MSAT 0.3

msat-guide

MSAT guide

Publican

BOOK PUBLISHING TOOL

Allard Berends

MSAT 0.3 msat-guide
MSAT guide
Edition 0.1

Author Allard Berends `msat.disruptivefoss@gmail.com`

This guide explains how to setup a Spacewalk server to demonstrate the functioning of the MSAT scripts. The MSAT scripts use the Spacewalk API to automatically fill the Spacewalk server. The MSAT scripts have been created to automate the process of defining system types. The goal is to make the mechanisme efficient, to avoid errors and be versionable with system definitions.

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to the first virtual terminal. Press **`Ctrl`**+**`Alt`**+**`F1`** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click

[1] https://fedorahosted.org/liberation-fonts/

> **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;
```

```
public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

Although we put time and effort in this guide, it is very possible that errors have slipped in. We like to hear about them on *msat.disruptivefoss@gmail.com* . Please put **msat-guide error** in the subject field of the email.

Furthermore, for suggestions, we would also like to hear from you on *msat.disruptivefoss@gmail.com* . Please put **msat-guide suggestion** in the subject field of the email.

Spacewalk

This chapter discusses how to setup a Spacewalk server to test the MSAT scripts. Without a Spacewalk server (or Satellite server for that matter), it is hard to understand what the MSAT scripts do. Experimenting with the scripts is, in our opinion, very important.

Get a working Spacewalk server first

Setting up a Spacewalk server is cumbersome. Hence, we advice the reader to strictly follow the instructions given. Once a working Spacewalk server is at your disposal, it is easier to tweak or customize.

In this chapter, we ask the reader to download items. We give an estimation of the download time. It is based on our own tests, but downloads can depend on a number of things: the speed of your local ISP (Internet Service Provider) and the speed of a server from which you download. So, we did a test via *www.speedtest.net*[1]. The results were:

Table 1.1. Speedtest results

test	speed
ping	6 ms
download	7.71 Mbps
upload	0.84 Mbps

Setting up a spacewalk server takes a number of steps. We mention them here and elaborate on them in later sections. We think that it is important to give the recipe here in order to get the concept accross:

1. Download the Fedora 18 DVD ISO image and store it on the Linux host on which we install the Spacewalk KVM machine.
2. Prepare a Linux host on which we can install a Fedora 18 x86_64 KVM machine. On the host, enough memory (>4 GB) and disk space (> 300 GB, initially 80 GB will do), must be made available to the Spacewalk node.
3. Install the Fedora 18 distribution on the KVM machine and configure the system.
4. Install Spacewalk 1.9 on the Fedora 18 KVM machine and configure Spacewalk.
5. Create a minimal working Spacewalk server by filling it with CentOS 5.8 RPM's and creating a kickstartable tree.
6. Test the Spacewalk server by provisioning a simple CentOS 5.8 system. By provisioning we mean: automatic installation, registration and configuration.

In the remainder of the chapter we discuss the steps. We keep the amount of information limited. In appendix ... we show screen dumps and extra information.

[1] http://www.speedtest.net/

1.1. Prerequisites

Table 1.2. Time needed

action	time
Fedora 18 download	76 minutes

On the host system, make a location where the downloaded ISO can be stored:

```
server# mkdir /var/isos
```

- Download the Fedora 18 DVD, see *Download Fedora 18*[2] and store it in **`/var/isos`**.

After the download, our **`/var/isos`** is:

```
server# ls -l /var/isos | grep Fedora-18
-rw-------. 1 qemu qemu 4573888512 Jan 15 20:30 Fedora-18-x86_64-DVD.iso
```

1.2. KVM host configuration

Table 1.3. Time needed

action	time
virtual network creation	10 minutes
storage creation	10 minutes
memory verification	3 minutes

In this section we discuss what needs to be done on the host system. For the Spacewalk KVM, we need:

- A virtual network in which we let the Spacewalk server operate. In this network, we also add our test machine. This is the machine on which we test the provisioning.
- Disk storage. Eventually we want to have CentOS 5 and 6 content including all the minor versions. Furthermore, we want to have some space available for making exports and imports when the occasion arises. So 300GB should suffice.
- Memory for the virutal CPU. At least 2GB is needed. To give the Spacewalk server some power we use 4GB.

1.2.1. Virtual network

We assume that most people use `192.168.1.0/24` as their home network. Since we don't want to interfere, we use `192.168.5.0/24`. This is a virtual network that needs to be created on the host. For more background information look at *libvirt.org/formatnetwork.html*[3].

[2] http://download.fedoraproject.org/pub/fedora/linux/releases/18/Fedora/x86_64/iso/Fedora-18-x86_64-DVD.iso

[3] http://libvirt.org/formatnetwork.html

KVM virtualization must work on host

We assume that on the Linux host system KVM virtualization is working. If this is not the case, please make sure it works first. The `libvirtd` service must run. For more information, refer to *Virtualization Administration Guide*[4].

The virtual network can be created via the **`virsh`** command line tool. Another way is the **`virt-manager`** graphical tool. We first show how to use the command line tool, and then the graphical tool.

We have the following XML file:

```
server# cat dmsat1.xml
<network>
  <name>dmsat1</name>
  <forward mode='nat'/>
  <bridge name='virbr1' stp='on' delay='0'/>
  <mac address='52:54:00:A8:05:01'/>
  <domain name='dmsat1.org'/>
  <ip address='192.168.5.1' netmask='255.255.255.0'/>
</network>
```

Note that the above MAC address, `52:54:00:A8:05:01`, is build up from the rules:

- **52:54:00**, which is fixed by definition. See the section **Network Configuration** of "man virt-install".
- **A8:05:01**, which is the last 3 parts of `192.168.5.1` in hexadecimal.

Now, create the network, mark it as autostart and activate it, with:

```
server# virsh net-define dmsat1.xml
Network dmsat1 defined from test.xml

server# virsh net-autostart dmsat1
Network dmsat1 marked as autostarted

server# virsh net-start dmsat1
Network dmsat1 started
```

The command line method above, requires that one knows what must be specified in the XML file. Another way is to use the GUI, the **`virt-manager`** command. Use the following clickpath: **Edit** → **Connection Details**. The **Connetion Details** window pops up. In here, click: **Virtual Networks** → **+ button**. The **Create a new virtual network** dialog pops up. Press **Forward**. Enter **`dmsat1`** in the **Network Name:** dialog box. Press **Forward** again. The **Network:** is **`192.168.5.0/24`**. Since, we don't use DHCP, unvink the **Enable DHCP:** box. Press **Forward**. Select the **Forwarding to physical network** radio button. The **Destination:** should be **`Any physical device`** and the **Mode:** should be **`NAT`**. Press **Forward** and then **Finish**.

[4] http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html-single/Virtualization_Administration_Guide/index.html

Note that via the GUI method, one does not have the option of setting the MAC address to a specific value, like `52:54:00:A8:05:01`. Consequently, we prefer the command line method to create the virtual network.

We assume that the newly created network is the second network after the **`default`** network. Hence, it is connected to the **`virbr1`** device. The network is defined in an XML file in **`/etc/libvirt/qemu/networks`**, named **`dmsat1.xml`**. It can be shown with:

```
server# virsh net-dumpxml dmsat1
<network>
  <name>dmsat1</name>
  <forward mode='nat'/>
  <bridge name='virbr1' stp='on' delay='0'/>
  <mac address='52:54:00:A8:05:01'/>
  <domain name='dmsat1.org'/>
  <ip address='192.168.5.1' netmask='255.255.255.0'/>
</network>
```

Make sure that the network is active and that it auto-starts:

```
server# virsh net-list
Name                 State      Autostart
-----------------------------------------
default              active     yes
dmsat1               active     yes
```

1.2.2. Storage setup

The KVM Spacewalk server needs disk space to run on. We provide it by means of a LV (Logical Volume) created on the host. The creation is done as follows:

```
server# pvs
  PV         VG        Fmt  Attr PSize   PFree
  /dev/sda3  vg_server lvm2 a--  931.00g       0
  /dev/sdb   b         lvm2 a--  931.51g 931.51g
  /dev/sdc   c         lvm2 a--  931.51g 931.51g
  /dev/sdd   d         lvm2 a--  931.51g 931.51g
  /dev/sde   e         lvm2 a--  931.51g 931.51g
  /dev/sdf   f         lvm2 a--  931.51g 931.51g

server# pvs -o +vg_extent_size /dev/sdb
  PV         VG   Fmt  Attr PSize   PFree   Ext
  /dev/sdb   b    lvm2 a--  931.51g 931.51g 4.00m

server# lvcreate -n sw -l $((25 * 10 * 300)) b
  Logical volume "sw" created
```

The newly created LV, **`/dev/b/sw`**, can now be presented to the virtual machine creation command as block device for storage.

[5] https://fedorahosted.org/spacewalk

1.2.3. Memory setup

From the information of the Spacewalk project, see *fedorahosted.org/spacewalk*[5], we know from the **Spacewalk Installation Instructions**, from a Prerequisites section that the minimum amount of RAM needed, is 2GB, but that 4GB is recommended. We know that on our host this requirement is no problem. We have 16GB of internal memory, which can be demonstrated with:

```
server# cat /proc/meminfo | grep ^MemTotal
MemTotal:       16435452 kB
```

1.3. Fedora 18 installation

Table 1.4. Time needed

action	time
Fedora 18 installation	30 minutes

We use the Fedora 18 OS, as downloaded from *Download Fedora 18 x86_64 ISO*[6]. We do an interactive installation. Since we do a server installation, the Desktop parts (groups) are skipped.

Explanation of the parameters in the **`virt-install`** command:

- d19sw1, d (development), 19 (version 1.9), sw (spacewalk), 1 (first server).

- mac=52:54:00:a8:05:0b, 192.168.5.11 -> 00:a8:05:0b, the 52:54:00 is obligatory by libvirt definition.

To make it easy to verify the **`virt-install`** options in the *virt-install man page*, we specify them in the same order as they appear in the man page.

The installation is started with:

```
server# virt-install \
  --connect qemu:///system \
  --name=d19sw1 \
  --ram=4096 \
  --arch=x86_64 \
  --description="Development Spacewalk server version 1.9" \
  --cdrom /var/isos/Fedora-18-x86_64-DVD.iso \
  --os-type=linux \
  --os-variant=fedora18 \
  --disk path=/dev/b/sw,device=disk,bus=scsi \
  --network=network=dmsat1,mac=52:54:00:a8:05:0b \
  --graphics vnc \
  --hvm \
  --autostart
.. output skipped ..
```

The interactive installation is started. Answer the questions with the parameter values as specified below.

