

RHCE BOOT CAMP

Various Additional Topics



redhat.®

CERTIFIED
E N G I N E E R

BUILDING RPMS

- Building an RPM can be simple or difficult, depending on if it's done incorrectly or correctly. ;)
- Fortunately, for the RHCE exam, you only need to know how to build a simple RPM that packages one file.
- Unfortunately, RPM's were designed to build and package software, so there are lots of extra details that need to be removed to package a simple file.

RPM PACKAGES

- Some important packages to install:

- rpm

Duh. ;)

- rpmdevtools

Very helpful

- rpmlint

Checks rpm's - can be handy

GETTING STARTED

- First, switch to a non-root user.
- Second, run `rpmdev-setuptree` to build a basic directory tree needed for rpm construction.
- `cd rpmbuild`
- Note the various folders - most important right now are SOURCES and SPECS.

SETTING UP SOURCES

- Setting up the SOURCES folder is a little involved:
 - `cd SOURCES`
 - `mkdir rhce-1.0`
 - `echo "test" > rhce-1.0/afile`
 - `tar czf rhce-1.0.tar.gz rhce-1.0`
- This will create your initial “source code” tarball

SETTING UP SPEC FILE

- Writing the spec file is the most difficult part.
- First, cd into the SPECS folder and create a template spec file by running:
 - `rpmdev-newspec rhce.spec`

SPEC FILES

- Spec files have a peculiar syntax:
 - There are tags that have short values associated with them, such as `Name` and `Version`
 - There are sections that are identified with a percent sign followed by a name, such as `%description` and `%prep`
 - There are “macros”, which behave similarly to environment variables: `%{version}` will substitute to the version number entered in the `Version` tag line.

SPEC FILES

- In your spec file, fill out the following areas:
 - Name: rhce
 - Version: 1.0
 - Summary: RPM for RHCE
 - Group: Documentation
 - License: None
 - URL: <http://www.redhat.com>
 - Source0: rhce-1.0.tar.gz

SPEC FILES

- You do not have any requirements, so just delete the lines:
 - `Requires`
 - `BuildRequires`
- For the Description, put a short, meaningful message:
 - `RHCE Exam RPM file`

SPEC FILES

- Remove the `%configure` macro, as this just tries to call `configure` for you automatically, which is not needed for our simple rpm.
- Also, remove the make lines - one is under `%build`, one is under `%install`. Same reasoning - we don't need make for our rpm.
- Under `%install`, below the `rm -rf`, add:
 - `mkdir -p $RPM_BUILD_ROOT`
 - `cp afile $RPM_BUILD_ROOT`

SPEC FILES

- Under `%files`, replace the `%doc` with:
 - `/afile`
- Verify all of your spec file contents

BUILD THE RPM

- Trial by fire! Build the rpm from the `rpmbuild` folder:
 - `rpmbuild -ba SPECS/rhce.spec`
- If your spec file is good, and your SOURCES tar file, you will have a new rpm under the RPMS folder
- Verify new rpm with:
 - `rpm -qlp RPMS/*/*.rpm`
- You should see the single pathname “/afile”. Install if you wish.

LAB

1. Build a simple rpm that packages a file called “I-rock-rpms” and installs it to /.
2. Install your rpm and verify /I-rock-rpms exists.

ISCSI

- iSCSI is a neat protocol which allows for the transport of SCSI commands over standard network stacks, such as TCP/IP.
- In iSCSI parlance, a “target” is a server/device that accepts commands and relays them to a storage system. An “initiator” is a client which sends commands to a target.
- For the RHCE exam, all you need to know is how to set up an initiator.

ISCSI

- Required package: `iscsi-initiator-utils`
- This provides the `iscsid` and `iscsi` services. `iscsid` manages the low level iSCSI communications, and `iscsi` automatically logs in and out of targets.
- You usually only want to start/stop `iscsi`, as it will take care of `iscsid`.

ISCSI

- Once the iSCSI package is installed, connecting to a target is super simple:
 - `iscsiadm -m discovery -t st -p <ip>`
- If any targets are discovered, they will be printed back, as:
 - `192.168.1.100:3260,1 iqn.2011-04.com.example.server1:server1.target1`
- This shows a single target on server1 is available.

ISCSI

- Once targets are discovered, they will be remembered. You can see your known targets with:
 - `iscsiadm -m node -o show`
- Once targets are found, start up the iscsi service:
 - `service iscsi start`
- Check dmesg to verify it finds and attaches the new SCSI devices.

ISCSI

- Once you identify the scsi device (`/dev/sdb` on our machines), you can partition, format and roll:
 - `fdisk /dev/sdb`
 - `mkfs /dev/sdb1`
 - *Add entry to `fstab` and mount!*

LAB

1. Check server1 for available iSCSI targets. You should see exactly one, and the target number will match your station number.
2. Attach the iSCSI device, partition it, build an ext4 filesystem and set it to mount at boot to `/iscsi`. Don't forget to `chkconfig iscsi on`!
3. Reboot your machine and verify the iSCSI filesystem comes up automatically.


```
slideshow.end();
```