Hands on with KIWI

Build Images/Appliances/Integrated Systems.....

Robert Schweikert
Public Cloud Architect
rjschwei@suse.com

Agenda

- Introduction
  - About me
  - About KIWI
- Prerequisites
  - Install your image
  - KIWI basic presentation 45 – 50 minutes
- Hands on work
  - Build an image with KIWI
- Related topics
  - SUSE Studio
  - OBS
  - Integrated Systems

Introduction

Introduction – About Me

- Work at SUSE – ISV Engineering
  - Public Cloud Architect
  - IBM SWG liaison
  - Integrated Systems

- Contribute to KIWI
  - Code, tests, documentation, architecture
- Contribute to openSUSE
  - Maintain packages
  - Board member
- Passionate about
  - Motorcycling
  - Soccer

Introduction – About KIWI

- It's a Fruit
- It's a Bird
- Someone from New Zealand
- It's a Linux image build system
  - KIWI is an open source project hosted on github https://github.com/openSUSE/kiwi
  - Licensed under GPL v2
  - Mostly written in Perl
  - Sponsored by SUSE
  - Backend of SUSE Studio (http://www.susestudio.com)

Prerequisites

Prereq – Install Your Image

- Everyone should have a USB stick
  - Power down the machine
  - Plug in the USB stick
  - Power back on and boot from USB
  - Watch the magic happen
- This will run while I talk

Prereq – Install Your Image

- What is being installed?
  - SUSE Linux Enterprise Server 11 SP3 (base OS)
  - KIWI build environment
  - Contains the SP3 iso which you will use as repository
- The image being installed
  - Image setup for fully automatic install
    - installboot=”install”, <oem-unattended>true</>, <oem-shutdown>true</>
  - OEM image configured for install from USB
    - installstick=”true”

Interlude – At Home

- 2 versions you might consider
  - Version released with your distribution
  - Weekly releases
- Distribution version
  - SLE – you will find KIWI in the SDK
  - openSUSE – KIWI is part of the standard distribution
- Weekly release
  - openSUSE Build Service (OBS)
    - Virtualization:Appliances
- In YaST or via zypper search for kiwi, install packages

Prereq – The Very Basics

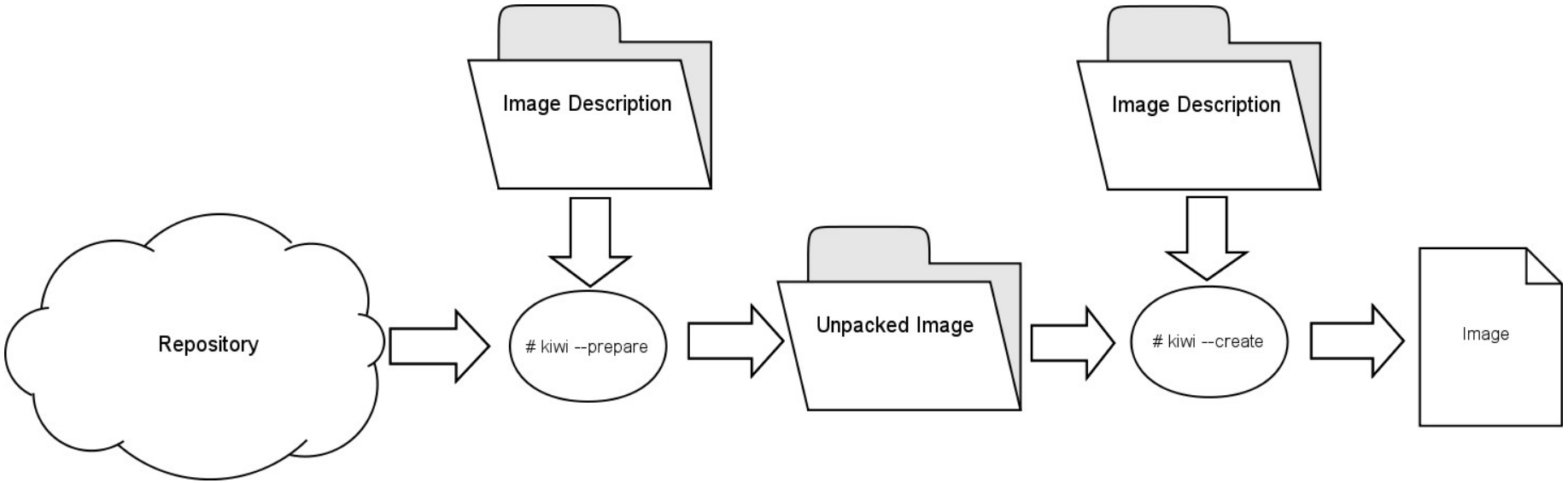
- kiwi is a command line tool
  - SUSE Studio (Online or Onsite) is web app provides GUI
- Most kiwi commands require root privileges
  - Need access to package management
  - Need access to user management
  - Need to create system configuration
- Be cautious with kiwi commands you are root
  - Dedicated build system or VM

