

SUSECON 2017 – Intro to Alchemy

Use Machinery and KIWI to build gold images based on existing systems. This session is an introduction to Machinery and KIWI

NOTE: We will login to all VM's as root using the password "linux" without the quotation marks.

Text with a grey background represents command that you will need to type

Start Virtual Machines

On the machines in front you, find and start Virtual Machine Manager. If it's not on the tool bar at the bottom of the screen, it should be in the Yast modules folder.

Please start virtual machines: sc1 & sca

Both VM's are running SLES 12 SP2. The VM named sca will be your administrative node that you will login to and do all the lab work on. Node sc1 is simply a minimal SLES install (JeOS) that makes for a smaller system to work with.

Open sca so you can see the console screen & maximize it. We'll be working in that VM for the course.

Login to sca as geeko with a password of "linux"

Machinery Inspection

Open a terminal and become root

```
sudo su -
```

Use Machinery to inspect sc1

We will use machinery to inspect system sc1. Before doing that, we have to enable ssh authentication by copying an ssh key from sca to sc1.

```
ssh-copy-id root@sc1
```

Respond to the password prompt by entering `linux`

Show Systems Inspected

Let's run machinery list to see if anything has been inspected before.

```
machinery list
```

The list should be empty at this point as we have not inspected any systems yet.

Inspect a system

Run machinery and inspect VM sc1

```
machinery inspect sc1 --name sc1 --extract-files
```

When I built the sca VM, I installed machinery from the Advanced System Management Module repository. This is a fully supported module for SUSE Linux Enterprise Server that provides DevOps tools and Machinery.

Show Systems Inspected

You can list the machines you have collected data from

```
machinery list
```

Show collected data

Let's see what data was collected

```
machinery show sc1
```

Machinery Export

Export machinery data to html format

Let's export the data collected about sc1 to an html format so we can explore the data in a nicely formatted way using a web browser.

```
machinery export-html --html-dir=/srv/www/htdocs/sc1 sc1
```

Review sc1 data using a web browser

On system sca start Firefox and browse to: <http://localhost/sc1>

Explore the interface and look at the data that was collected. Enter "bash" in the filter and see what version of bash is on the system. Also look at the "services" tab and filter on ssh and you will see that the sshd.service was enabled as would be expected since we used ssh to access sc1.

Hover your mouse over the "scopes" letters across the top and see what they represent. Click on several of those to see what data is shown.

Mount the ramdisk

Since we're running in a VM we are going to use a ramdisk to make this faster for lab purposes.

```
mount -t tmpfs -o size=4g tmpfs /mnt/ramdisk
```

Let's verify that the ramdisk now exists

```
df | grep ram
```

You should see a line output that looks like this:

```
tmpfs    4194304    0    4194304    0%    /mnt/ramdisk
```

IF YOU SKIP CREATION OF THE RAMDISK THE REST OF THE LAB WILL NOT WORK AS WRITTEN

Export Machinery data to KIWI config

Now you are going to export the collected machinery data into a kiwi configuration.

```
machinery export-kiwi -k /mnt/ramdisk sc1
```

This will create a directory called "/tmp/kiwi/sc1-kiwi" which will contain the data needed to re-create the system using kiwi.

Explore KIWI config data

```
cd /mnt/ramdisk/sc1-kiwi
```

Looks through the files in the directory.

config.xml – This is the kiwi config file. It represents the system description in terms of filesystem, packages installed etc.

config.sh – This is a kiwi script that is run when the image is built.

./root – This is a file overlay directory. When the image is being built, kiwi will create a filesystem structure and install packages defined in the config.xml file. After the "system" is built, the files in /root are over-laid and will be in place on the resulting system over writing anything with the same name.

config-yast-autoyast.xml - (not there) This is an autoyast file that can be put in place that runs on first boot of the built image. It can be used to provide hostname/networking information along with other configuration information.

