25 Automated Deployment of Preload Images

With KIWI you can create operating system images. This chapter describes the process of deploying a system image to an empty client machine. For this, you need to create a preload image which contains a bootable RAW image. This file contains two important parts: a partition table and the actual operating system. This RAW image will be written to the empty hard disk and the operating system extends to the remaining disk space at first boot.

To create such an image, see http://doc.opensuse.org/projects/kiwi/doc/. When you build the ISO image, you can find the RAW file in the destination directory. There are many ways to dump a raw image onto a disk.

- Plug the disk into a deployment server and copy the image to the raw device.
- Provide the raw image by means of an HTTP or FTP server and dump it on the disk of the client machine.
- Create a netboot image to get the image and dump it on the disk. This is a good method for mass deployment.
- Boot a rescue disk and do the dump manually from the rescue image.

For a quick start, it is good to use one of the methods described in *Section 25.1, "Deploying system manually from rescue image"*.

25.1 Deploying system manually from rescue image

Deploying with generated ISO file from KIWI:

1. Burn the ISO image you get from the KIWI building process on CD/DVD.
2. Boot from this medium onto the client machine.
3. Select the hard disk for installation.
4. Restart the client machine and boot from hard disk.

Deploying over rescue system:

1. Boot the client machine with a rescue system. Such systems are available on all SUSE installation CDs or DVDs.

2. Log in as `root`. Do not enter password.

3. Configure your network. If you have DHCP available in your network, this is merely the command **`ifup-dhcp eth0`**. If you must do this manually, use the command **`ip`** to configure your network. The output starting DHCP also tells you the IP address of the computer.

4. Listen on an unused port of your network like `1234` and dump the incoming data to disk with the following command:

```
netcat -l -p 1234 > /dev/sda
```

5. On the imaging server, send the raw image to the client machine with the command:

```
netcat <IP of client> 1234 < $HOME/preload_image/<image_name>
```

6. When the image is transferred, remove the rescue system from your CD or DVD drive and shut down the client machine. On reboot, the boot loader `GRUB` should be started on the client and the firstboot system will take over.

25.2 Automated Deployment with PXE Boot

When doing multiple installations of an operating system on similar hardware, it is useful to put some effort into preparing a mass deployment of the operating system and to minimize the time needed for the actual deployment. This chapter describes this process. The goal is to simply plug in a computer, connect it to a network, start a network boot, and wait until it powers down.

The following actions need to be performed to accomplish this task:

Set up a boot and install server

A dedicated machine is needed, that should be prepared to offer PXE boot and an FTP or Web server to provide a preload image. It is a good idea to give the machine enough memory to hold all necessary installation data in memory. For a default installation, you

should have at least 4 GByte of memory. All the necessary tasks can be accomplished with SUSE Linux Enterprise Server. For more details, see *Section 25.2.1, "Set Up a Boot and Install Server"*.

Prepare a preload Image

The actual installation is done by the copying of a raw image of the operating system to the new hard disk. All features and settings must be prepared and tested carefully. To provide such an image, KIWI can be used (available in the SDK of the SUSE Linux Enterprise operating system). More information about image creation with KIWI is available at http://doc.opensuse.org/projects/kiwi/doc/. For more details about the requirements of the preload image, see *Section 25.2.2, "Creating a Preload Image"*.

KIWI is available from the SUSE Linux Enterprise SDK. The SDK is a module for SUSE Linux Enterprise and is available via an online channel from the SUSE Customer Center. Alternatively, go to http://download.suse.com/, search for `SUSE Linux Enterprise Software Development Kit` and download it from there. Refer to *Chapter 8, Installing Modules, Extensions, and Third Party Add-On Products* for details.

Create an initial system for deployment

This is a task that requires some Linux expertise. A description on how this can be achieved by means of an example installation is available at *Section 25.2.3, "Creating an Initial System to Deploy a Preload Image"*.

Configure the boot server for automatic deployment

PXE boot must be told to boot the installation system, that in turn will take the preload image from the server and copy it to the hard disk.

25.2.1 Set Up a Boot and Install Server

There are four steps to accomplish to perform this task after a SUSE Linux Enterprise Server installation:

1. Set up the installation source as described in *Section 12.2, "Setting Up the Server Holding the Installation Sources"*. Choose an HTTP, or FTP network server.
2. Set up a TFTP server to hold a boot image (this image will be created in a later step). This is described in *Section 12.3.2, "Setting Up a TFTP Server"*.

3. Set up a DHCP server to assign IP addresses to all machines and to reveal the location of the TFTP server to the target system. This is described in *Section 12.3.1, "Setting Up a DHCP Server"*.

4. Prepare the installation server PXE boot. This is described in further detail in *Section 12.3.3, "Using PXE Boot"*.

Note that the actual installation process will greatly benefit if you provide enough memory on this machine to hold the preload image. Also, using gigabit Ethernet will speed up the deployment process considerably (compared to slower networks).

25.2.2 Creating a Preload Image

The process of creating images with KIWI is described at http://doc.opensuse.org/projects/kiwi/doc/. However, to create a useful image for mass deployment, several considerations should be taken into account:

- A typical preload image will use the following type:

```
<type primary="true" filesystem="btrfs" boot="oemboot/suse-SLES12">vmx</type>
```

- During the setup of a preload image, the image creation process is run multiple times. The repositories needed to build the image should be available on the local computer.

