



Advanced Alchemy

Building Gold-Master Images with Kiwi

Rick Ashford
Sales Engineering Manager – US West
rick.ashford@suse.com



Agenda

What's the problem?

Why is Kiwi the answer?

Kiwi Overview

Advanced Kiwi

Embedded Autoyast

Multiple image types

Using profiles

Hooks

Hands-On Time!



Let's get started

What's the problem?

Creating disk image templates in Linux has traditionally been a manual process

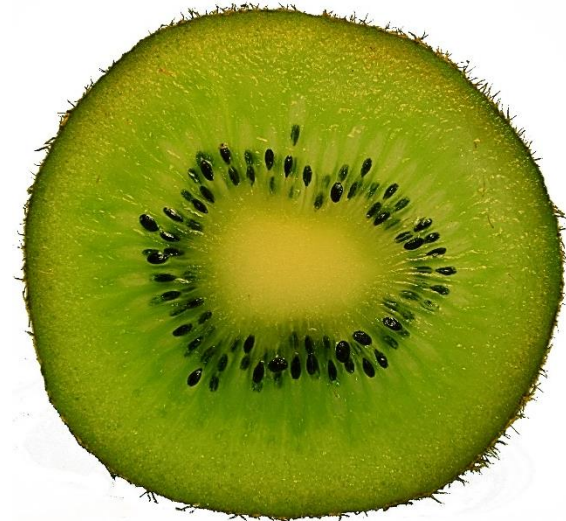


Why is Kiwi the answer?

Kiwi simplifies the process of building images

Benefits:

- **Consistent images across multiple platforms from single source**
- **Easily managed via version-control systems**
- **Application-based images make self-service deployment faster, more responsive**
- **Integrated into Open Build Service**



Kiwi Overview

Image Directory

1. Kiwi file

- config.xml – or – <custom name>.kiwi

2. Overlay

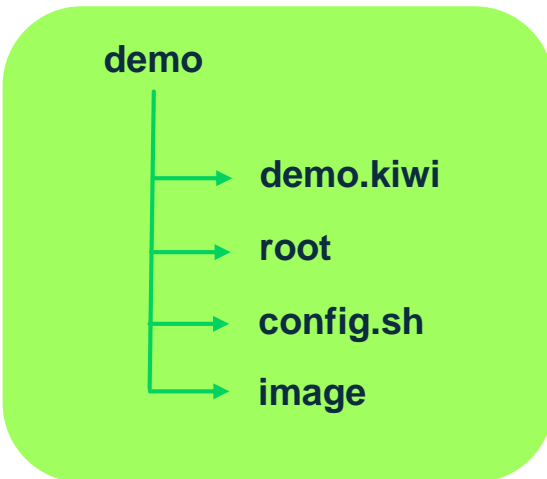
- “root” directory = /

3. Config script

- config.sh

4. Optional: Image build directory

- Keeps build “stuff” organized with configs
- I call mine “image”
- Tip: “build” directory inside “image” directory can be tmpfs