Table 1.5. Installation parameters

question	answer
language	English

[6] http://download.fedoraproject.org/pub/fedora/linux/releases/18/Fedora/x86_64/iso/Fedora-18-x86_64-DVD.iso

question	answer
Hostname	d19sw1.dmsat1.org
IPv4 settings	
Date & Time	Europe/Amsterdam
Software selection	Minimal install

With respect to partitioning, we don't need a **/home** partition. Consequently, it needs to be removed. The trick is to let the installer choose the partitioning automatically and then remove the space assigned to **/home** and add it to /. We want to end up with:

Table 1.6. Partitioning layout

device	mountpoint	size
/dev/sda1	**/boot**	500 MB
/dev/sda2	**PVS**	300 GB (remainder of disk)
/dev/fedora_d19sw1/swap	**None**	3968 MB
/dev/fedora_d19sw1/root	/	295524 MB (remainder of disk)

We choose to update the Fedora 18 installation here with the latest patches:

```
d19sw1# yum update
.. takes a while, about 10 minutes ..
```

After the installation we want to add some packages, do as root:

```
yum -y install bind-utils
yum -y install vim-enhanced
yum -y install tree
yum -y install screen
yum -y install net-tools
yum -y install tar
yum -y install mailx
yum -y install strace
yum -y install setroubleshoot
yum -y install syslinux
yum -y install bind-chroot
yum -y install mlocate
yum -y install wget
.. output of commands skipped ..
reboot
.. only if new kernel, rpm -qa | grep kernel ..
```

We want to enable the virtual console on the KVM. This means that Grub2 and the kernel must know about the serial console. Adapt the **/etc/default/grub** file to include the following lines:

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=fedora_d19sw1/swap rd.md=0 rd.dm=0 rd.lvm.lv=fedora_d19sw1/
root $([ -x /usr/sbin/rhcrashkernel-param ] && /usr/sbin/rhcrashkernel-param || :) rd.luks=0
 vconsole.keymap=us console=ttyS0,115200n8"
GRUB_TERMINAL=serial
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --stop=1"
```

Then, from the root prompt:

```
d19sw1# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-3.6.10-4.fc18.x86_64
Found initrd image: /boot/initramfs-3.6.10-4.fc18.x86_64.img
done
```

Add ttyS0 to **/etc/securetty**:

```
d19sw1# echo "ttyS0" >> /etc/securetty
```

Enable the service in Fedora 18 with systemd:

```
d19sw1# cd /etc/systemd/system/getty.target.wants
d19sw1# ln -sf /usr/lib/systemd/system/serial-getty@.service serial-getty@ttyS0.service
d19sw1# systemctl start serial-getty@ttyS0.service
```

Configure the firewall for Spacewalk and DNS, as `root`:

```
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --permanent --add-service=dns
firewall-cmd --permanent --add-port=5222/tcp
firewall-cmd --permanent --add-port=5269/tcp
firewall-cmd --reload
firewall-cmd --list-all
public
  interfaces: eth0
  services: mdns http dhcpv6-client https ssh dns
  ports: 6269/tcp 5222/tcp
  forward-ports:
  icmp-blocks:
```

1.4. DNS configuration

Table 1.7. Time needed

action	time
DNS configuration	10 minutes

First of all, to be able to configure DNS, we need the software, if not yet available:

```
d19sw1# yum install bind-chroot
.. output skipped ..
```

We want to set up the Spacewalk server as DNS server. In order for Spacewalk to function correctly, one needs DNS. During registration of provisioned hosts, Spacewalk does a reverse DNS query for the hostname of the registering host. In this section we describe how we configure DNS in the *dmsat1* environment.

We want DNS to behave in the following way:

- All DNS queries outside our domain, `dmsat1.org`, must go to the upstream DNS server, `192.168.1.1`. Consequently, we are a forwarding DNS server. The result queries must be cached.
- We want logging.

- The Spacewalk server is the name authority for `dmsat1.org`.

Since we also want to keep track of the virtual MAC addresses, we use the main DNS zone file for keeping track of this administration.

We just show all the DNS files here and explain afterwards how the directory structure looks:

```
d19sw1# cat /var/named/chroot/etc/named.conf
// /etc/named.conf

options {
  /*
   * Check names following RFC 952 and RFC 1123. The
   * master has strict checking. Incorrect names make
   * the nameserver fail. The slaves receive the names
   * from the master, hence those names are not
   * checked.
   */
  check-names master fail;
  check-names slave ignore;
  listen-on port 53 {
    192.168.5.11;
    127.0.0.1;
  };
  //listen-on-v6 port 53 { ::1; };
  directory   "/var/named";
  dump-file   "/var/named/data/cache_dump.db";
  statistics-file "/var/named/data/named_stats.txt";
  memstatistics-file "/var/named/data/named_mem_stats.txt";
  /*
   * To start out, we want all hosts to be able to
   * query this nameserver.
   */
  //allow-query     { localhost; };
  allow-query     { any; };
  //allow-query-on  { any; };
  //allow-query-on  { 192.168.1.16; 127.0.0.1; };
  recursion yes;
};

/*
 * Allowable log levels:
 * critical
 * error
 * warning
 * notice
 * info
 * debug [level]
 * dynamic
 */

/*
 * Allowable log categories:
 * client
 * config
 * database
 * default (all categories except queries)
 * delegation-only
 * dispatch
 * dnssec
 * general
 * lame-servers
 * network
 * notify
 * queries
 * resolver
 * security
 * unmatched
 * update
 * update-security
 * xfer-in
 * xfer-out
 */

logging {
  channel default_debug {
    file "/var/log/named.log" versions 5 size 5m;
    severity dynamic;
    print-time yes;
    print-severity yes;
    print-category yes;
  };

  category default { default_debug; };
  category queries { default_debug; };
};

/*
 * The '.' represents all other zones. We depend on the
 * upstream DNS server, 192.168.1.1, so all requests go
 * there. The answers are cached.
 */
zone "." IN {
  type forward;
  forward first;
  forwarders { 192.168.1.1; };
};

/*
 * Instead of including named.rfc1912.zones, we define our
 * localhost zones ourselves below.
 */
//include "/etc/named.rfc1912.zones";

zone "localhost.localdomain." IN {
  type master;
  file "named.localhost";
  allow-update { none; };
};

zone "localhost." IN {
  type master;
  file "named.localhost";
  allow-update { none; };
};

zone "127.in-addr.arpa." IN {
  type master;
  file "127.in-addr.arpa";
  allow-update { none; };
};

zone "dmsat1.org." IN {
  type master;
  file "dmsat1.org.zone";
  notify yes;
  allow-transfer { any; };
};

zone "5.168.192.in-addr.arpa." IN {
  type master;
  file "192.168.5.in-addr.arpa";
  notify yes;
  allow-transfer { any; };
};
```

```
d19sw1# cat /var/named/chroot/var/named/127.in-addr.arpa
$TTL    86400
@       IN SOA d19sw1.dmsat1.org. root.d19sw1.dmsat1.org. (
        2012012300      ; serial, format: YYYYMMDDNN, where
      ; NN is the daily sequence number.
      ; After editing this file, always
      ; update this serial number to give
      ; slaves a chance to keep
      ; synchronized.
        3H              ; refresh. The refresh interval
      ; tells a slave for the zone how
      ; often to check that the data for
      ; this zone is up to date.
        15M             ; retry. If a slave fails to connect
      ; to the master, it will retry in
      ; this amount of time. So here it is
      ; 15 minutes.
        1W              ; expiry. If a slave did not contact
      ; the server for this amount of
      ; time, its records become useless,
      ; i.e. are expired.
        1D )            ; minimum. TTL for negative
      ; responses of this nameserver for
      ; authoriative requests. So the
      ; client must not ask for the same
      ; RR (Resource Record) it got a
      ; negative response for in the
      ; specified amount of time.

;
; @ is implied!
;
        IN NS   d19sw1.dmsat1.org.

;
; 127.0.0.0/8
;
; Satellite.
1.0.0   IN  PTR localhost.localdomain.
```

```
d19sw1# cat /var/named/chroot/var/named/192.168.5.in-addr.arpa
$TTL    86400
@       IN SOA d19sw1.dmsat1.org. root.d19sw1.dmsat1.org. (
        2012012300      ; serial, format: YYYYMMDDNN, where
      ; NN is the daily sequence number.
      ; After editing this file, always
      ; update this serial number to give
      ; slaves a chance to keep
      ; synchronized.
        3H              ; refresh. The refresh interval
      ; tells a slave for the zone how
      ; often to check that the data for
      ; this zone is up to date.
        15M             ; retry. If a slave fails to connect
      ; to the master, it will retry in
      ; this amount of time. So here it is
      ; 15 minutes.
        1W              ; expiry. If a slave did not contact
      ; the server for this amount of
      ; time, its records become useless,
      ; i.e. are expired.
        1D )            ; minimum. TTL for negative
      ; responses of this nameserver for
      ; authoriative requests. So the
      ; client must not ask for the same
      ; RR (Resource Record) it got a
      ; negative response for in the
      ; specified amount of time.

;
; @ (short for domain name) is implied!
;
        IN NS   d19sw1.dmsat1.org.

;
; 192.168.5.0/24
;

; Development Spacewalk server.
10  IN  PTR d19sw0.dmsat1.org.
11  IN  PTR d19sw1.dmsat1.org.
12  IN  PTR d19sw2.dmsat1.org.
100 IN  PTR dmm01.dmsat1.org.
```

```
d19sw1# cat /var/named/chroot/var/named/dmsat1.org.zone
$TTL    86400
@       IN SOA d19sw1.dmsat1.org. root.d19sw1.dmsat1.org. (
        2012012300      ; serial, format: YYYYMMDDNN, where
      ; NN is the daily sequence number.
      ; After editing this file, always
      ; update this serial number to give
      ; slaves a chance to keep
      ; synchronized.
        3H              ; refresh. The refresh interval
      ; tells a slave for the zone how
      ; often to check that the data for
      ; this zone is up to date.
        15M             ; retry. If a slave fails to connect
      ; to the master, it will retry in
      ; this amount of time. So here it is
      ; 15 minutes.
        1W              ; expiry. If a slave did not contact
      ; the server for this amount of
      ; time, its records become useless,
      ; i.e. are expired.
        1D )            ; minimum. TTL for negative
      ; responses of this nameserver for
      ; authoriative requests. So the
      ; client must not ask for the same
      ; RR (Resource Record) it got a
      ; negative response for in the
      ; specified amount of time.

;
; @ (short for domain name) is implied!
;
        IN NS   d19sw1.dmsat1.org.

;
; 192.168.5.0/24
;

;network   IN      A       192.168.5.0
;gateway   IN      A       192.168.5.1
```

```
; MAC_ETH0=52:54:00:a8:05:0a
d19sw0     IN      A       192.168.5.10
; MAC_ETH0=52:54:00:a8:05:0b
d19sw1     IN      A       192.168.5.11
; MAC_ETH0=52:54:00:a8:05:0c
d19sw2     IN      A       192.168.5.12
; MAC_ETH0=52:54:00:a8:05:64
dmm01      IN      A       192.168.5.100
;broadcast IN      A       192.168.5.255
```

```
d19sw1# cat /var/named/chroot/var/named/named.localhost
$TTL    86400
@       IN SOA @ rname.invalid. (
        2012012300      ; serial, format: YYYYMMDDNN, where
      ; NN is the daily sequence number.
      ; After editing this file, always
      ; update this serial number to give
      ; slaves a chance to keep
      ; synchronized.
        3H              ; refresh. The refresh interval
      ; tells a slave for the zone how
      ; often to check that the data for
      ; this zone is up to date.
        15M             ; retry. If a slave fails to connect
      ; to the master, it will retry in
      ; this amount of time. So here it is
      ; 15 minutes.
        1W              ; expiry. If a slave did not contact
      ; the server for this amount of
      ; time, its records become useless,
      ; i.e. are expired.
        1D )            ; minimum. TTL for negative
      ; responses of this nameserver for
      ; authoriative requests. So the
      ; client must not ask for the same
      ; RR (Resource Record) it got a
      ; negative response for in the
      ; specified amount of time.