View the config.xml file

```
view config.xml
```

Scroll down about 11 lines and find the line that begins with "<type" and you should see that the filesystem definition is ext3. If you were to login to

KIWI Build

Build image using KIWI

Let's build an image! This will duplicate the system sc1 that we inspected with no changes, except it will have an ext3 file system as machinery doesn't yet output btrfs configurations.

```
kiwi -b /mnt/ramdisk/sc1-kiwi -d /mnt/ramdisk/sc1
```

The build process will take several minutes. Stand up and stretch & continue when it's finished.

```
cd /mnt/ramdisk/sc1
```

Find the name of the qcow2 file in the dir. It's probably "sc2.x86_64-0.0.1.qcow2"

```
cp sc1.x86_64-0.0.1.qcow2 /mnt/ramdisk/sc1.qcow2
```

Test VM

Run new Virtual Machine from new image

Start Virtual Machine Manager from Favorites. The gnome menu is in the lower left corner of the screen. Clicking on "Applications" will bring up the gnome menu and you can find Favorites there.

Rather than manually defining a VM I wrote a small script to define and start the virtual machine in KVM

```
start-sc1.sh
```

The VM will start automatically. Since we're using nested virtualization it will take a bit longer than normal to fully boot up. You will need to select the VM in Virtual Machine Manager and click the "Open" button to view the console screen.

Select "View" -> "Scale Display" -> "Always" to make sure the console fits on the screen

Login to the VM

Login to the VM on its console using "root/linux" as the credentials. Execute "mount" and see what filesystem is used for "/"

```
mount
```

You should see that the file system for / is ext3, even though the file system on the sc1 system we inspected is btrfs. We'll change that a little later.

Shutdown VM & Clean up previous build

MAKE SURE YOU ARE ON sca before executing the following!

```
virsh shutdown sc1
```

```
virsh undefine sc1
```

```
cleanup.sh
```

```
cd /mnt/ramdisk
```

```
rm -r sc1 sc1.qcow2 # PLEASE BE CAREFUL!
```

Modify KIWI Configuration

Modify KIWI configuration

Now we're going to modify the kiwi configuration to build the system with BTRFS. I have prepared a config.xml file and it is named: /usr/local/share/btrfs-config.xml

```
cp /usr/local/share/btrfs-config.xml /mnt/ramdisk/sc1-kiwi/config.xml
```

Take a look at the new kiwi config.xml file

```
cd /mnt/ramdisk/sc1-kiwi
```

```
view config.xml
```

Notice the section starting at line 13 which defines the btrfs file system.

Also notice the line "format=qcow2". This can be changed to "format=vmdk" to output a VMWare vmdk file rather than a qcow2 file. There are many different type of output formats available.

Rebuild KIWI Image

Re-Build image using KIWI

```
kiwi -b /mnt/ramdisk/sc1-kiwi -d /mnt/ramdisk/sc1
```

```
cp /mnt/ramdisk/sc1/sc1.x86_64-0.0.1.qcow2 /mnt/ramdisk/sc1.qcow2
```

Re-Test VM

Run new Virtual Machine from modified image

See above and repeat the steps the run the VM or run `start-sc1.sh &` # Yes I scripted it!

Go to Virtual Machine Manager and open the console window for sc1

When the VM is up and running, login and verify it's running a BTRFS file system.

Login and check / filesystem

Login as root and run:

```
mount
```

You should see that the file system for / is btrfs.

Machinery Compare

Machinery Compare

Let's compare two system descriptions. First we need to move the old sc1 machinery description to a new name.

```
machinery move sc1 sc1-old
```

Let's make a change to sc1

Login to sc1 as root. You can ssh from sca to sc1

```
ssh root@sc1
```

We need to make a change to sc1. Let's install apache2.

```
zypper in -y apache2
```

Logout of sc1 and make sure you are root on sca.

Now we have to re-inspect sc1

```
machinery inspect sc1 --name sc1 --extract-files
```

List the machinery data that you have now on sca

```
machinery list
```

You should see to listings, one named sc1-old and another named sc1. Now let's see what is different between the two inspections

```
machinery compare sc1 sc1-old
```

If you want to just see the how the installed packages compare, then there are scope options

```
machinery compare -s packages sc1 sc1-old
```

Lab Free Time

You are free to explore if you have time left over. Please let us know if you have any additional questions.

Links:

<http://www.suse.com/>

<http://machinery-project.org/>

<https://doc.opensuse.org/projects/kiwi/doc/>