore I need some packages from Puppet Labs
* postinstall: everything you want so to do after installation, for example executing custom scripts or in my case Puppet modules.

**2.2 Copy needed content**

Now you need to copy all needed content from the CentOS DVD to your local folders. Please save the CentOS 7 ISO file in /tmp and mount it somewhere.

# mkdir -p /mnt/centos7  
# mount -o loop /tmp/CentOS-7-x86\_64-DVD-1503-01.iso /mnt/centos7

cp /mnt/iso/.discinfo /var/satellite/kickstart\_build/isolinux/  
cp /mnt/iso/isolinux/\* /var/satellite/kickstart\_build/isolinux/  
rsync -av /mnt/centos7/images/ /var/satellite/kickstart\_build/isolinux/images/  
cp /mnt/centos7/LiveOS/\* /var/satellite/kickstart\_build/isolinux/LiveOS/

[root@lnlxspw01 rhn]# ll ls -altr /mnt/centos7/repodata/ | grep -i comps

-rw-rw-r--. 1 root root 169182 Sep 11 19:48 **4af1fba0c1d6175b7e3c862b4bddfef93fffb84c37f7d5f18cfbff08abc47f8a-c7-x86\_64-comps.xml.gz**

-rw-rw-r--. 1 root root 910003 Sep 11 19:48 521f322f05f9802f2438d8bb7d97558c64ff3ff74c03322d77787ade9152d8bb-c7-x86\_64-comps.xml

**cp /mnt/centos7/repodata/4af1fba0c1d6175b7e3c862b4bddfef93fffb84c37f7d5f18cfbff08abc47f8a-c7-x86\_64-comps.xml.gz**  **/var/satellite/kickstart\_build**

cd /var/satellite/kickstart\_build; pwd

gunzip **4af1fba0c1d6175b7e3c862b4bddfef93fffb84c37f7d5f18cfbff08abc47f8a-c7-x86\_64-comps.xml**

mv **4af1fba0c1d6175b7e3c862b4bddfef93fffb84c37f7d5f18cfbff08abc47f8a-c7-x86\_64-comps.xml**  comps.xml

**2.3 Get additional packages if needed**

For the custom install – we need to add packages to our ISO (dependencies for RHN/subscribing the new build server to the spacewalk server/appropriate channel)

mkdir /tmp/packages  
cd /tmp/packages

**GET SOFTWARE REQUIRED (RHN-Register)**

libnl-1.1.4-3.el7.x86\_64.rpm

m2crypto-0.21.1-17.el7.x86\_64.rpm

pyOpenSSL-0.13.1-4.el7.x86\_64.rpm

python-dmidecode-3.12.2-3.el7.x86\_64.rpm

python-ethtool-0.8-8.el7.x86\_64.rpm

python-gudev-147.2-7.el7.x86\_64.rpm

python-hwdata-1.7.3-4.el7.noarch.rpm

rhn-check-2.0.2-24.el7.x86\_64.rpm

rhn-client-tools-2.0.2-24.el7.x86\_64.rpm

rhnlib-2.5.65-8.el7.noarch.rpm

rhnsd-5.0.13-10.el7.x86\_64.rpm

rhn-setup-2.0.2-24.el7.x86\_64.rpm

yum-rhn-plugin-2.0.1-10.el7.noarch.rpm

**Use Google/Search and download/locate path to file and save RPm into /tmp/package**

**GET SOFTWARE REQUIRED (Ansible)**

**================================================================================**

**Package Arch Version Repository Size**

**================================================================================**

Ansible noarch 2.9.1-1.el7 centos-7-base-x86\_64 17 M

**Installing for dependencies:**

python-babe noarch 0.9.6-8.el7 base 1.4 M

python-httplib2 noarch 0.9.2-1.el7 centos-7-base-x86\_64 115 k

python-jinja2 noarch 2.7.2-4.el7 base 519 k

python-markupsafe x86\_64 0.11-10.el7 base 25 k

python-paramiko noarch 2.1.1-9.el7 base 269 k

python2-jmespath noarch 0.9.0-3.el7 centos-7-base-x86\_64 39 k

sshpass x86\_64 1.06-2.el7 centos-7-base-x86\_64 21 k

**Use Google/Search and download/locate path to file and save RPM into /tmp/package**

As I already mentioned I will also install Puppet Opensource client on my machines to install and configure my machines as needed. So I need to implement additional packages on the custom boot ISO to be able to install and run Puppet manifests.  
  
