HO126885 - Advanced Alchemy: Building Gold-Master Images with Kiwi

TL;DR (I'll do this later)

You can grab everything you need to reproduce the lab here: https://github.com/darthzen/conferences/tree/master/susecon/2017/HO126885/lab

Helpful Resources

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

Introduction

This lab focuses on taking a working Kiwi configuration file and building on it to include multiple image types, hooks, and embedded autoyast.

We will use the SLES12-SP3-Jeos-kvm.kiwi file as our base template. This file is taken from the JeOS templates available for download at the SUSE download site and has been modified by running it through a custom script available in my GitHub repository called `kiwi-strip-pattern-deps`, which will look at the installed patterns in a Kiwi file and strip out any packages specified which will be installed by default via the pattern. This makes the Kiwi file much smaller, and makes it more portable when modifying it to newer SUSE-based OS versions, as the pattern names tend to be more stable than the package names that make up its dependencies.

Getting Started

Virtual Machine Access

- Virtual machine IP address: `192.168.130.2`
- Root password: `linux`

Setup our build environment

- [Optional] Generate SSH key (unless you really enjoy repetitive typing)
- `ssh-keygen`
- `ssh-copy-id root@192.168.130.2`
- [Optional] Setup vim to detect kiwi files as XML for syntax coloring
- `ssh root@192.168.130.2 "mkdir -p .vim/ftdetect"`
- `scp kiwi.vim root@192.168.130.2:.vim/ftdetect/`

- `scp .vimrc root@192.168.130.2:`
- `ssh root@192.168.130.2 "zypper -n in vim-data"`
- [Optiona] Install updated kiwi-templates rpm to virtual machine
- Not necessary, but these templates can be useful as a reference when completing the labs.
- `scp kiwi-templates-SLES12-JeOS-12.3-12.16.x86_64.rpm root@192.168.130.2:`
- `ssh root@192.168.130.2 "rpm -Uvh kiwi-templates-SLES12-JeOS-12.3-12.16.x86_64.rpm"`
- Copy the Kiwi files over to the virtual machine
- `scp SLES12-SP3-Jeos-kvm.kiwi root@192.168.130.2:`
- `scp config.sh root@192.168.130.2:`
- SSH to the virtual machine and mount the DVD drive
- `ssh root@192.168.130.2`
- `mkdir -p /exports/SLES12-SP3-x86_64`
- `mount -o loop,ro /dev/dvd /exports/SLES12-SP3-x86_64`

Exercise 1 - Build the base image

Perform the following on the virtual machine:

- Go to build directory
- Can be anywhere on filesystem, but we'll use a directory I've already shared as an NFS mount point (so we can access our images to deploy them after we build)
- `cd /exports/kiwi`
- Make a directories for the build and put the kiwi file in place
- `mkdir -p base/image`
- `cd base`
- `cp -fv ~/SLES12-SP3-Jeos-kvm.kiwi ~/config.sh ./`
- Edit the kiwi file to use local repository

```
<repository type="yast2" alias="SLES-12-SP3-x86_64" imageinclude="false">
    <source path="dir:///exports/SLES12-SP3-x86_64"/>
</repository>
```

- Adjust the package set (some of the JeOS packages are not part of the standard media)
- Change the cracklib-dict-small package to cracklib-dict-full

```
    <package name="cracklib-dict-full"/>
```

- Remove the `jeos-firstboot` package
- Remove the `salt-minion` package

  - Optional if you have access to the Advanced Systems Management module repository

- Initiate the build
- `kiwi -b ./ -d image/`

Exercise 2 - Profiles

Resources

https://doc.opensuse.org/projects/kiwi/doc/#sec.description.profiles

Documentation also available on the virtual machine at `/usr/share/doc/packages/kiwi/html/index.html`

Sample Syntax

```
<profiles>
   <profile name="name" description="text"/>
   <!-- repeat as necessary -->
</profiles>
```

Steps

1. Create profiles for the following:

- KVM
- Xen
- VMware

2. [Optional] Create the following additional profiles

- **NOTE** - The packages these will eventually need are part of the Public Cloud module, not the base SLES media
- Amazon AWS
- Microsoft Azure
- Google Cloud Platform
- OpenStack
- Docker

3. Create package sets based on your profiles

```
<packages type="image" profiles="<comma-separated list of profiles>">
<package name="..."/>>
</packages>
```

- Separate the Xen-specific packages into their own profile-based packages element
- Apply the profiles attribute to the Xen drivers section
- Create a packages element that will only install `open-vm-tools` on VMware images
- [Optional] Create individual packages elements for the public clouds
  - Amazon
  - Azure
  - Google
  - OpenStack
- [Optional] Create a packages element to **delete** the kernel from Docker installs

Exercise 3 - Autoyast

Resources

https://doc.opensuse.org/projects/kiwi/doc/#chap.description

https://www.suse.com/documentation/sles-12/singlehtml/book_autoyast/book_autoyast.html