Prereq – Docs and Other Help

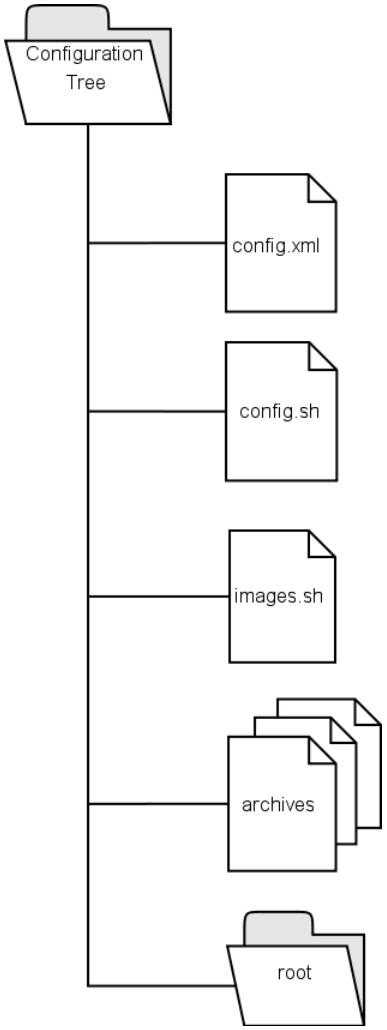
- KIWI User's Manual a.k.a Cookbook
  - http://doc.opensuse.org/projects/kiwi/doc/
  - /usr/share/doc/packages/kiwi/kiwi.pdf (kiwi-doc package)
- KIWI Examples
  - http://en.opensuse.org/Portal:KIWI
- Configuration file schema
  - http://doc.opensuse.org/projects/kiwi/schema-doc/
- Mailing list
  - http://groups.google.com/group/kiwi-images
- IRC (irc.freenode.net)
  - #opensuse-kiwi

Prereq – High Level Overview

- Two step process
  - Prepare step
    - Output is the unpacked image tree (directory)
  - Create step
    - Output is the desired image



Prereq – Configuration Tree



- Configuration is a directory structure (tree)
  - config.xml → required
  - config.sh → optional
  - images.sh → optional
  - archives → optional
  - root → optional

Prereq - Configuration Details

- config.xml – required
  - Configuration file for image attributes, type, users, packages
- config.sh - optional
  - Shell script executed at root level of unpacked tree at the end of prepare. Customize set up, fiddle with files
- images.sh - optional
  - Shell script executed at root level of unpacked tree at the beginning of the create step. Remove drivers, modify based on image type

Prereq - Configuration Details

- archives – optional
  - One or more archives in known format, tar, tar.gz, tar.bz2 etc.
  - Name specified in config.xml with <archive>
  - Extracted at root level of unpacked image tree prior to execution of config.sh
- root – optional directory
  - Root file system representation. This directory is referred to as overlay tree, all files and directories copied to unpacked image tree in the location found in this directory.