mkdir /tmp/packages  
cd /tmp/packages  
wget http://mirror.centos.org/centos/7/os/x86\_64/Packages/libselinux-ruby-2.2.2-6.el7.x86\_64.rpm  
wget -e robots=off --mirror --no-parent --no-host-directories --cut-dirs=4 http://yum.puppetlabs.com/el/7/products/x86\_64/  
wget -e robots=off --mirror --no-parent --no-host-directories --cut-dirs=4 http://yum.puppetlabs.com/el/7/dependencies/x86\_64/  
rm -Rf index\* repodata  
  
**2.4 Copy all your packages and create repodata**  
Now it`s time to bring the CentOS packages and your additional needed packages together. First copy all packages from CentOS 7 ISO to your local folder. Afterwards you can move all additional packages from /tmp/packages to your kickstart packages folder:  
  
rsync -av /mnt/centos7 /var/satellite/kickstart\_build/isolinux/Packages/  
rsync -av /tmp/packages/ /var/satellite/kickstart\_build/isolinux/Packages/  
  
Now we need to create the repodata folder in /var/satellite/kickstart\_build/isolinux/Packages/:

**[CMD]**  
yum install -y createrepo  
cd /var/satellite/kickstart\_build/isolinux  
createrepo -g /var/satellite/kickstart\_build/comps.xml .

**[OUTPUT]**

[root@lnlxspw01 isolinux]# createrepo -g /var/satellite/kickstart\_build/comps.xml .

Spawning worker 0 with 4080 pkgs

Workers Finished

Saving Primary metadata

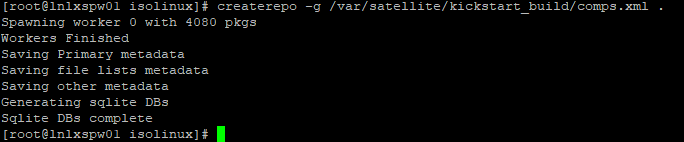
Saving file lists metadata

Saving other metadata

Generating sqlite DBs

Sqlite DBs complete

**E.g.**



**3) Prepare Kickstart file**

**3.1 Create a kickstart**  
You need to create the kickstart file in /var/satellite/kickstart\_build/isolinux/ks and name it for example ks.cfg. The content can look like this:  
  
#version=RHEL7  
# System authorization information  
auth --enableshadow --passalgo=sha512  
# Use CDROM installation media  
cdrom  
# Use text install  
install  
text  
# Run the Setup Agent on first boot  
firstboot --disable  
#ignoredisk --only-use=sda  
# Keyboard layouts

keyboard --vckeymap=gb --xlayouts='gb'

# System language

lang en\_GB.UTF-8

# Network information  
network --bootproto=static --device=eth0 --noipv6 --activate --ip=192.168.130.67 --netmask=255.255.255.0 --gateway=192.168.130.1 --nameserver=192.168.130.2 --hostname=lnlxtst01   
# Root password  
rootpw --iscrypted "some-crypted-password"  
# System timezone  
timezone Europe/Berlin --isUtc  
# System bootloader configuration  
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda  
# Partition clearing information  
clearpart --all --initlabel  
# Disk partitioning information  
part /boot --fstype="xfs" --size=512  
part pv.219 --fstype="lvmpv" --size 1 --grow  
volgroup vg\_system --pesize=4096 pv.219  
logvol / --fstype="xfs" --size=1 --grow --label="rootlv" --name=rootlv --vgname=vg\_system  
logvol swap --fstype="swap" --size=2048 --name=swaplv --vgname=vg\_system  
reboot  
%packages  
@core  
@Base  
kexec-tools  
git  
mc  
screen  
#puppet  
tree  
%end  
  
**You can validate your kickstart file like this:  
  
ksvalidator /var/satellite/kickstart\_build/isolinux/ks/ks.cfg**  
To create a crypted root password which you can use within your kickstart file do the following:  
  
python -c 'import crypt; print(crypt.crypt("My Password", "$6$My Salt"))'  
  
This generates a SHA512 crypted password.