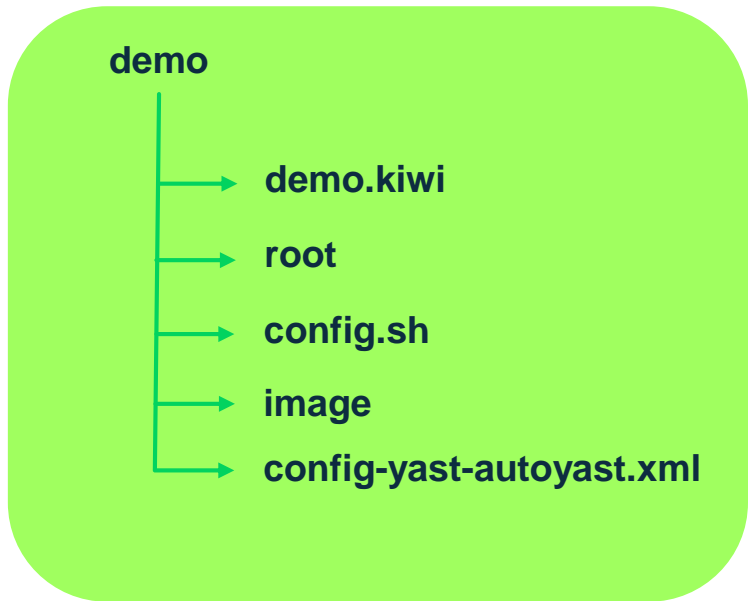
Advanced Kiwi

Customizing a Template

Image Directory

- **AutoYast file**

- config-yast-autoyast.xml
- Executes on-boot after network start
- Used to configure things too specific for a generic template
- NOT used to modify the basic structure of the image
 - Partitioning
 - Package installation

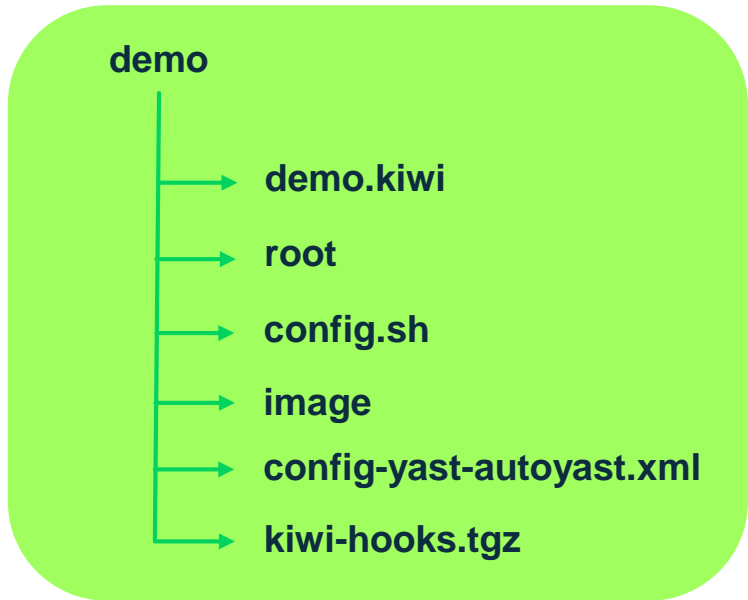


Customizing a Template

Image Directory

- **Hooks**

- kiwi-hooks.tgz
- Embedded into initrd
- Available hooks differs by image type
- Tar file MUST include:
 - Root level directory called “kiwi-hooks”
 - Scripts inside the kiwi-hooks directory
 - No subdirectories
 - Script names must match the hook
 - <hook name>.sh
 - Capitalization matters

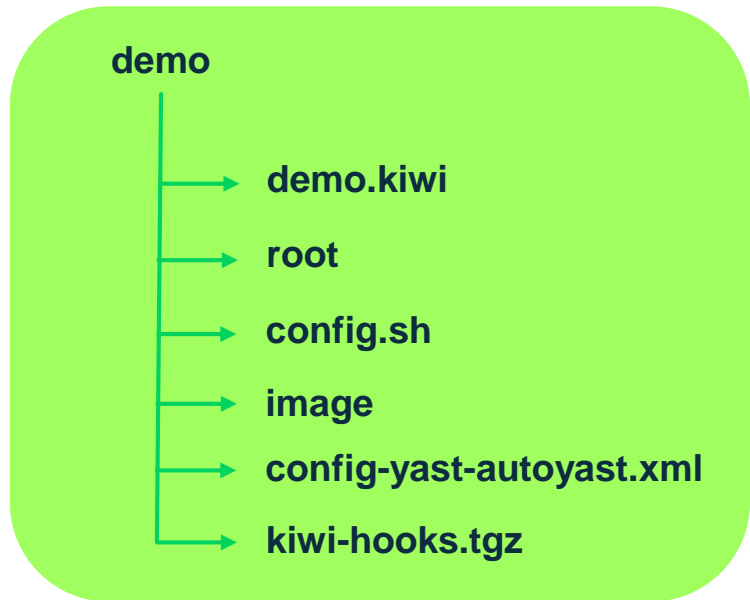


Customizing a Template

Image Directory

- **Hooks**

- ALL
 - handleSplash
 - Called prior to any dialog/exception message or progress dialog. Can be used to customize the behavior of the splash screen.
 - preException
 - Called before a system error is handled