Documentation also available on the virtual machine at `/usr/share/doc/packages/autoyast2/html/index.ht`

Steps

1. In the directory with your kiwi file, create a file called `config-yast-autoyast.xml`

- Autoyast in Kiwi should not do things that change the structure of the image (partitioning, package installation, users, etc). Basically, if it can be done in the Kiwi XML file, it should not be done in Autoyast
- Used for configuration items (setup network, enable services, etc)
- Easy way to start: run `yast2 autoyast` and export a configuration
- Create an Autoyast file to set a static IP address

Exercise 4 - Hooks

Resources

https://doc.opensuse.org/projects/kiwi/doc/#sec.hooks

Steps

1. Create the directory for the hooks in the same directory as your kiwi file. It **MUST** be called kiwi-hooks

- `mkdir kiwi-hooks`
- All hook scripts go in this directory, and must be named `<hook name>.sh`

2. The vast majority of the available hooks *only* apply to the PXE format (not covered in this lab – it's complex enough that it could be its own class)

- ALL image types:
  - `handleSplash`
  - `preException`
- VMX
  - `preCallInit`
- OEM
  - `preHWdetect` and `postHWdetect`
  - `preImageDump` and `postImageDump`
  - `preCallInit`
  - `preRecovery` and `postRecovery`
  - `preRecoverySetup` and `postRecoverySetup`

3. Create a preCallInit script for your KVM image that will create `/helloworld` and place it in the kiwi-hooks directory

- This hook is called before the initialization process, init or systemd, is started. At call time the root file system has already been moved to /

4. Tar up the kiwi-hooks directory

- `tar -cf kiwi-hooks.tar kiwi-hooks/`
- **NOTE** The documentation says to call the file `kiwi-hooks.tgz`, but since there is no gzip compression being applied here, I corrected it for the purposes of this exercise

5. Add a package reference to the kiwi file

```
<packages type="image">
  <!-- other packages for the image  -->
  <archive name="kiwi-hooks.tar" bootinclude="true"/>
</packages>
```

- The hooks must be part of the initrd to be executed, hence the `bootinclude=true` attribute