;
; @ is implied!
;
        NS  @
    A   127.0.0.1
    AAAA    ::1
```

In a directory tree, the structure looks like this:

```
d19sw1# tree --charset=ASCII -F /var/named/chroot
/var/named/chroot
|-- dev/
|   |-- null
|   |-- random
|   `-- zero
|-- etc/
|   |-- localtime
|   |-- named/
|   |-- named.conf
|   |-- named.iscdlv.key
|   |-- named.rfc1912.zones
|   |-- named.root.key
|   |-- pki/
|   |   `-- dnssec-keys/
|   `-- rndc.key
|-- usr/
|   `-- lib64/
|       `-- bind/
`-- var/
    |-- log/
    |   `-- named.log
    |-- named/
    |   |-- 127.in-addr.arpa
    |   |-- 192.168.5.in-addr.arpa
    |   |-- dmsat1.org.zone
    |   |-- named.ca
    |   |-- named.empty
    |   |-- named.localhost
    |   `-- named.loopback
    |-- run/
    |   `-- named/
    |       |-- named.pid
    |       `-- session.key
    `-- tmp/
```

Reconfigure the **`/etc/resolv.conf`** so that the `d19sw1.dmsat1.org` also resolvs to its own DNS:

```
d19sw1# sed -i 's/^DNS1=.*$/DNS1=192.168.5.11/' /etc/sysconfig/network-scripts/ifcfg-eth0
.. inotify informs NetworkManager ..
d19sw1# cat /etc/resolv.conf
# Generated by NetworkManager
search dmsat1.org
nameserver 192.168.5.11
```

If the change is not picked up, do:

```
d19sw1# systemctl restart NetworkManager.service
```

Make sure to enable the service and start it:

```
d19sw1# systemctl enable named-chroot.service
ln -s '/usr/lib/systemd/system/named-chroot.service' '/etc/systemd/system/multi-
user.target.wants/named-chroot.service'
d19sw1# systemctl start named-chroot.service
```

Do some tests:

```
d19sw1# for i in d19sw0.dmsat1.org d19sw1.dmsat1.org d19sw2.dmsat1.org dmm01.dmsat1.org
 d19sw0 d19sw1 d19sw2 dmm01 192.168.5.10 192.168.5.11 192.168.5.12 192.168.5.100; do host $i;
 done
d19sw0.dmsat1.org has address 192.168.5.10
d19sw1.dmsat1.org has address 192.168.5.11
d19sw2.dmsat1.org has address 192.168.5.12
dmm01.dmsat1.org has address 192.168.5.100
d19sw0.dmsat1.org has address 192.168.5.10
d19sw1.dmsat1.org has address 192.168.5.11
d19sw2.dmsat1.org has address 192.168.5.12
dmm01.dmsat1.org has address 192.168.5.100
10.5.168.192.in-addr.arpa domain name pointer d19sw0.dmsat1.org.
11.5.168.192.in-addr.arpa domain name pointer d19sw1.dmsat1.org.
12.5.168.192.in-addr.arpa domain name pointer d19sw2.dmsat1.org.
100.5.168.192.in-addr.arpa domain name pointer dmm01.dmsat1.org.
```

1.5. Spacewalk installation

Table 1.8. Time needed

action	time
Spacewalk installation	35 minutes
Spacewalk configuration	15 minutes

The information in this section is obtained from *fedorahosted.org/spacewalk/wiki/HowToInstall*[7] and our own experimenting.

Because we experienced some problems with SELinux, we switch it off during the installation. Afterwards, we generate a new policy to ensure that Spacewalk run with SELinux in **enforcing** mode. So, now set:

```
d19sw1# vim /etc/selinux/config
d19sw1# grep ^SELINUX= /etc/selinux/config
SELINUX=permissive
d19sw1# setenforce 0
```

We first show the screen dumps of the installation. At some points we skip output. Otherwise a lot of space (about 4000 lines) is consumed. When noticable, but ignorable errors occur, we show them. Packages that just install well are skipped in the output.

```
d19sw1# rpm -Uvh http://yum.spacewalkproject.org/1.9/Fedora/18/x86_64/spacewalk-
repo-1.9-1.fc18.noarch.rpm
Retrieving http://yum.spacewalkproject.org/1.9/Fedora/18/x86_64/spacewalk-
repo-1.9-1.fc18.noarch.rpm
warning: /var/tmp/rpm-tmp.R8ZEmI: Header V4 DSA/SHA1 Signature, key ID 863a853d: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:spacewalk-repo-1.9-1.fc18        ################################# [100%]
```

```
d19sw1# cat > /etc/yum.repos.d/jpackage-generic.repo << EOF
[jpackage-generic]
name=JPackage generic
#baseurl=http://mirrors.dotsrc.org/pub/jpackage/5.0/generic/free/
mirrorlist=http://www.jpackage.org/mirrorlist.php?dist=generic&type=free&release=5.0
enabled=1
gpgcheck=1
gpgkey=http://www.jpackage.org/jpackage.asc
EOF
```

```
d19sw1# yum install spacewalk-setup-embedded-postgresql
.
.
================================================================================
 Package                              Arch    Version          Repository  Size
================================================================================
Installing:
 spacewalk-setup-embedded-postgresql  noarch  1.9.1-1.fc18     spacewalk   12 k
```

[7] https://fedorahosted.org/spacewalk/wiki/HowToInstall

```
Installing for dependencies:
 postgresql                          x86_64  9.2.3-1.fc18     updates    3.2 M
 postgresql-libs                     x86_64  9.2.3-1.fc18     updates    221 k
 postgresql-server                   x86_64  9.2.3-1.fc18     updates    3.6 M

Transaction Summary
================================================================================
Install  1 Package (+3 Dependent packages)

Total download size: 7.0 M
Installed size: 33 M
Is this ok [y/N]: y
.
.
```

Note, the following step takes about 30 minutes!

```
d19sw1# yum install spacewalk-postgresql
.
.
Transaction Summary
================================================================================
Install  1 Package  (+478 Dependent packages)
Upgrade             (  10 Dependent packages)

Total download size: 209 M
Is this ok [y/N]: y
Downloading Packages:
(1/489): MessageQueue-3.26.7-1.fc18.noarch.rpm             |  18 kB   00:00
.
.
Retrieving key from http://www.jpackage.org/jpackage.asc
Importing GPG key 0xC431416D:
 Userid     : "JPackage Project (JPP Official Keys) <jpackage@zarb.org>"
 Fingerprint: 1f81 c0fb c2b8 22b3 de12 33a4 5c6c fff7 c431 416d
 From       : http://www.jpackage.org/jpackage.asc
Is this ok [y/N]: y
.
.
  Installing : java-1.5.0-gcj-1.5.0.0-40.fc18.x86_64                    277/499
/var/tmp/rpm-tmp.Cn2QAL: line 11: /usr/share/java-utils/abs2rel.sh: No such file or directory
/var/tmp/rpm-tmp.Cn2QAL: line 18: /usr/share/java-utils/abs2rel.sh: No such file or directory
/var/tmp/rpm-tmp.Cn2QAL: line 24: /usr/share/java-utils/abs2rel.sh: No such file or directory
/var/tmp/rpm-tmp.Cn2QAL: line 28: /usr/share/java-utils/abs2rel.sh: No such file or directory
  Installing : sinjdoc-0.5-13.fc18.x86_64                               278/499
.
.
```

At this point we have the Spacewalk software installed. Now we will configure it. Again, we show this in screen dumps.

Follow the instructions exactly

It turns out that deviating from the installation instructions can result in a lot of trouble. Here, we want to make sure that we end up with a working Spacewalk server. Tweaking and experimenting can be done later!

```
d19sw1# passwd postgres
Changing password for user postgres.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

Completing the following section takes about 5 minutes!

```

d19sw1# spacewalk-setup --disconnected
** Database: Setting up database connection for PostgreSQL backend.
** Database: Installing the database:
** Database: This is a long process that is logged in:
** Database:   /var/log/rhn/install_db.log
*** Progress: ###
** Database: Installation complete.
** Database: Populating database.
*** Progress: ###################################
* Setting up users and groups.
** GPG: Initializing GPG and importing key.
** GPG: Creating /root/.gnupg directory
You must enter an email address.
Admin Email Address? root@localhost
* Performing initial configuration.
* Activating Spacewalk.
** Loading Spacewalk Certificate.
** Verifying certificate locally.
** Activating Spacewalk.
* Enabling Monitoring.
* Configuring apache SSL virtual host.
Should setup configure apache's default ssl server for you (saves original ssl.conf)
 [Y]? <Enter>
** /etc/httpd/conf.d/ssl.conf has been backed up to ssl.conf-swsave
* Configuring tomcat.
** /etc/sysconfig//tomcat has been backed up to tomcat-swsave
** /etc/tomcat//server.xml has been backed up to server.xml-swsave
** /etc/tomcat//web.xml has been backed up to web.xml-swsave
* Configuring jabberd.
* Creating SSL certificates.
CA certificate password? <redhat>
Re-enter CA certificate password? <redhat>
Organization? Example
Organization Unit [d19sw1.dmsat1.org]? dmsat1
Email Address [root@localhost]? <Enter>
City? Utrecht
State? Utrecht
Country code (Examples: "US", "JP", "IN", or type "?" to see a list)? NL
** SSL: Generating CA certificate.
** SSL: Deploying CA certificate.
** SSL: Generating server certificate.
** SSL: Storing SSL certificates.
* Deploying configuration files.
* Update configuration in database.
* Setting up Cobbler..
Processing /etc/cobbler/modules.conf
‘/etc/cobbler/modules.conf’ -> ‘/etc/cobbler/modules.conf-swsave’
Processing /etc/cobbler/settings
‘/etc/cobbler/settings’ -> ‘/etc/cobbler/settings-swsave’
cobblerd does not appear to be running/accessible
Cobbler requires tftp and xinetd services be turned on for PXE provisioning functionality.
 Enable these services [Y]? Enter
ln -s '/usr/lib/systemd/system/tftp.socket' '/etc/systemd/system/sockets.target.wants/
tftp.socket'
cobblerd does not appear to be running/accessible
* Restarting services.
```

```
*******************************************************************
 Using the default of SSL_verify_mode of SSL_VERIFY_NONE for client
 is depreciated! Please set SSL_verify_mode to SSL_VERIFY_PEER
 together with SSL_ca_file|SSL_ca_path for verification.
 If you really don't want to verify the certificate and keep the
 connection open to Man-In-The-Middle attacks please set
 SSL_verify_mode explicitly to SSL_VERIFY_NONE in your application.
*******************************************************************
  at /usr/share/perl5/LWP/Protocol/http.pm line 31.
*******************************************************************
 Using the default of SSL_verify_mode of SSL_VERIFY_NONE for client
 is depreciated! Please set SSL_verify_mode to SSL_VERIFY_PEER
 together with SSL_ca_file|SSL_ca_path for verification.
 If you really don't want to verify the certificate and keep the
 connection open to Man-In-The-Middle attacks please set
 SSL_verify_mode explicitly to SSL_VERIFY_NONE in your application.
*******************************************************************
  at /usr/share/perl5/LWP/Protocol/http.pm line 31.
Installation complete.
Visit https://d19sw1.dmsat1.org to create the Spacewalk administrator account.
```

The installation program points us to the Spacewalk's web interface: *http://192.168.5.11*. Here we need to configure the Spacewalk administrator account. We use the following set of parameters:

Table 1.9. Spacewalk admin parameters

parameter	value
Desired Login:	example
Desired Password:	redhat
Confirm Password:	redhat
Firt, Last Name:	Mr. Example Example
Email:	root@localhost

Now click Create Login. Remove **HTTP proxy settings** and uncheck **Disconnected Spacewalk** and press Update.