Customizing a Template

Image Directory

- **Hooks**

- OEM
 - preHWdetect | postHWdetect
 - Called before & after boot code detects possible target storage device(s)
 - preImageDump | postImageDump
 - Called before & after the install image is dumped on the target disk
 - preCallInit
 - Called before initialization process (root mounted)
 - preRecovery | postRecovery
 - Called before & after recovery code is processed.
 - preRecoverySetup | postRecoverySetup
 - Called before & after the recovery setup is processed.



Customizing a Template

Image Directory

- **Hooks**
 - VMX
 - preCallInit
 - Called before initialization process (root mounted)

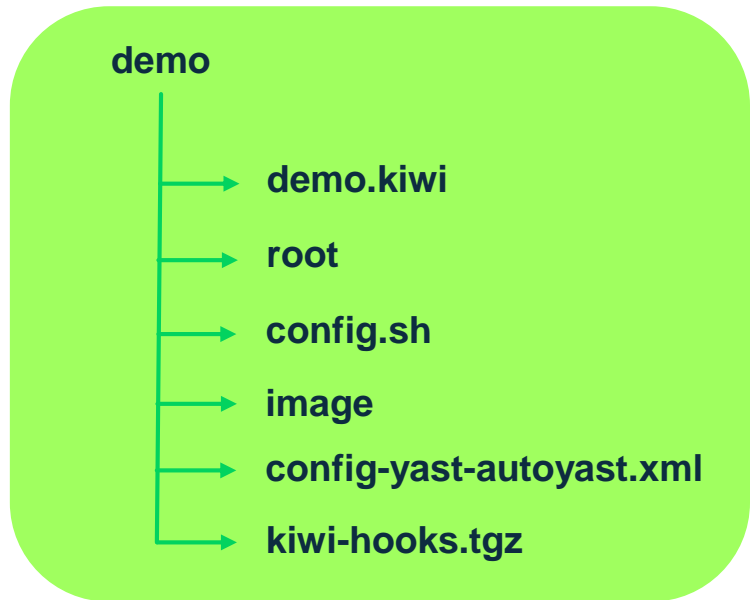


Customizing a Template

Image Directory

- **Hooks**

- Post Commands
 - Run after hook script executes
 - Specified as kernel parameter
 - `KIWI_HOOK_CMD_<hook name>=""`
 - Ex: `KIWI_HOOK_CMD_preHWdetect="ls -l"`
 - Can be disabled via `KIWI_FORBID_HOOK_CMDS=1`



Expanding Your Options

Profiles

Maintain multiple image types in a single image definition while allowing for variation in packages, drivers, and preferences for specific image types

```
<profiles>  
  <profile name="name" description="text"/>  
  <!-- ... -->  
</profiles>
```

```
<packages type="image" profiles="vmware">  
  <package name="open-vm-tools"/>  
</packages>
```

Let's get to it!

LAB

Access all the session materials at:

<https://github.com/darthzen/conferences/tree/master/susecon/2017/HO126885/>

Kiwi Commands:

Use Cache For Speed

```
kiwi --init-cache /usr/share/kiwi/image/vmxboot/suse-SLES12
```

Basic

```
kiwi -b <kiwi file path> -d <destination> --cache /var/cache/kiwi-images
```

Profile

```
kiwi -b <file path> -d <dest> --cache /var/cache/kiwi-images --add-profile <profile> [ --type <image type> ]
```



Unpublished Work of SUSE LLC. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.