Appendix

SLES12-SP3-Jeos-kvm.kiwi

```
<image schemaversion="6.1" name="SLES12-SP3-JeOS-for-kvm-and-xen">
    <description type="system">
        <author>Anja Stock</author>
        <contact>ast@suse.com</contact>
        <specification>SUSE Linux Enterprise 12 SP3 JeOS</specification>
    </description>
    <preferences>
        <version>1.3.0</version>
        <packagemanager>zypper</packagemanager>
        <bootsplash-theme>SLE</bootsplash-theme>
        <bootloader-theme>SLE</bootloader-theme>
    <!-- those settings are applied by suseConfig in config.sh
        <locale>en_US</locale>
        <keytable>us.map.gz</keytable>
        <timezone>Europe/Berlin</timezone>
        <hwclock>utc</hwclock>
    -->
        <rpm-excludedocs>true</rpm-excludedocs>
        <type image="vmx" filesystem="btrfs" boot="vmxboot/suse-SLES12" format="qcow2" bootl
            <systemdisk name="SLES12-SP3-JeOS-for-kvm-and-xen">
                <volume name="home"/>
                <volume name="tmp"/>
                <volume name="opt"/>
                <volume name="srv"/>
                <volume name="boot/grub2/i386-pc"/>
                <volume name="boot/grub2/x86_64-efi" mountpoint="boot/grub2/x86_64-efi"/>
                <volume name="var/cache"/>
                <volume name="var/crash"/>
                <volume name="var/lib/machines"/>
                <volume name="var/lib/mailman"/>
                <volume name="var/lib/named"/>
                <volume name="var/lib/mysql"/> <!-- Requires No CoW, applied by kiwi-hook --
                <volume name="var/lib/mariadb"/> <!-- Requires No CoW, applied by kiwi-hook
                <volume name="var/lib/pgsql"/> <!-- Requires No CoW, applied by kiwi-hook --
                <volume name="var/lib/libvirt/images"/> <!-- Requires No CoW, applied by kiw
                <volume name="usr/local"/>
                <volume name="var/log"/>
                <volume name="var/opt"/>
                <volume name="var/spool"/>
                <volume name="var/tmp"/>
```

```
        </systemdisk>
        <size unit="G">24</size>
    </type>
</preferences>
<repository type="rpm-md">
    <source path="obsrepositories:/"/>
</repository>
<drivers>
    <file name="drivers/xen/*"/>
    <file name="drivers/block/xen-blkfront.ko"/>
    <file name="drivers/net/xen-netfront.ko"/>
    <file name="drivers/scsi/xen-scsifront.ko"/>
    <file name="drivers/input/misc/xen-kbdfront.ko"/>
    <file name="drivers/pci/xen-pcifront.ko"/>
</drivers>
<packages type="image">
    <package name="patterns-sles-Minimal"/>
    <package name="aaa_base-extras"/> <!-- wouldn't be SUSE without that ;-) -->
    <package name="acl"/>
    <package name="btrfsprogs"/>
    <package name="btrfsmaintenance"/>
    <package name="cron"/> <!-- needed by btrfsmaintenance -->
    <package name="curl"/> <!-- needed for openQA, maybe delete -->
    <package name="dracut"/>
    <package name="fipscheck"/>
    <package name="grub2-branding-SLE" bootinclude="true"/>
    <package name="iputils"/>
    <package name="jeos-firstboot"/>
    <package name="zypper-lifecycle-plugin"/> <!-- bsc#1030278 fate#320597 -->
    <package name="vim"/>
    <package name="gettext-runtime"/>
    <package name="shim" arch="x86_64"/>
    <package name="grub2"/>
    <package name="grub2-x86_64-efi" arch="x86_64"/>
    <package name="fontconfig"/>
    <package name="fonts-config"/>
    <package name="haveged"/>
    <package name="less"/>
    <package name="openslp"/>
    <package name="tar"/>
    <package name="parted"/>
    <package name="snapper"/>
    <package name="SUSEConnect"/>
    <package name="SuSEfirewall2"/>
    <package name="systemd"/>
    <package name="systemd-sysvinit"/>
```

```
    <package name="timezone"/>
    <package name="wicked"/>
    <package name="iproute2"/>
    <package name="openssh"/>
    <package name="elfutils"/>
    <package name="kernel-default-base" bootinclude="true" replaces="kernel-default"/>
    <package name="python-base"/>
    <package name="rsync"/>
<package name="salt-minion"/>
    <!-- packages required by file provides, BS can't resolve them -->
    <package name="suse-build-key"/>
    <package name="pkg-config"/>
    <package name="sg3_utils"/>
    <package name="ncurses-utils"/>
    <package name="krb5"/>
    <package name="dialog"/>                <!-- this is needed to support simple dialog
    <package name="grub2-snapper-plugin"/>  <!-- snapper - grub2 - integration, "small"
    <package name="snapper-zypp-plugin"/>   <!-- this is key snapper/zypp integration, c

    <!-- XEN specific packages -->
    <package name="xen-tools-domU"/>
    <package name="grub2-x86_64-xen"/>
    <package name="xen-libs"/>
</packages>
<packages type="bootstrap">
    <package name="udev"/>
    <package name="filesystem"/>
    <package name="glibc-locale"/>
    <package name="cracklib-dict-small"/>
    <package name="ca-certificates"/>
    <package name="sles-release"/>
    <package name="sles-release-DVD"/>
</packages>
<packages type="delete">
    <package name="mtools"/>
    <package name="initviocons"/>
    <package name="cryptsetup"/>
    <package name="autoyast2-installation"/>
    <package name="bind-utils"/>
    <package name="Mesa"/>
    <package name="Mesa-libGL1"/>
    <package name="Mesa-libglapi0"/>
    <package name="Mesa-EGL1"/>
    <package name="Mesa-libEGL1"/>
    <package name="lvm2"/>
    <package name="sg3_utils"/>
```

```
<package name="libcairo2"/>
<package name="libxcb-dri2-0"/>
<package name="libxcb-dri3-0"/>
<package name="libgbm1"/>
<package name="libgio-2_0-0"/>
<package name="libharfbuzz0"/>
<package name="libpango-1_0-0"/>
<package name="libpixman-1-0"/>
<package name="libply-splash-graphics2"/>
<package name="libX11-6"/>
<package name="libX11-xcb1"/>
<package name="libxcb-dri3"/>
<package name="libxcb-present0"/>
<package name="libxcb-shm0"/>
<package name="libxcb-xfixes0"/>
<package name="libdrm_intel1"/>
<package name="libdrm_nouveau2"/>
<package name="libLLVM"/>
<package name="libyui7"/>
<package name="libyui-ncurses-pkg7"/>
<package name="libyui-ncurses7"/>
<package name="libthai0"/>
<package name="libwayland-client0"/>
<package name="libsgutils2"/>
<package name="libpciaccess0"/>
<package name="libgraphite2"/>
<package name="libdatrie1"/>
<package name="libdrm2"/>
<package name="libxcb1"/>
<package name="libX11-data"/>
<package name="libXdamage1"/>
<package name="libXext6"/>
<package name="libXfixes3"/>
<package name="libXft2"/>
<package name="libXrender1"/>
<package name="libXxf86vm1"/>
<package name="libpng16-16"/>
<package name="os-prober"/>
<package name="pango-modules"/>
<package name="plymouth"/>
<package name="plymouth-plugin-label"/>
<package name="plymouth-plugin-script"/>
<package name="plymouth-scripts"/>
<package name="plymouth-branding-SLE"/>
<package name="fontconfig"/>
<package name="fonts-config"/>
```

```
        <package name="gnu-unifont-bitmap-fonts"/>
        <package name="gio-branding-upstream"/>
        <package name="libXau6"/>
        <package name="libfreetype6"/>
        <package name="shared-mime-info"/>
    </packages>
</image>
```