The Spacewalk server mentions on the web interface that we need to restart the Spacewalk server:

```
d19sw1# spacewalk-service restart
.. output skipped ..
```

In our case the Spacewalk certificate was still missing, so we copy it:

```
d19sw1# cd ~/ssl-build
d19sw1# cp RHN-ORG-TRUSTED-SSL-CERT /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
```

We also want the Spacewalk service to start automatically at system boot:

```
d19sw1# systemctl enable spacewalk.target
ln -s '/usr/lib/systemd/system/spacewalk.target' '/etc/systemd/system/multi-
user.target.wants/spacewalk.target'
```

The Cobbler daemon still needs some adjustments. Change the following parameters in **`/etc/cobbler/settings`**:

```
d19sw1# vim /etc/cobbler/settings
.
.
allow_duplicate_hostnames: 1
allow_duplicate_ips: 1
allow_duplicate_macs: 1
redhat_management_permissive: 1
.
.
d19sw1# systemctl restart cobblerd.service
```

This concludes our Spacewalk installation. The Spacewalk server is still in a state that we cannot provision systems. Hence, it needs filling!

1.6. Initial Spacewalk Filling

Table 1.10. Time needed

action	time
Download DVD1	77 minutes
Download DVD2	12 minutes
Download Spacewalk client tools	6 minutes

In this section we explain how we fill the Spacewalk server. Initially we fill it with a minimal set of RPM's to make it work.

This raises the question, what is a functional Spacewalk server? We think that we have a minimal functioning Spacewalk server if:

- it can be used to install the OS on a system. The system may be a KVM machine.
- the installed system automatically registers to the Spacewalk server. I must show up in the list of systems of the Spacewalk web interface.
- the installed system automatically has access to configured software channel, besides the base channel. In other words: "yum repolist" on the installed system should give the configured software channels.
- the installed system automatically obtains RPM's from connected activation keys, and config files from connected config channels.

To make the Spacewalk server function, we need to add RPM's from a Linux distribution. For this purpose, we use CentOS 5.8. Hence, we need to create a base channel and put RPM's into it. For registration and configuration we need some tools. We put these in a child channel. Finally, we need to create a kickstartable tree to do bare metal deployment. We show the steps in the following sections.

1.6.1. Base channel

First, we create the base channel. Point the browser to *http://192.168.5.11* and do: **Sign In** → **Spacewalk Login: example** → **Password: redhat+Sign In** → **Channels** → **Manage Software Channels** → **create new channel** → **Channel Name: centos-x86_64-server-5** → **Channel Label: centos-x86_64-server-5** → **Parent Channel: None** → **Architecture: x86_64** → **Yum Repository**

Checksum Type: sha1 → Channel Summary: centos-x86_64-server-5 → GPG key URL: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5+Create Channel

The newly created channel needs to be filled. We use the following commands:

```
d19sw1# cd
server# wget http://mirror.nsc.liu.se/centos-store/5.8/isos/x86_64/CentOS-5.8-x86_64-bin-
DVD-1of2.iso
.. output skipped ..
.. 77 minutes ..
server# wget http://mirror.nsc.liu.se/centos-store/5.8/isos/x86_64/CentOS-5.8-x86_64-bin-
DVD-2of2.iso
.. output skipped ..
.. 12 minutes ..
d19sw1# wget http://mirror.nsc.liu.se/centos-store/5.8/isos/x86_64/md5sum.txt
.. output skipped ..

d19sw1# md5sum CentOS-5.8-x86_64-bin-DVD-1of2.iso
8a3bf0030f192022943f83fe6b2cf373  CentOS-5.8-x86_64-bin-DVD-1of2.iso
d19sw1# grep CentOS-5.8-x86_64-bin-DVD-1of2.iso md5sum.txt
8a3bf0030f192022943f83fe6b2cf373  CentOS-5.8-x86_64-bin-DVD-1of2.iso

d19sw1# md5sum CentOS-5.8-x86_64-bin-DVD-2of2.iso
1d0d05edeb310c99a48fbef075695bb3  CentOS-5.8-x86_64-bin-DVD-2of2.iso
d19sw1# grep CentOS-5.8-x86_64-bin-DVD-2of2.iso md5sum.txt
1d0d05edeb310c99a48fbef075695bb3  CentOS-5.8-x86_64-bin-DVD-2of2.iso

d19sw1# mkdir -p /var/www/html/pub/iso/{1,2}
d19sw1# mount -o loop /root/CentOS-5.8-x86_64-bin-DVD-1of2.iso /var/www/html/pub/iso/1
d19sw1# mount -o loop /root/CentOS-5.8-x86_64-bin-DVD-2of2.iso /var/www/html/pub/iso/2

d19sw1# spacewalk-repo-sync -c centos-x86_64-server-5 -u http://192.168.5.11/pub/iso/1
.. output skipped ..
3536/3595 : openoffice.org-langpack-hr_HR-3.1.1-19.5.el5_5.6-1.x86_64
failure: CentOS/openoffice.org-langpack-hr_HR-3.1.1-19.5.el5_5.6.x86_64.rpm from centos-
x86_64-server-5: [Errno 256] No more mirrors to try.
http://192.168.5.11/pub/iso/1/CentOS/openoffice.org-langpack-
hr_HR-3.1.1-19.5.el5_5.6.x86_64.rpm: [Errno 14] curl#22 - "The requested URL returned error:
 404 Not Found"
.. output skipped ..
.. 4 hours and 30 minutes ..
d19sw1# rhnpush -d /var/www/html/pub/iso/2/CentOS/ -c centos-x86_64-server-5 -u example -p
 redhat --server $(hostname)
.. output skipped ..
.. 9 minutes ..
```

The langpack errors can be ignored. Via the **`rhnpush`** command, these packages are added.

1.6.2. Tools child channel

Just using CentOS 5.8 will not do. Believe us, we learned the hard way! We also need extra RPM's to make automatic registration and configuration work. Hence, a second software channel is created.

Sign In → Spacewalk Login: example → Password: redhat+Sign In → Channels → Manage Software Channels → create new channel → Channel Name: con-tools-centos-x86_64-server-5 → Channel Label: con-tools-centos-x86_64-server-5 → Parent Channel: centos-x86_64-server-5 → Architecture: x86_64 → Yum Repository Checksum Type: sha1 → Channel Summary: con-tools-centos-x86_64-server-5 → GPG key URL: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-spacewalk-2012+Create Channel

A software channel name cannot start with rhn

It is not possible to use the name **rhn-tools-centos-x86_64-server-5**. Spacewalk forbids the use of **rhn**. It is reserved for the Satellite server of Red Hat.

We fill the tools channel with:

```
d19sw1# spacewalk-repo-sync -c con-tools-centos-x86_64-server-5 -u http://
spacewalk.redhat.com/yum/1.9-client/RHEL/5/x86_64
.. less than a minute ..
```

During one of the test installations (happens when `rhnreg_ks` is run) we found out that the **con-tools-centos-x86_64-server-5** channel is incomplete. We need an extra RPM called *python-hashlib-20081119-4.el5.x86_64.rpm* because *spacewalk-backend-libs-1.9.48-1.el5.noarch* depends on it. We download and push it to the channel with:

```
d19sw1# cd
d19sw1# wget http://dl.fedoraproject.org/pub/epel/5/x86_64/python-
hashlib-20081119-4.el5.x86_64.rpm
.. output skipped ..
d19sw1# rhnpush -c con-tools-centos-x86_64-server-5 -u example -p redhat --server=$(hostname)
 python-hashlib-20081119-4.el5.x86_64.rpm
```

1.6.3. Registration packages

Some of the packages of the **con-tools-centos-x86_64-server-5** channel need to be moved to the **centos-x86_64-server-5** base channel. The reason lies in the fact that the `comps.xml` for the **base** component contains the *rhn-check*, *rhn-setup* and *yum-rhn-plugin* packages. These are needed for system registration. Including their dependencies, we come up with the following list:

1. *rhn-check-1.9.10-1.el5.noarch.rpm*
2. *rhn-client-tools-1.9.10-1.el5.noarch.rpm*
3. *rhnlib-2.5.55-1.el5.noarch.rpm*
4. *rhnsd-5.0.9-1.el5.x86_64.rpm*
5. *rhn-setup-1.9.10-1.el5.noarch.rpm*
6. *rhn-setup-gnome-1.9.10-1.el5.noarch*
7. *yum-rhn-plugin-1.9.4-1.el5.noarch.rpm*

Note that *rhn-setup-gnome-1.9.10-1.el5.noarch* is added because it is in `comps.xml` too, under **x-base**.

Installing MSAT

Download the tarball from *http://msat.disruptivefoss.org/Downloads/tarball/msat_scripts.tgz*. On the Spacewalk server, do:

```
d19sw1# echo 'PATH=$PATH:/usr/local/bin/msat' > /etc/profile.d/msat.sh
d19sw1# mkdir -p /usr/local/bin/msat
d19sw1# cd /usr/local/bin/msat
d19sw1# wget msat.disruptivefoss.org/Downloads/tarball/msat_scripts.tgz
d19sw1# tar xzf msat_scripts.tgz
```

A configuration file, **`~/.sat.conf`** is needed to. See . Make sure to replace the *`login`* with *`example`* and *`password`* with *`redhat`*.

Before we do the move, we use our first **`msat`** command. We need to save the above set of RPM's in a directory to be able to calculate the repository metadata later! We must do it here:

```
d19sw1# cd
d19sw1# mkdir registration
d19sw1# msat_wr_sc_rpms.py -l con-tools-centos-x86_64-server-5 -e .
d19sw1# cd con-tools-centos-x86_64-server-5
d19sw1# mv rhn-check-1.9.10-1.el5.noarch.rpm rhn-setup-* rhn-client-
tools-1.9.10-1.el5.noarch.rpm rhnlib-2.5.55-1.el5.noarch.rpm rhnsd-5.0.9-1.el5.x86_64.rpm
 yum-rhn-plugin-1.9.4-1.el5.noarch.rpm ../registration/
```

Moving the RPM's means: delete them from the **con-tools-centos-x86_64-server-5** channel at the web interface and them place them in **centos-x86_64-server-5** channel.