Prereq - config.xml

- Contains the image description
  - XML validated against schema
  - Checked for consistency of data
  - When we update the schema we provide XSLT based upgrade functionality
  - Upgrade is automatic

Prereq - config.xml – The Guts

```
<?xml version="1.0" encoding="utf-8"?>
<image schemaversion="" name="">
    <description type="system">
        <author></author>
        <contact></contact>
        <specification></specification>
    </description>
    <preferences>
        <type image="" primary="true" filesystem="" boot="" format=""/>
        <version>1.0.0</version>
        <packagemanager>zypper</packagemanager>
        <rpm-check-signatures>false</rpm-check-signatures>
        <rpm-force>true</rpm-force>
        <locale>en_US</locale>
        <keytable>us.map.gz</keytable>
    </preferences>
```

Prereq - config.xml – The Guts

```
    <users group="root">
      <user pwd="" home="/root" name="root"/>
    </users>
    <repository type="yast2">
      <source path=""/>
    </repository>
    <packages type="image">
      <package name=""/>
    </packages>
    <packages type="bootstrap">
      <package name="filesystem"/>
      <package name="glibc-locale"/>
    </packages>
</image>
```

Prereq - XML Schema

- Implemented in RELAX NG (relaxing) compact notation
  - http://www.relaxng.org
  - http://www.relaxng.org/tutorial-20011203.html
  - http://en.wikipedia.org/wiki/RELAX_NG
- Schema located in
  - /usr/share/kiwi/modules/KIWISchema.rnc
  - Installed as part of the kiwi package
- Parser messages are not always the most useful
  - Jing is helpful in getting better error messages

Prereq - Image Types

- Specified by value of “image” attribute of <type> element in config.xml
- Amazon EC2 (ec2)
  - Support both S3 and EBS backed images
  - Need to also specify settings with <ec2config> element
  - Requires install of Amazon tools
  - For S3 result is an AMI that can be uploaded
  - For EBS result is a disk image that can be dumped onto a volume in EC2 via dd_rescue

Prereq – Image Types

- Linux Container (lxc)
  - Image is a fully functional container with configuration file
    - Unpack the resulting tarball at the root level of the host machine
    - Start the container with lxc start
  - The tarball (result of image build)
    - contains the container
      - Placed in /var/lib/lxc
    - Contains the container config
      - Placed in /etc/lxc

Prereq- Image Types

- Self installing pre-load (oem)
  - Image within an image
    - Inner image is the one configured
    - Outer image is a “live system” that automatically dumps image to target storage
  - Use <oemconfig> child to configure various aspects
  - Partitioning via LVM use <systemdisk> element
  - Create install media for USB stick or optical (CD/DVD)

Prereq - Image Types

- Network install (pxe)
  - Boot image and system image are separated
  - Use requires pxe infrastructure
    - DHCP server
    - atftp server
  - Client configuration via config.<MAC_ADDRESS>

Prereq - Image Types

- Live image (ISO or USB)
  - Iso is separate type specified in config.xml
    - Result is a .iso file than can be burned to optical (CD/DVD)
  - USB
    - Special build, i.e. config.xml specified use “kiwi –bootstick” to deploy
    - Use regular disk image and dump to stick with “dd” command
      - Do not forget “bs=32k” on dd command or the stick will not boot

Prereq - Image Types

- Virtual images (vmx)
  - ec2 → Amazon AMI
  - ovf → Open Virtualization Format
  - ova → Modified ovf
  - qcow2 → Native qemu format
  - vmdk → Native VMWare format also generate .vmx file
  - vhd → Hyper-V format
  - vhd-fixed → Specialized Hyper-V format
  - raw → raw disk image
    - Kiwi always produces a raw disk image
    - No configuration needed

Prereq - Image Types

- Xen (xen)
  - Build guest or host
    - Host
      - Set “bootprofile” attribute to “xen”
      - Set “bootkernel” attribute to “xenk”
      - Set “domain” attribute <xenconfig> to “dom0”
  - Need to install the kernel-xen and other xen packages
  - Generates .xenconfig file

