



redhat.[®]

RHCSA Rapid Track Course
Student Workbook
Red Hat Enterprise Linux 6
RH200 2-20110115

RED HAT TRAINING

Red Hat Enterprise Linux 6 RH200

RHCSA Rapid Track Course

Edition 2

| | |
|--------|-------------------|
| Author | Bowe Strickland |
| Author | George Hacker |
| Author | Joshua Hoffman |
| Author | Robert Locke |
| Author | Brad Smith |
| Author | Forrest Taylor |
| Editor | Steven Bonneville |
| Editor | Mark Howson |

Copyright © 2011 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2011 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please e-mail training@redhat.com or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, Hibernate, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

All other trademarks are the property of their respective owners.

Contributors: Brian Butler, Victor Costea, Andrew Dingman, Chris Negus



| | |
|---|-------|
| Document Conventions | vii |
| Notes and Warnings | vii |
| Introduction | ix |
| Welcome to class! | ix |
| About Red Hat Enterprise Linux | ix |
| Additional Red Hat Enterprise Linux Software | x |
| Contacting Red Hat Technical Support | xii |
| About This Course | xv |
| RHCSA Rapid Track Course | xv |
| Structure of the Course | xv |
| Orientation to the Classroom Network | xvi |
| Internationalization | xvii |
| Language Support | xvii |
| System-wide Default Language | xvii |
| Per-user Language Selection | xvii |
| Input Methods | xviii |
| Language Codes Reference | xviii |
| 1. Network Configuration and Troubleshooting | 1 |
| Understanding Network Configuration Files | 2 |
| Basic Troubleshooting Process | 8 |
| Network Troubleshooting Toolkit | 10 |
| Criterion Test | 13 |
| 2. Administering Users and Groups | 17 |
| Managing Local Users and Groups | 18 |
| Managing Passwords | 23 |
| Network Authentication Using an LDAP Server | 26 |
| Network Mounting Home Directories | 30 |
| Criterion Test | 33 |
| 3. Command-line Process Management | 37 |
| Launching Graphical Tools from Bash | 38 |
| Monitoring Processes | 41 |
| Terminating and Governing Processes | 42 |
| Managing Periodic Tasks | 45 |
| Criterion Test | 48 |
| 4. Get Help from Red Hat | 51 |
| Research On-line Documentation | 52 |
| Getting the Most from Red Hat Global Support Services | 54 |
| 5. Manage System Resources | 59 |
| Determine Log Destinations | 60 |
| Locate and Analyze a Log Summary Report | 64 |
| Change the Log Summary e-mail Address | 66 |
| Criterion Test | 69 |
| 6. Installing and Managing Software | 73 |
| Register with Red Hat Network (RHN) | 74 |
| Using yum | 78 |
| Handling Third-Party Software | 81 |

| | |
|--|------------|
| Using Third-Party Repositories | 85 |
| Criterion Test | 89 |
| 7. Administer Remote Systems | 93 |
| Remote Shell Access | 94 |
| Remote File Transfers | 96 |
| Using SSH Keys | 98 |
| Securing SSH Access | 101 |
| Archives and Compression | 103 |
| Criterion Test | 106 |
| 8. Deploy and Secure Services | 109 |
| Manage the System Clock | 110 |
| Manage Services | 112 |
| Configuring a VNC Server | 113 |
| Secure Access to a Remote GNOME Desktop | 115 |
| Deploy an FTP Server | 117 |
| FTP Server Configuration | 119 |
| Deploy a Web Server | 121 |
| Protect Services with a Firewall | 123 |
| Criterion Test | 125 |
| 9. SELinux Management | 129 |
| Basic SELinux Security Concepts | 130 |
| SELinux Modes | 133 |
| Display and Modify SELinux Modes | 136 |
| Display and Modify SELinux File Contexts | 138 |
| Managing SELinux Booleans | 141 |
| Monitoring SELinux Violations | 142 |
| Criterion Test | 145 |
| 10. Managing Simple Partitions and File Systems | 149 |
| Simple Partitions and File Systems | 150 |
| Enabling Data Privacy with Partition Encryption | 154 |
| Managing Swap Space | 158 |
| Criterion Test | 161 |
| 11. Controlling Access to Files | 165 |
| Managing File System Access Control Lists | 166 |
| Criterion Test | 170 |
| 12. Managing Flexible Storage with Logical Volume Manager | 173 |
| Recognize the Components of LVM | 174 |
| Implement LVM Storage with Command-line Tools | 178 |
| Extend a Logical Volume and Ext4 File System | 181 |
| Extending and Reducing a Volume Group | 185 |
| Create a Snapshot to Facilitate Data Backup | 187 |
| Criterion Test | 190 |
| 13. Control the Boot Process | 193 |
| Booting an Alternate Kernel | 194 |
| Booting into a Different Runlevel | 196 |
| Resolve GRUB Issues | 198 |
| Making Persistent GRUB Changes | 201 |