Sign In → **Spacewalk Login: example** → **Password: redhat**+**Sign In** → **Channels** → **Manage Software Channels** → **con-tools-centos-x86_64-server-5** → **Packages** → **List / Remove Packages** → **Check: rhn-check-1.9.10-1.el5.noarch.rpm, rhn-client-tools-1.9.10-1.el5.noarch.rpm, rhnlib-2.5.55-1.el5.noarch.rpm, rhnsd-5.0.9-1.el5.x86_64.rpm, rhn-setup-1.9.10-1.el5.noarch.rpm, rhn-setup-gnome-1.9.10-1.el5.noarch, yum-rhn-plugin-1.9.4-1.el5.noarch.rpm,** → **Confirm Removal** → **Remove**

Channels → **Manage Software Channels** → **centos-x86_64-server-5** → **Packages** → **Add** → **Packages in no channels.** → **Select All (should be 7 just removed)** → **Confirm Addition** → **Add Packages(s)**

1.6.4. Kickstartable tree

After filling the software channels, we need to create a kickstartable tree. A **kickstartable tree** is the software from which we can start the OS installation. It is created on the Spacewalk server and on the web interface. We start with the Spacewalk server part.

The part on the Spacewalk server consists of the RPM's, images and the repository metadata. A standard yum repository is a directory in which one finds the RPM's and the metadata of the repository. The metadata is calculated with the **`createrepo`** command.

Since the Satellite server stores the RPM's in its own way, it is not possible to calculate the repodata with **`createrepo`** in the normal way. We tackle this problem by using the CentOS DVD and the **`/root/registration`** directory.

```
d19sw1# mkdir -p /var/satellite/rhn/kickstart/ks-centos-x86_64-server-5u8
d19sw1# cd /var/satellite/rhn/kickstart/ks-centos-x86_64-server-5u8
d19sw1# mount | grep CentOS-5.8-x86_64-bin-DVD-1of2.iso
.. not mounted ..
d19sw1# mount -o loop /root/CentOS-5.8-x86_64-bin-DVD-1of2.iso /var/www/html/pub/iso/1
d19sw1# cp -a /var/www/html/pub/iso/1/images .
d19sw1# cp -a /var/www/html/pub/iso/1/isolinux .
d19sw1# cp -a /var/www/html/pub/iso/1/repodata .
d19sw1# cp repodata/comps.xml .
d19sw1# createrepo -o /var/satellite/rhn/kickstart/ks-centos-x86_64-server-5u8 -p --
simple-md-filenames --no-database -s sha -g /var/satellite/rhn/kickstart/ks-centos-x86_64-
server-5u8/comps.xml --split /var/www/html/pub/iso/1/CentOS --split /var/www/html/pub/iso/2/
CentOS /root/registration
.. takes about 2 minutes ..
```

The next thing we need to do, is to create the kickstartable tree on the Web interface by defining a kickstart distribution.

Sign In → **Spacewalk Login: example** → **Password: redhat**+**Sign In** → **Systems** → **Kickstart** → **Distributions** → **create new distribution** → **Distribution Label: ks-centos-x86_64-server-5u8** → **Tree Path: /var/satellite/rhn/kickstart/ks-centos-x86_64-server-5u8** → **Base Channel: centos-x86_64-server-5** → **Installer Generation: Red Hat Enterprise Linux 5**+**Create Kickstart Distribution**

Make sure to set the **Variables** correctly. **Sign In** → **Spacewalk Login: example** → **Password: redhat**+**Sign In** → **Systems** → **Kickstart** → **Distributions** → **ks-centos-x86_64-server-5u8** → **Variables** → **media_path=/ks/dist/org/1/ks-centos-x86_64-server-5u8** → **org=1**+**Update Variables**

Next, we install the public GPG keys, needed for RPM verification. We download them to the Spacewalk server and also copy them to our local PC on which we run the web browser to the Spacewalk web interface.

```
d19sw1# cd /etc/pki/rpm-gpg
d19sw1# wget http://vault.centos.org/5.8/os/x86_64/RPM-GPG-KEY-CentOS-5
d19sw1# wget http://spacewalk.redhat.com/yum/RPM-GPG-KEY-spacewalk-2012
d19sw1# wget http://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-5
```

Next we dump them to stdout so that we can copy them to our local Linux PC:

```
d19sw1# cd /etc/pki/rpm-gpg
d19sw1# cat RPM-GPG-KEY-CentOS-5
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.6 (GNU/Linux)

mQGiBEWfB6MRBACrnYW6yKMT+MwJlCIhoyTxGf3mAxmnAiDEy6HcYN8rivssVTJk
CFtQBlBOpLV/OW2YtKrCO2xHn46eNfnMri8FGT8g+9JF3MUVi7kiV1He4iJynHXB
+F2ZqIvHf3IaUj1ys+p8TK64FDFxDQDrGQfIsD/+pkSGx53/877IrvdwjwCguQcr
Ioip5TH0Fj0OLUY4asYVZH8EAIqFHEqsY+9ziP+2R3/FyxSllKkjwcMLrBug+cYO
LYDD6eQXE9Mq8XKGFDj9ZB/0+JzK/XQeStheeFG75q3noq5oCPVFO4czuKErIRAB
qKbDBhaTj3JhOgM12XsUYn+rI6NeMV2ZogoQCC2tWmDETfRpYp2moo53NuFWHbAy
XjETA/sHEeQT9huHzdi/lebNBj0L8nBGfLN1nSRP1GtvagBvkR4RZ6DTQyl0UzOJ
RA3ywWlrL9IV9mrpb1Fmn60l2jTMMCc7J6LacmPK906N+FcN/Docj1M4s/4CNanQ
```

```
NhzcFhAFtQL56SNyLTCk1XzhssGZ/jwGnNbU/aaj4wOj0Uef5LRGQ2VudE9TLTUg
S2V5IChDZW50T1MgNSBPZmZpY2lhbCBTaWduaW5nIEtleSkgPGNlbnRvcy01LWtl
eUBjZW50b3Mub3JnPohkBBMRAgAkBQJFnwekAhsDBQkSzAMABgsJCAcDAgMVAgMD
FgIBAh4BAheAAAoJEKikR9zoViiXKlEAmwSoZDvZo+WChcg3s/SpNoWCKhMAAJwI
E2aXpZVrpsQnInUQWwkdrTiL5YhMBBMRAgAMBQJFnwiSBYMSzAIRAAoJEDjCFhY5
bKCk0hAAn134bIx3wSbq58E6P6U5RT7Z2Zx4AJ9VxnVkoGHkVIgSdsxHUgRjo27N
F7kBDQRFnwezEAQA/HnJ5yiozwgtf6jt+kii8iua+WnjqBKomPHOQ8moxbWdv5Ks
4e1DPhzRqxhshjmub4SuJ93sgMSAF2ayC9t51mSJV33KfzPF2gIahcMqfABe/2hJ
aMzcQZHrGJCEX6ek8l8SFKou7vICzyajRSIK8gxWKBuQknP/9LKsoczV+xsAAwUD
/idXPkk4vRRHsCwc6I23fdI0ur52bzEqHiAIswNfO521YgLk2W1xyCLc2aYjc8Ni
nrMX1tCnEx0/gK7ICyJoWH1Vc7//79sWFtX2EaTO+Q07xjFX4E66WxJlCo9lOjos
Vk5qc7R+xzLDoLGFtbzaTRQFzf6yr7QTu+BebWLoPwNTiE8EGBECAA8FAkWfB7MC
GwwFCRLMAwAACgkQqKRH3OhWKJfvvACfbsF1WK193zM7vSc4uq51XsceLwgAoI0/
9GxdNhGQEAweSlQfhPa3yYXH
=o/Mx
-----END PGP PUBLIC KEY BLOCK-----

localpc$ cat > /tmp/RPM-GPG-KEY-CentOS-5
.. the base64 encoded key from above ..

d19sw1# cat RPM-GPG-KEY-spacewalk-2012
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.18 (GNU/Linux)

mQGiBE9V9U0RBADjRxY1+Ng5gzaAj2LYHNwXHzyH65p+jL80+2vkf6WCNvewa+zK
SY8JH3syZMhjGi/vW3TcDy5KVqiXS2rpMJS6zCBrOJbtcFdV3VvbsPd7hK9COlph
NUx5RSIIwZRg1wyEjgeuYOSLuIhqNsI+fjXk+uzletSLtIYUF3TUq5jCvwCg4XQ/
/RPOFH6KiHfIx8QUZmvT0IkD/ip7pOn5uSPNiIbj3X5RYbz8PB/z5OuoenfzYn4R
WDBXZBlWMaPJCupAYwdP9IsiXo8WvbyXPHgG91P/MStrCgOffACNhuMus1FwqlCQ
VFuKENB6f2g4DY9Mow6bKYgsSEy0qnEF5I2M+BqjApO0oznxzcyJ4coXQdA9/oCP
vYU5A/9FTgukU1p/CbNJIhT4iH+vE6cKFGAtzhwwQxMdnii8ctcKiCYgiimuU7OG
rcPKR1xsRX/sc6XZYJeLmc1Lt7Btkdl36RY5BgesVJkUNyKh8+7MllW5PvVFvtGv
vvvrRVK+mJYxdBU0/Sqc94ZgBW6vdvRwHgXhpKor6NsCfJUWvLQmU3BhY2V3YWxr
IDxzcGFjZXdhbGstZGV2ZWxAcmVkaGF0LmNvbT6IaAQTEQIAKAUCT1X1TQIbAwUJ
BKKGAAYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQDmRvaIY6hT16sQCfVuJv
KnSWzWXo0ek16NrqUSgdTYIAn3Q6Jc3GAzkx/JgmA6T6pLxp2f1uiEYEEBECAAYF
Ak9V/egACgkQ7WNTebOJITK97gCfdeNye+OKidB2huEfzxP37ZkdfSIAniAA4NMY
QsmjPPkUJSVguO7GbdXSuQINBE9V9U0QCACBbWXaPP5RqAIGnr9LeFbSYpWs2SnL
KPv0qylyOGMc3GwmpJjeuz1d4y5fHIQxsQl7+RLIxOARcYH1QL6gHN450JcpYzau
/Lj1ArsAP/QK/J8YSmjG0E0rU8sYRWIu7jzISgfMAs3c3wNvOMUCiXtXFV6crm9q
3kbrZTedWLkShSLDAp5394hmBbRicWnc9h4LwBKp+BBtsvJFfDydc8IJUGZLZNZT
RayYWC5VMDe1xmOZXhNpDxfNrjhUUI6E6qWwdinmzifhP6z6YNtRON1yWSLBwtcr
LghyEJlilijIbCDgZ8Nvtqy/c2ypcLxMrXy70x05fjcQK2KIuLI58GGDAAMFB/9n
m5YLj4S4XxF8/xP7givuPq37C7NYLOuzFKKx6xxqngV8TNGDVU2zv/lieh2ussXE
XF/S9Du2abpXYgxsxwFJNMMKXUam03ExFSUvUAco6nmxySzvtgudCZ/9HqLhbNVu
RpyUNA/g9vkYK6KVdxAeHUdu/EwfSL/pCEc11Dw1jYqbihktlTQPcizsSPG4Y7jh
OUrKgxJop9DovUGzuUSl/7zYr4W7TiGV3my0f7U3qkpVT0bQoMnw8TxCZj4x3umo
F4klakeGX0FK4vwV5hDq32Q6vWN804F34Vi3WsaUT9G1E3H7/s/OQgWLOm16mgjR
pN1E79kYwtPMCAz0pYMciE8EGBECAA8FAk9V9U0CGwwFCQSihgAACgkQDmRvaIY6
hT3D2ACfYsiYS39rFdR1aBHviE0hVm9MWxUAnR3Ua92zW/0UoJ7TgxR0Dxzta37z
=yqYQ
-----END PGP PUBLIC KEY BLOCK-----

localpc$ cat > /tmp/RPM-GPG-KEY-spacewalk-2012
.. the base64 encoded key from above ..

d19sw1# cat RPM-GPG-KEY-EPEL-5
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.6 (GNU/Linux)

mQGiBEXopTIRBACZDBMOoFOakAjaxw1LXjeSvh/kmE35fU1rXfM7T0AV31NATCLF
l5CQiNDA4oWreDThg2Bf6+LIVTsGQb1V+XXuLak4Em5yTYwMTVB//4/nMxQEbpl/
QB2XwlJ7EQ0vW+kiPDz/7pHJz1p1jADzd9sQQicMtzysS4qT2i5A23j0VwCg1PB/
lpYqo0ZhWTrevxKMa1n34FcD/REavj0hSLQFTaKNLHRotRTF8V0BajjSaTkUT4uk
/RTaZ8Kr1mTosVtosqmdIAA2XHxi8ZLiVPPSezJjfElsSqOAxEKPL0djfpp2wrTm
l/1iVnX+PZH5DRKCbjdCMLDJhYap7YUhcPsMGSeUKrwmBCBJUPc6DhjFvyhA9IMl
1T0+A/9SKTv94ToP/JYoCTHTgnG5MoVNafisfe0wojP2mWU4gRk8X4dNGKMj6lic
```

```
vM6gne3hESyjcqZSmr7yELPPGhI9MNauJ6Ob8cTR2T12Fmv9w03DD3MnBstR6vhP
QcqZKhc5SJYYY7oVfxlSOfF4xfwcHQKoD5TOKwIAQ6T8jyFpKbQkRmVkb3JhIEVQ
RUwgPGVwZWxAZmVkb3JhcHJvamVjdC5vcmc+iGQEExECACQFAkXopTICGwMFCRLM
AwAGCwkIBwMCAxUCAwMWAgECHgECF4AACgkQEZzANiF1IfabmQCgzvE60MnHSOBa
ZXXF7uU2Vzu8EOkAoKg9h+j0NuNom6WUYZyJQt4zc5seuQINBEXopTYQCADapnR/
blrJ8FhlgNPl0X9S3JE/kygPbNXIqne4XBVYisVp0uzNCRUxNZq30MpY027JCs2J
nL2fMpwvx33f0phU029vrIZKA3CmnnwVsjcWfMJOVPBmVN7m5bGU68F+PdRIcDsl
PMOWRLkTBZOGolLgIbM4719fqA8etewILrX6uPvRDwywV7/sPCFpRcfNNBUY+Zx3
5bf4fnkaCKxgXgQS3AT+hGYhlzIqQVTkGNveHTnt4SSzgAqR9sSwQwqvEfVtYNeS
w5rDguLG41HQm1Hojv59HNYjH6F/S1rClZi21bLgZbKpCFX76qPt8CTw+iQLBPPd
yoOGHfzyp7nsfhUrAAMFB/9/H9Gpk822ZpBexQW4y3LGFo9ZSnmu+ueOZPU3SqDA
DW1ovZdYzGuJTGGM9oMl6bL8eZrcUBBOFaWge5wZczIE3hx2exEOkDdvq+MUDVD1
axmN45q/7h1NYRp5GQL2ZsoV4g9U2gMdzHOFtZCER6PP9ErVlfJpgBUCdSL93V4H
Sgpkk7znmTOklbCM6l/G/A6q4sCRqfzHwVSTiruyTBiU9lfROsAl8fjIq2OzWJ2T
P9sadBe1llUYaow7txYSUxssW+89avct35gIyrBbof5M+CBXyAOUaSWmpM2eub24
0qbqiSr/Y6Om0t6vSzR8gRk7g+1H6IE0Tt1IJCvCAMimiE8EGBECAA8FAkXopTYC
GwwFCRLMAwAACgkQEZzANiF1IfZQYgCgiZHCv4xb+sTHCn/otc1Ovvi/OgMAnRXY
bbsLFWOfmzAnNIGvFRWy+YHi
=MMNL
-----END PGP PUBLIC KEY BLOCK-----
```

```
localpc$ cat > /tmp/RPM-GPG-KEY-EPEL-5
.. the base64 encoded key from above ..
```