Fingers On The Keyboard

Lab Prereq

- Login to your machine as root
- Password kiwiLab
- Use the repositories from the local SMT server
- Work in the /work directory

Lab Info

- Available editors
  - vi
  - emacs
  - gedit
  - gvim
- You may keep the USB stick you used to install the lab machine

Lab 1

- Lab_1 is incomplete config tree
  - Setup build as virtual machine in qcow2 format
  - Add SLES 11 SP3 repository from SMT server
  - Add a user tux → password SUSECon → group users
  - Find and fix the syntax error
  - Build the image
  - Test with
    - qemu-kvm PATH_TO_IMAGE_FILE

Lab 2

- Lab_2 is incomplete config tree
  - Add OEM image type build as installiso
  - Install the LAMP pattern for the OEM image only
    - The pattern does not specify a DB, thus you need to include one explicitly
  - Setup automatic install
  - All web services should start automatically
    - Hint: fiddle with config.sh
  - Remove the kernel infiniband drivers
    - Hint: Use images.sh
  - Setup eth0 to use a static address from given range
    - Hint: Use root overlay tree
  - Build the image

Lab 2

- Test your image
  - Create a virtual disk
    - qemu-img create webserv.qcow2 -f qcow2 20G
  - Install your webserver image
    - qemu-kvm webserv.qcow2 -net nic,vlan=1,macaddr=00:16:3e:7e:18:XX,model=rtl8139 -net tap,vlan=1,script=/usr/share/qemu-kvm/qemu-ifup -cdrom PATH_TO_YOUR_IMAGE -boot d
  - Log in and verify the services are running

Related Topics

Related Topics – SUSE Studio

- SUSE Studio
  - Hosted at http://www.susestudio.com
  - Also available as product to run behind your firewall
  - Web app to point an click your way to an image configuration
  - Uses KIWI in the back end to build the image of choice
  - Does not support all the features KIWI has

Related Topics - OBS

- OBS – Open Build Service
  - One instance at build.opensuse.org
  - Find needed packages that may not be in SLE repositories
    - There is an OBS presentation covering continuous integration with OBS
  - Build you own packages if you have open source
  - Run your own OBS instance behind your firewall if you have proprietary code
- OBS can build kiwi images

Related Topics – Integrated Systems

- Integrated Systems
  - Scalable and highly customized OEM solutions
  - Image that incorporates OS and your application
    - Deliver to customer in various form factors
    - Plug and play of your application
    - Reduces setup time at end customer
    - You can pre-tune everything to get optimum performance
  - Partners and Customers include:

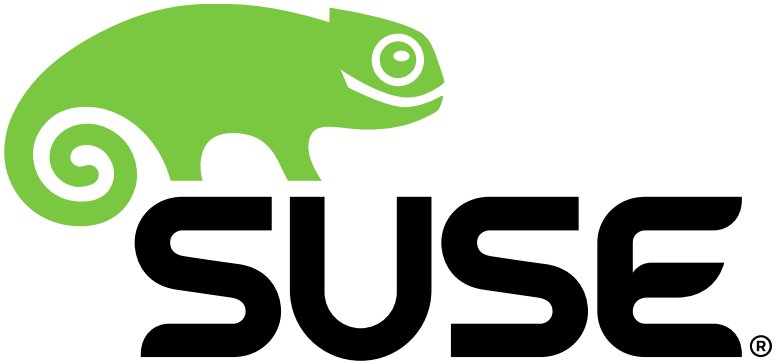
Summary

Summary

- Kiwi is a very flexible image build tool
- Command line driven
  - Easy to integrate into existing build processes
- Support many image formats
  - Can build multiple image formats from one configuration
- Open source licensed under GPL v2
- Mailing list:
  http://groups.google.com/group/kiwi-images
- IRC: #opensuse-kiwi on irc.freenode.net

More Questions....

- Darn this was great but we're out of time and I have so many more questions.....
  - I am happy to answer any questions, just find me at the technology show case or hanging out somewhere in the venue.
  - A longer lab is running during the openSUSE Summit over the weekend



Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org