- Depending on the desired usage of the preload, some effort should be invested in configuring firstboot. Find more details about firstboot in *Chapter 23, Deploying Customized Preinstallations*. With this method you can also require the user to do initial configurations at the first bootup of the system.

- Many additional features can be configured into the image, like adding update repositories or doing an update on initial bootup. However, it is impossible to describe all possibilities in this document, and (depending on the requirements) the creation of the preload image requires in-depth knowledge of the imaging system KIWI, and several other technologies used in SUSE Linux Enterprise Server.

The actual image to be deployed should be available from the FTP or HTTP server that you provided on the installation server.

25.2.3 Creating an Initial System to Deploy a Preload Image

To run an automatic deployment, it is necessary to start an initial Linux system on the target computer. During a typical installation, the kernel and initial RAM file system are read from some boot medium and started by the bios. The needed functionality can be implemented in the RAM file system , which together with the kernel will serve as the initial system.

The main features that must be provided by the initial system is the enabling of access to the hard disk and the making available of the network connection. Both of these functions are dependent on the hardware onto which you want to deploy. In theory it is possible to create an initial system from scratch, but to simplify this task it is also possible to modify the initial RAM file system used by the machine during boot.

The following procedure is only one example of how to create the needed initial RAM file system:

1. Do a standard installation of SUSE Linux Enterprise Server on the target system.
2. Install the package `busybox` on the system.
3. Create a new RAM file system with the following command:

   ```
   dracut -a busybox
   ```

   The parameter `-a busybox` adds the multi call binary `busybox` to the RAM file system. After doing this, many standard unix commands are available inside this system.
4. Copy the new RAM file system and the kernel to your boot server with the command:

   ```
   scp /boot/initrd /boot/vmlinuz pxe.example.com:
   ```

   Replace pxe.example.com with the name of your local boot server or IP address.
5. Log in to your boot server as user `root`, and create a directory where you can modify the RAM file system:

   ```
   mkdir ~/bootimage
   ```
6. Change your working directory to this directory with the command **`cd ~/bootimage`**.
7. Unpack the previously copied initial RAM file system with the command:

   ```
   zcat ../initrd | cpio -i
   ```

8. Edit the file `run_all.sh`.

9. Search for the following line, delete it and the rest of the file:

```
[ "$debug" ] && echo preping 21-nfs.sh
```

10. Add the following lines to the end of the files `run_all.sh`:

```
[ "$debug" ] && echo preping 92-install.sh
[ "$debug" ] && echo running 92-install.sh
source boot/92-install.sh
[ "$modules" ] && load_modules
```

11. Create a new script `boot/92-install.sh` with the following content:

```
#!/bin/bash
if [ "$(get_param rawimage)" ]; then
  rawimage=$(get_param rawimage)
  if  [ "$(get_param rawdevice)" ]; then
    rawdevice=$(get_param rawdevice)
    echo "wget -O ${rawdevice} ${rawimage}"
    wget -O ${rawdevice} ${rawimage}
    sync
    sleep 5
    echo "DONE"
  fi
fi
# /bin/bash
/bin/poweroff -f
```

12. If you want to have a debug shell before the computer switches off, remove the comment sign before `/bin/bash`.

13. Make this script executable with the command **`chmod 755 boot/92-install.sh`**.

14. Create a new initial RAM file system with the commands:

```
mkdir -p /srv/tftpboot
```

```
find . | cpio --quiet -H newc -o | gzip -9 -n > \
/srv/tftpboot/initrd.boot
```

15. Copy the kernel to this directory:

```
cp ../vmlinuz /srv/tftpboot/linux.boot
```

The initial RAM file system is now prepared to take two new kernel command line parameters. The parameter `rawimage=<URL>` is used to identify the location of the preload image. Any URL that is understood by wget can be used. The parameter `rawdevice=<device>` is used to identify the block device for the hard disk on the target machine.

25.2.4 Boot Server Configuration

The configuration of the boot server is covered in detail in several different chapters as listed in *Section 25.2.1, "Set Up a Boot and Install Server"*. This section should give a checklist that covers steps that are necessary to configure the system.

- Set up a DHCP server. The subnet where the machines are installed needs the additional lines:

  ```
  filename "pxelinux.0";
  next-server 192.168.1.115;
  ```

  In this example, 192.168.1.115 is the ip address of the PXE server pxe.example.com.

- Configure a PXE server as described in *Section 12.3.3, "Using PXE Boot"*. When editing `/srv/tftpboot/pxelinux.cfg/default`, add the following entries:

  ```
  default bootinstall
  label bootinstall
    kernel linux.boot
    append initrd=initrd.boot \
    rawimage=ftp://192.168.1.115/preload/preloadimage.raw rawdevice=/dev/sda
  ```

- Set up an FTP server and copy your prepared preload image to `/srv/ftp/preload/preloadimage.raw`.

Test your setup by booting the target system with PXE network boot. This will automatically copy the prepared preload image to hard disk and switch off the machine when ready.