Next, we need to upload the public GPG keys to the Spacewalk server, via the web interface.

Sign In → **Spacewalk Login: example** → **Password: redhat**+**Sign In** → **Systems** → **Kickstart** → **GPG and SSL Keys** → **create new stored key/cert**

Table 1.11. Create CentOS 5 GPG/SSL Key

parameter	value
Description:	RPM-GPG-KEY-CentOS-5
Type:	GPG
Select file to upload:	**`/tmp/RPM-GPG-KEY-CentOS-5`**

Table 1.12. Create Spacewalk 2012 GPG/SSL Key

parameter	value
Description:	RPM-GPG-KEY-spacewalk-2012
Type:	GPG
Select file to upload:	**`/tmp/RPM-GPG-KEY-spacewalk-2012`**

Table 1.13. Create EPEL 5 GPG/SSL Key

parameter	value
Description:	RPM-GPG-KEY-EPEL-5
Type:	GPG
Select file to upload:	**`/tmp/RPM-GPG-KEY-EPEL-5`**

1.7. Test Spacewalk

Table 1.14. Time needed

action	time
Config channel	5 minutes
Activation key	5 minutes

action	time
Kickstart profile	20 minutes
Cobbler	5 minutes
Test installation	15 minutes

To test Spacewalk, we need the following items:

- A configuration file in a configuration channel.
- An activation key with at least one RPM and a link to the configuration channel.
- A kickstart profile with a link to the activation key and 1 post install script.
- A Cobbler system and the Cobbler boot ISO.
- The environment on the host to create a KVM virtual machine.

1.7.1. Config Channel

Create the config channel. **Configuration** → **Configuration Channels** → **create new config channel**

Table 1.15. Config Channel

parameter	value
Name:	test01
Label:	test01
Description:	test01

Add a configuration file with: **Add Files** → **Create File**

Now, add the following parameters:

Table 1.16. `/etc/motd`

parameter	value
Filename/Path:	/etc/motd
File Contents:	`Welcome to test01.`

Click **Create Configuration File**.

1.7.2. Activation Key

The **test01** config channel needs to be refered to by an activation key. We create it here.

Sign In → **Spacewalk Login: example** → **Password: redhat**+**Sign In** → **Systems** → **Activation Keys** → **create new key**

Table 1.17. Activation Key

parameter	value
Description:	test01
Key:	test01
Usage:	<empty>

parameter	value
Base Channels:	centos-x86_64-server-5
Add-On Entitlements:	Provisioning
Universal Default:	<unchecked>

Click **Create Activation Key**

Make sure to also check **Configuration File Deployment** since we want to deploy configuration file too.

Details → **Configuration File Deployment** → **check**+**Update Key**

Add an RPM package with: **Packages** → **screen**+**Update Key**

Link the configuration channel with **Configuration** → **Subscribe to Channels** → **test01**+**Continue**

1.7.3. Kickstart Profile

To further determine how the system should be provisioned, we define the **test01** kickstart profile. In it, we refer to the activation key.

Sign In → **Spacewalk Login: example** → **Password: redhat**+**Sign In** → **Systems** → **Kickstart** → **Profiles** → **create new kickstart profile**

Table 1.18. Kickstart Profile

parameter	value
Label:	test01
Base Channel:	cnetos-x86_64-server-5
Kickstartable Tree:	ks-centos-x86_64-server-5u8
Base Channels:	centos-x86_64-server-5
Virtualization Type:	None

Next+**Next**

Table 1.19. Root password

parameter	value
New Root Password:	redhat
Verify New Root Password:	redhat

Finish

Table 1.20. Kickstart: test01

parameter	value
Kickstart Label:	test01
Operating System:	centos-x86_64_server-5
Virtualization Type:	None
Active:	<checked>
Log custom post scripts:	<checked>
Log custom pre scripts:	<checked>
Preserve ks.cfg:	<checked>

parameter	value
Organization Default Profile:	<unchecked>
Kernel Options:	<empty>
Post Kernel Options:	<empty>
Comments:	<empty>

Update Kickstart

Advanced Options

Note, some values need to be unchecked and cleared! Really, only the values presented here must be present.

Table 1.21. Kickstart: test01, System Details

parameter	value
auth:	--enablemd5 --enableshadow
bootloader:	--location mbr
clearpart:	--all
firewall:	--enabled
install:	<checked>
key:	--skip
keyboard:	us
lang:	en_US
reboot:	<checked>
rootpw:	$1$RsnjsSwK$KbtWynXvaUGZ0ZZh0qXQu/
selinux:	--enforcing
skipx:	<checked>
text:	<checked>
timezone:	--utc Europe/Amsterdam
url:	--url /var/satellite/rhn/kickstart/ks-centos-x86_64-server-5u8
zerombr:	<checked>

Update Kickstart

System Details+Details

We only show the checked options and their value!