* 1. **Create kickstart postinstallation section**  
     If you want to perform some postinstallation tasks within your kickstart installation you can add an appropriate section in the kickstart file. In my case I want to perform my Puppet configuration during the kickstart installation.  
     **HINT:** Be careful. In this case the first step will be the copy of the Puppet manifests from the ISO from which you boot your machine you want to kickstart. This action takes place in the NON-CHROOTED environment. The second step will be the Puppet run itself. This takes place in the CHROOTED environment.  
     Add the following at the end of your kickstart file:  
       
     # Copy needed Puppet files to /root/postinstall  
     %post --nochroot  
     #!/bin/sh  
     set -x -v  
     exec 1>/mnt/sysimage/root/kickstart-stage1.log 2>&1  
     echo "==> copying files from media to install drive..."  
     cp -r /run/install/repo/postinstall /mnt/sysimage/root  
     %end  
     %post  
     #!/bin/sh  
     set -x -v  
     exec 1>/root/kickstart-stage2.log 2>&1  
     ls -l /root/postinstall  
     puppet apply -l /root/puppetrun.log /root/postinstall/puppet/manifests/site.pp --modulepath=/root/postinstall/puppet/modules/ $\*  
     %end
  2. **Validate the ks.cfg**

# /bin/ksvalidator ks.cfg

**4) Time for action: create your custom CentOS 7 ISO file and test it**

yum install -y genisoimage  
cd /var/satellite/kickstart\_build/  
mkisofs -o centos-7-custom.iso -b isolinux.bin -c boot.cat -no-emul-boot -V 'CentOS 7 x86\_64' -boot-load-size 4 -boot-info-table -R -J -v -T isolinux/  
  
Now start a new virtual machine from your custom CentOS 7 ISO file and insert the following option at kernel boot:  
  
linux inst.ks=cdrom:/dev/cdrom:/ks/ks.cfg  
  
**Congratulations http://www.frankreimer.de/?p=522**

# **Git Server Setup**

Previously git server was hosted on it’s own server (git01) decision was made to host on the new admin server lnlxspw01 (alias git01 will be added to DNS)

# yum install git (package git-1.8.3.1-11.el7.x86\_64 or later)

# adduser

User: giteca:x:1001:1001::/var/git/giteca:/bin/bash

# addgroup

gitusers:x:1001:

**Giteca ssh config (not under .ssh)**

**[FILE]**

Copy/Clone existing **/etc/ssh/gitusers-keys/giteca**

**[FILE]**

**/etc/ssh/sshd\_config** – add these lines (ensure “>” and white spaces are removed)

> Match Group gitusers

> ForceCommand /usr/bin/git-shell -c "$SSH\_ORIGINAL\_COMMAND"

> PubkeyAuthentication yes

> AuthorizedKeysFile /etc/ssh/gitusers-keys/%u

> PasswordAuthentication no

> PermitTunnel no

> AllowAgentForwarding no

> AllowTcpForwarding no

> X11Forwarding no

**[CMD]**

# systemctl enable sshd

# systemctl restart sshd

# systemctl status sshd

## Git – add new users

As new user (on their system)

**[CMD]**

# ssh-keygen (accept defaults/empty passphrase)

# cat ~/.ssh/rsa\_key.pub (if using putty highlight the test (ensure no white space at end of line)

On lnlxspw01/git01 host

# cp /etc/ssh/gitusers-keys/giteca /etc/ssh/gitusers-keys/giteca.YYYYMMDD

# vi /etc/ssh/gitusers-keys/giteca # move to bottom of file, open a new line (o) and paste (right click if using putty) Save file Esc, Esc : wq!

|  |  |  |
| --- | --- | --- |
| Related documents | | |
|  |  | |
|  |  | |
|  | |  |