Table 1.22. Kickstart: test01, Advanced Options

parameter	value
SeLinux:	Enforcing
Enable Spacewalk Configuration Management:	<checked>
Enable Spacewalk Remote Commands:	<checked>
Spacewalk Profile:	Re-connect to the existing system profile. Do not create a new system profile.
New Root password:	<empty>

parameter	value
Verify New Root password:	<empty>

Update System Details

System Details+Partitioning

Partition Details:

```
part /boot --fstype=ext3 --size=200
part pv.01 --size=1 --grow
part swap --size=512
volgroup myvg pv.01
logvol / --vgname=myvg --name=rootvol --size=1 --grow
```

Update Partitions

System Details+GPG & SSL+Select All+Update keys

Activation Keys+Select All+Update Activation Keys

Scripts+add new kickstart script

Table 1.23. Kickstart: Script

parameter	value
Script Name	1
Script Contents	`/bin/rm /etc/yum.repos.d/CentOS-*.repo`
Script Execution Time	Post Script
Template	<checked>

Update Kickstart

1.7.4. Cobbler

Since we want to start our bare metal provisioning with a boot ISO, we need to create a system in Cobbler. This task can be accomplished by using a script. We present it here:

```
d19sw1# cat /root/cobbler-dmm01.sh
#!/bin/bash
#
# SCRIPT
#   cobbler-dmm01.sh
# DESCRIPTION
#   This script should be run on the Spacewalk server:
#   # ./cobbler-dmm01.sh
#
#   IP details
#
#   Host       Prod            Cluster        ILO
#   dmm01      192.168.5.100   -              -
#
#   Subnets:
#   192.168.5.0/24  gw: 192.168.5.1
#
#   DNS servers:
#   * 192.168.5.11 (d19sw1.dmsat1.org)
#
```

```
# ARGUMENTS
#   None.
# RETURN
#   Value from cobbler command. See cobbler man page.
# DEPENDENCIES
#   The profile should not yet exist in cobbler. If it does,
#   remove it with:
#   sudo cobbler system remove \
#     --name=SpacewalkDefaultOrganization_kvm_dmm01
#   Adding an existing profile results in a clear warning
#   from cobbler. No harm is done.
# FAILURE
# AUTHORS
#   Date strings made with 'date +"\%Y-\%m-\%d \%H:\%M"'.
#   Allard Berends (AB), 2013-05-02 19:18
# HISTORY
# LICENSE
#   Copyright (C) 2013 Allard Berends
#
#   cobbler-dmm01.sh is free software; you can
#   redistribute it and/or modify it under the terms of the
#   GNU General Public License as published by the Free
#   Software Foundation; either version 3 of the License, or
#   (at your option) any later version.
#
#   cobbler-dmm01.sh is distributed in the hope
#   that it will be useful, but WITHOUT ANY WARRANTY;
#   without even the implied warranty of MERCHANTABILITY or
#   FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
#   Public License for more details.
#
#   You should have received a copy of the GNU General
#   Public License along with this program; if not, write to
#   the Free Software Foundation, Inc., 59 Temple Place -
#   Suite 330, Boston, MA 02111-1307, USA.
# DESIGN
#

COBBLER="sudo cobbler"
# AB: non base Spacewalk organizations have orgnum != 1
#ORGNUM=$(msat_ls_org.py)
ORGNUM=1

########## PARAMETERS TO EDIT ##########
NAME="dmm01"
OWNERS="example"
MACH="kvm"
PROFILE="test01"
ORG="SpacewalkDefaultOrganization"
COMMENT="empty"
GATEWAY=192.168.5.1
NAMESERVERS="192.168.5.11"
NAMESERVERS_SEARCH="dmsat1.org"
HOSTNAME=${NAME}.${NAMESERVERS_SEARCH}
PROD_IP=192.168.5.100
PROD_SUBNET=255.255.255.0
PROD_DNS_NAME=${HOSTNAME}
MAC_ETH0=52:54:00:a8:05:64

########## START SCRIPT ##########
$COBBLER system add \
  --name=${ORG}_${MACH}_${NAME} \
  --owners=${OWNERS} \
  --profile=${PROFILE}:${ORGNUM}:${ORG} \
  --kopts="ksdevice=${MAC_ETH0} ip=${PROD_IP} netmask=${PROD_SUBNET}" \
  --netboot-enabled=0 \
  --comment=${COMMENT} \
  --power-type=ipmitool \
  --hostname=${HOSTNAME} \
  --gateway=${GATEWAY} \
  --name-servers="$NAMESERVERS" \
  --name-servers-search=$NAMESERVERS_SEARCH \
  --redhat-management-key='<<inherit>>' \
  --redhat-management-server='<<inherit>>'
  #--uid=UID
  #--image=IMAGE
  #--kopts-post=KOPTS_POST
  #--ksmeta=KSMETA
  #--kickstart=KICKSTART
  #--depth=DEPTH
  #--server=SERVER
  #--virt-path=VIRT_PATH
  #--virt-type=VIRT_TYPE
  #--virt-cpus=VIRT_CPUS
  #--virt-file-size=VIRT_FILE_SIZE
  #--virt-ram=VIRT_RAM
  #--virt-auto-boot=VIRT_AUTO_BOOT
  #--ctime=CTIME
  #--mtime=MTIME
  #--power-address=POWER_ADDRESS
  #--power-user=POWER_USER
  #--power-pass=POWER_PASS
  #--power-id=POWER_ID
  #--ipv6-default-device=IPV6_DEFAULT_DEVICE
  #--ipv6-autoconfiguration=IPV6_AUTOCONFIGURATION
  #--mgmt-classes=MGMT_CLASSES
  #--template-files=TEMPLATE_FILES
  #--template-remote-kickstarts=TEMPLATE_REMOTE_KICKSTARTS
  #--clobber
  #--template-files=TEMPLATE_FILES
  #--in-place

/usr/bin/cobbler system edit \
  --name=${ORG}_${MACH}_${NAME} \
  --mac-address=${MAC_ETH0} \
  --ip-address=${PROD_IP} \
  --static=1 \
  --subnet=${PROD_SUBNET} \
  --dns-name=${PROD_DNS_NAME} \
  --interface=eth0
  #--mtu=MTU
  #--dhcp-tag=DHCP_TAG
  #--static-routes=STATIC_ROUTES
  #--virt-bridge=VIRT_BRIDGE
  #--ipv6-address=IPV6_ADDRESS
  #--ipv6-secondaries=IPV6_SECONDARIES
  #--ipv6-mtu=IPV6_MTU
  #--ipv6-static-routes=IPV6_STATIC_ROUTES
  #--ipv6-default-gateway=IPV6_DEFAULT_GATEWAY
```

We run the above script and verify that the system is known in Cobbler:

```
d19sw1# chmod u+x cobbler-dmm01.sh
d19sw1# ./cobbler-dmm01.sh
d19sw1# sudo cobbler system list | grep dmm01
   SpacewalkDefaultOrganization_kvm_dmm01
```

We create the Cobbler boot ISO with:

```
d19sw1# sudo cobbler get-loaders
```

```
task started: 2013-05-02_155133_get_loaders
task started (id=Download Bootloader Content, time=Thu May  2 15:51:33 2013)
downloading http://dgoodwin.fedorapeople.org/loaders/README to /var/lib/cobbler/loaders/
README
downloading http://dgoodwin.fedorapeople.org/loaders/COPYING.elilo to /var/lib/cobbler/
loaders/COPYING.elilo
downloading http://dgoodwin.fedorapeople.org/loaders/COPYING.yaboot to /var/lib/cobbler/
loaders/COPYING.yaboot
downloading http://dgoodwin.fedorapeople.org/loaders/COPYING.syslinux to /var/lib/cobbler/
loaders/COPYING.syslinux
downloading http://dgoodwin.fedorapeople.org/loaders/elilo-3.8-ia64.efi to /var/lib/cobbler/
loaders/elilo-ia64.efi
downloading http://dgoodwin.fedorapeople.org/loaders/yaboot-1.3.14-12 to /var/lib/cobbler/
loaders/yaboot
downloading http://dgoodwin.fedorapeople.org/loaders/pxelinux.0-3.61 to /var/lib/cobbler/
loaders/pxelinux.0
downloading http://dgoodwin.fedorapeople.org/loaders/menu.c32-3.61 to /var/lib/cobbler/
loaders/menu.c32
downloading http://dgoodwin.fedorapeople.org/loaders/grub-0.97-x86.efi to /var/lib/cobbler/
loaders/grub-x86.efi
downloading http://dgoodwin.fedorapeople.org/loaders/grub-0.97-x86_64.efi to /var/lib/
cobbler/loaders/grub-x86_64.efi
*** TASK COMPLETE ***
d19sw1# yum install syslinux
.. might already be installed ..
d19sw1# cd /var/lib/cobbler/loaders
d19sw1# cp /usr/share/syslinux/menu.c32 .
d19sw1# cp /usr/share/syslinux/vesamenu.c32 .
d19sw1# mkdir -p /var/www/cobbler/pub/example
d19sw1# sudo cobbler buildiso --iso=/var/www/cobbler/pub/example/centos5u8-test01-kvm.iso --
profiles="" --systems="SpacewalkDefaultOrganization_kvm_dmm01" --tempdir=/tmp
task started: 2013-05-02_155717_buildiso
task started (id=Build Iso, time=Thu May  2 15:57:17 2013)
using/creating buildisodir: /tmp/buildiso
building tree for isolinux
copying miscellaneous files
copying kernels and initrds for profiles
copying kernels and initrds for systems
generating a isolinux.cfg
generating profile list
generating system list
 - ksdevice 52:54:00:a8:05:64 set for system SpacewalkDefaultOrganization_kvm_dmm01
append line length is greater than 254 chars: (276 chars)
done writing config
running: mkisofs -o /var/www/cobbler/pub/example/centos5u8-test01-kvm.iso -r -b isolinux/
isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -V Cobbler
\ Install -R -J -T /tmp/buildiso
received on stdout:
received on stderr: I: -input-charset not specified, using utf-8 (detected in locale
 settings)
Size of boot image is 4 sectors -> No emulation
 68.62% done, estimate finish Thu May  2 15:57:18 2013
Total translation table size: 4029
Total rockridge attributes bytes: 1320
Total directory bytes: 4096
Path table size(bytes): 40
Max brk space used 1c000
7301 extents written (14 MB)

ISO build complete
You may wish to delete: /tmp/buildiso
The output file is: /var/www/cobbler/pub/example/centos5u8-test01-kvm.iso
*** TASK COMPLETE ***
d19sw1# ls -lh /var/www/cobbler/pub/example/centos5u8-test01-kvm.iso
-rw-r--r--. 1 root root 15M May  2 15:57 /var/www/cobbler/pub/example/centos5u8-test01-
kvm.iso
```

```
d19sw1# scp /var/www/cobbler/pub/example/centos5u8-test01-kvm.iso root@192.168.5.1:/tmp
```

1.7.5. KVM installation

On the host machine we need to create storage space for our test machine:

```
server# pvs
  PV         VG        Fmt  Attr PSize   PFree
  /dev/sda3  vg_server lvm2 a--  931.00g      0
  /dev/sdb   b         lvm2 a--  931.51g 638.54g
  /dev/sdc   c         lvm2 a--  931.51g 931.51g
  /dev/sdd   d         lvm2 a--  931.51g 931.51g
  /dev/sde   e         lvm2 a--  931.51g 931.51g
  /dev/sdf   f         lvm2 a--  931.51g 931.51g

server# pvs -o +vg_extent_size /dev/sde
  PV         VG   Fmt  Attr PSize   PFree   Ext
  /dev/sde   e    lvm2 a--  931.51g 931.51g 4.00m

server# lvcreate -n mm -l $((25 * 10 * 5)) e
  Logical volume "mm" created
```

Explanation of the parameters in the **virt-install** command:

- dmm01, d (development), mm (model machine), 01 (first).
- mac=52:54:00:a8:05:64, 192.168.5.100 -> 00:a8:05:64, the 52:54:00 is obligatory by libvirt definition.

To make it easy to verify the **virt-install** options in the *virt-install man page*, we specify them in the same order as they appear in the man page.

The installation is started with:

```
server# virt-install \
  --connect qemu:///system \
  --name=dmm01 \
  --ram=512 \
  --arch=x86_64 \
  --description="Provisioning test machine" \
  --cdrom /tmp/centos5u8-test01-kvm.iso \
  --os-type=linux \
  --os-variant=rhel5.4 \
  --disk path=/dev/e/mm,device=disk,bus=scsi \
  --network=network=dmsat1,mac=52:54:00:a8:05:64 \
  --graphics vnc \
  --hvm \
  --autostart
.. output skipped, about 10 minutes ..
```

Select **SpacewalkDefaultOrganization_kvm_dmm01** and press **Enter**.

To check on the installed system if everything is working fine, one does:

```
dmm01# rhncfg-client list
Using server name d19sw1.dmsat1.org
DoFoS   Config Channel   File
F           test01     /etc/motd
dmm01# yum repolist
```

```
Loaded plugins: fastestmirror, rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
Loading mirror speeds from cached hostfile
repo id                              repo name                            status
centos-x86_64-server-5               centos-x86_64-server-5               3,602
con-tools-centos-x86_64-server-5     con-tools-centos-x86_64-server-5        15
repolist: 3,617
```

1.8. Switching on SELinux

Table 1.24. Time needed

action	time
Create custom policy	5 minutes

In the last section, we have installed, registered and configured a system via Spacewalk. Meaning that all SELinux denials should now be logged in `/var/log/audit/audit.log`. In the standard logging, `/var/log/messages`, one should see **sealert** messages. Running one message gives us the recipe to create a custom policy to make **cobbler** run under SELinux.

First we detect the last **sealert** message:

```

d19sw1# grep sealert /var/log/messages
.
.
May 10 15:34:57 d19sw1 setroubleshoot: SELinux is preventing /usr/bin/python2.7 from
 link access on the file vmlinuz. For complete SELinux messages. run sealert -l 6cc70fe5-
a8b9-4320-bf39-cac3b8c10656
.
.
May 10 19:48:07 d19sw1 setroubleshoot: SELinux is preventing /usr/sbin/httpd from search
 access on the directory /var/log/cobbler. For complete SELinux messages. run sealert -l
 4fd01220-fd4c-4d89-9862-fe9d8e4397c0
.
.
```

As the messages suggest, we run `sealert`:

```

d19sw1# sealert -l 6cc70fe5-a8b9-4320-bf39-cac3b8c10656
.
.
# grep cobblerd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp
.
.
d19sw1# grep cobblerd /var/log/audit/audit.log | audit2allow -M mycbpol
******************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i mycbpol.pp

d19sw1# semodule -i mycbpol.pp
d19sw1# sealert -l 4fd01220-fd4c-4d89-9862-fe9d8e4397c0
.
.
# grep /usr/sbin/httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp
.
.
d19sw1# grep /usr/sbin/httpd /var/log/audit/audit.log | audit2allow -M myhdpol
******************** IMPORTANT ***********************
```

```
To make this policy package active, execute:

semodule -i myhdpol.pp

d19sw1# semodule -i myhdpol.pp
d19sw1# vim /etc/selinux/config
d19sw1# grep ^SELINUX= /etc/selinux/config
SELINUX=enforcing
d19sw1# setenforce 1
```

To verify if SELinux is now behaving correctly, we have to recreate the Spacewalk content with respect to the **test01** profile. Furthermore, we need to unregister the system and do the installation again.

We demonstrate here how to unregister the system from the Spacewalk server. In the section *Section 1.9, “MSAT Demo”*, we show how to easily fill the Spacewalk with the **test01** profile and attached items.

SELinux web interface denials

SELinux is also preventing the web interface to work correctly. It manifests itself as the error: *There are errors in your kickstart template. Please check the 'Kickstart File' tab to determine the problem with the template.* One can see this when clicking on a kickstart profile. To solve this issue, also run the **`sealert`** command, as described earlier.

1.9. MSAT Demo

Table 1.25. Time needed

action	time
Save Spacewalk content	1 minute
Remove Spacewalk content	1 minute
Recreate Spacewalk content	1 minute
Install test system	15 minutes

Since we have a functional Spacewalk server, we can demonstrate how the MSAT scripts function. In later chapters detailed information is provided about the MSAT scripts. Here, we merely demonstrate how we can create the same **test01** kickstart profile in a scripted way.

First, we start by saving the current content of the Spacewalk by saving **test01**.

```

d19sw1# msat_ls_kp.py
test01

d19sw1# msat_wr_kp_hy.py -l test01
d19sw1# tree --charset=ASCII -F --noreport test01
test01
|-- ak-test01.sh*
|-- cc-test01.sh*
`-- kp-test01.sh*

d19sw1# cat test01/kp-test01.sh
#!/bin/bash
#
# SCRIPT
#   kp-test01.sh
# DESCRIPTION
#   This script creates the test01
#   kickstart profile.
# ARGUMENTS
#   None.
# RETURN
#   0: success.
# DEPENDENCIES
# FAILURE
#   If you put single quotes around value, but forget to
#   escape embedded single quotes, this script will fail.
#   Escaping works like this:
#   $ echo 'don'"'"'t'
#   don't
#   So ' -> '"'"'
#   Complicated huh?
# AUTHORS
#   Date strings made with 'date +"\%Y-\%m-\%d \%H:\%M"'.
#   Allard Berends (AB), 2013-05-10 17:17
# HISTORY
# LICENSE
#   Copyright (C) 2013 Allard Berends
#
#   kp-test01.sh is free software; you can
#   redistribute it and/or modify it under the terms of the
#   GNU General Public License as published by the Free
#   Software Foundation; either version 3 of the License, or
#   (at your option) any later version.
#
#   kp-test01.sh is distributed in the hope
#   that it will be useful, but WITHOUT ANY WARRANTY;
#   without even the implied warranty of MERCHANTABILITY or
#   FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
#   Public License for more details.
#
#   You should have received a copy of the GNU General
#   Public License along with this program; if not, write to
#   the Free Software Foundation, Inc., 59 Temple Place -
#   Suite 330, Boston, MA 02111-1307, USA.
# DESIGN
#


ORGNUM=$(msat_ls_org.py)

SATELLITE=$(msat_ls_sn.py)

msat_mk_kp.py \
  --kickstart-label "test01" \
  --kickstart-virt none \
  --kickstart-tree ks-centos-x86_64-server-5u8 \
  --kickstart-satellite $SATELLITE \
  --kickstart-root redhat \
  --kickstart-childchannels '' \
  --kickstart-configmgt true \
  --kickstart-remotecmds true \
  --kickstart-partitioning 'part /boot --fstype=ext3 --size=200,part pv.01 --size=1 --
grow,part swap --size=512,volgroup myvg pv.01,logvol / --vgname=myvg --name=rootvol --size=1
 --grow' \
  --kickstart-keys 'RPM-GPG-KEY-EPEL-5,RPM-GPG-KEY-spacewalk-2012,RPM-GPG-KEY-CentOS-5,RHN-
ORG-TRUSTED-SSL-CERT' \
  --kickstart-install true \
  --kickstart-text true \
  --kickstart-url '--url /var/satellite/rhn/kickstart/ks-centos-x86_64-server-5u8' \
  --kickstart-lang 'en_US' \
  --kickstart-keyboard 'us' \
  --kickstart-zerombr true \
  --kickstart-clearpart '--all' \
  --kickstart-bootloader '--location mbr' \
  --kickstart-timezone '--utc Europe/Amsterdam' \
  --kickstart-auth '--enablemd5 --enableshadow' \
  --kickstart-rootpw '$1$yfdvYZ2s$mXFhCqaiSjZ0mt9M3ZKFr0' \
  --kickstart-selinux '--enforcing' \
  --kickstart-reboot true \
  --kickstart-firewall '--enabled' \
  --kickstart-skipx true \
  --kickstart-key '--skip' \
  --kickstart-software '@ Base' \
  --kickstart-activationkey $ORGNUM-test01 \
  --kickstart-script '/bin/rm /etc/yum.repos.d/CentOS-*.repo'
```

As one can see, the kickstart profile and the items it refers to are saved in the **`test01`** directory. Examining the content of, for example, the **`test01/kp-test01.sh`** script, shows that the items are saved in regeneration scripts. This means, if the script is run, it fills the Spacewalk server with the item.

Order of running scripts

The kickstart profile depends on the activation key. The activation key, on its turn, depends on the config channel with config file. Running the kickstart regeneration script first, results in an error. It can not find the specified activation key.

Now, it is time to unregister the already installed system. We do this with: **Sign In** → **Spacewalk Login: example** → **Password: redhat**+**Sign In** → **Systems** → **dmm01.dmsat1.org** → **delete system** → **Delete Profile**

Here, we show how to delete the system items (system in Cobbler, kickstart profiles, activation keys, config channels) in the Spacewalk server.

```
d19sw1# sudo cobbler system list
   SpacewalkDefaultOrganization_kvm_dmm01
d19sw1# sudo cobbler system remove --name SpacewalkDefaultOrganization_kvm_dmm01
d19sw1# msat_rm_kp_all.sh; msat_rm_ak_all.sh; msat_rm_cc_all.sh
```

Run msat_rm_*_all.sh only on a test Spacewalk

Although very handy during a development phase of a system, the **`msat_rm_*_all.sh`** commands completely remove all items (kickstart profiles, activation keys, config channels) in the Spacewalk server. Do this on a production system to find out that all your system definitions are gone. Even if you can restore it with MSAT regeneration scripts, still all the production systems need to be reregistered on the Spacewalk server!

Now, we run the regeneration scripts in the right order to create the system items again. The right order is automatically executed by **`msat_mk_kp_hy.sh`**.

```
d19sw1# cd
d19sw1# msat_mk_kp_hy.sh -d /root/test01
/usr/local/bin/msat/msat_mk_kp_hy.sh: line 134: ./cs*: No such file or directory
d19sw1# ./cobbler-dmm01.sh
d19sw1# sudo cobbler buildiso --iso=/var/www/cobbler/pub/example/centos5u8-test01-kvm.iso --
profiles="" --systems="SpacewalkDefaultOrganization_kvm_dmm01" --tempdir=/tmp
.. output skipped ..
d19sw1# scp /var/www/cobbler/pub/example/centos5u8-test01-kvm.iso root@192.168.5.1:/tmp
root@192.168.5.1's password: redhat
centos5u8-test01-kvm.iso                      100%   14MB  14.3MB/s   00:00
```

The installation is started with:

```
server# virt-install \
  --connect qemu:///system \
  --name=dmm01 \
  --ram=512 \
  --arch=x86_64 \
  --description="Provisioning test machine" \
  --cdrom /tmp/centos5u8-test01-kvm.iso \
  --os-type=linux \
  --os-variant=rhel5.4 \
  --disk path=/dev/e/mm,device=disk,bus=scsi \
  --network=network=dmsat1,mac=52:54:00:a8:05:64 \
  --graphics vnc \
  --hvm \
  --autostart
.. output skipped, about 10 minutes ..
```

Select **SpacewalkDefaultOrganization_kvm_dmm01** and press **`Enter`**.

Appendix A. Revision History

Revision 0.1-0 **Thu May 9 2013** **Allard Berends**
msat.disruptivefoss@gmail.com

Brought book under publican

Index