| | |
|---|------------|
| Passing Kernel Arguments | 203 |
| Changing the Default Runlevel | 205 |
| Repairing Boot Issues | 207 |
| Criterion Test | 211 |
| 14. Tuning and Maintaining the Kernel | 215 |
| Supported Architectures and Kernel Identification | 216 |
| Managing Kernel Modules | 218 |
| Upgrading Your Kernel | 220 |
| 15. Manage Virtual Machines | 225 |
| Introduction to KVM Virtualization | 226 |
| Virtual Guest Installation | 228 |
| Configuring Guests to Start at Boot Time | 230 |
| Criterion Test | 232 |
| 16. Automated Installation of Red Hat Enterprise Linux | 235 |
| Introductory Overview | 236 |
| Create a Kickstart File with system-config-kickstart | 237 |
| Make the Kickstart File Available to Installers | 239 |
| Create Boot Media | 241 |
| Point the Installer to a Kickstart File | 242 |
| Modify a Kickstart File | 245 |
| Criterion Test | 248 |
| A. Solutions | 251 |
| Network Configuration and Troubleshooting | 251 |
| Administering Users and Groups | 255 |
| Command-line Process Management | 261 |
| Get Help from Red Hat | 265 |
| Manage System Resources | 266 |
| Installing and Managing Software | 269 |
| Administer Remote Systems | 273 |
| Deploy and Secure Services | 280 |
| SELinux Management | 287 |
| Managing Simple Partitions and File Systems | 292 |
| Controlling Access to Files | 300 |
| Managing Flexible Storage with Logical Volume Manager | 303 |
| Control the Boot Process | 311 |
| Tuning and Maintaining the Kernel | 318 |
| Manage Virtual Machines | 320 |
| Automated Installation of Red Hat Enterprise Linux | 328 |

Document Conventions

Notes and Warnings



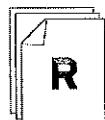
Note

"Notes" are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Comparison

"Comparisons" look at similarities and differences between the technology or topic being discussed and similar technologies or topics in other operating systems or environments.



References

"References" describe where to find external documentation relevant to a subject.



Important

"Important" boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled "Important" will not cause data loss, but may cause irritation and frustration.



Warning

"Warnings" should not be ignored. Ignoring warnings will most likely cause data loss.

Introduction

Welcome to class!

Thank you for attending this Red Hat training class. Please let us know if you have any special needs while at our training facility.

Please ask the instructor if you have any questions about the facility, such as operating hours of the facility and when you will have access to the classroom, locations of restrooms and break rooms, availability of telephones and network connectivity, and information about the local area.

As a courtesy to other students, please place your pager or cell phone's ringer on vibrate or mute, or turn off your devices during class. We ask that you only make calls during break periods.

If you have a personal emergency and are unable to attend or complete the class, please let us know. Thank you!

About Red Hat Enterprise Linux

This course is taught using Red Hat Enterprise Linux, an enterprise-targeted Linux distribution focused on mature open source software designed specifically for organizations using Linux in production settings.

Red Hat Enterprise Linux is sold on a subscription basis, where the subscription gives you continuous access to all supported versions of the operating system in binary and source form, not just the latest one, including all updates and bug fixes. Extensive support services are included: a support contract and Update Module entitlement to Red Hat Network are included for the subscription period. Various Service Level Agreements are available that may provide up to 24x7 coverage with a guaranteed one hour response time for Severity 1 issues. Support will be available for up to seven years after a particular major release (ten years with the optional "Extended Update Support" Add-On).

Red Hat Enterprise Linux is released on a multi-year cycle between major releases. Minor updates to major releases are released roughly every six months during the lifecycle of the product. Systems certified on one minor update of a major release continue to be certified for future minor updates of the major release. A core set of shared libraries have APIs and ABIs which will be preserved between major releases. Many other shared libraries are provided, which have APIs and ABIs which are guaranteed within a major release (for all minor updates) but which are not guaranteed to be stable across major releases.

Red Hat Enterprise Linux is based on code developed by the open source community, which is often first packaged through the Red Hat sponsored, freely-available Fedora distribution (<http://fedoraproject.org/>). Red Hat then adds performance enhancements, intensive testing, and certification on products produced by top independent software and hardware vendors. Red Hat Enterprise Linux provides a high degree of standardization through its support for four processor architectures (32-bit Intel x86-compatible, AMD64/Intel 64 (x86-64), IBM POWER, and IBM mainframe on System z). Furthermore, we support the 4000+ ISV certifications on Red Hat Enterprise Linux whether the RHEL operating system those applications are using

is running on "bare metal", in a virtual machine, as a software appliance, or in the cloud using technologies such as Amazon EC2.

Currently, the Red Hat Enterprise Linux product family includes:

- *Red Hat Enterprise Linux for Servers*: the datacenter platform for mission-critical servers running Red Hat Enterprise Linux. This product includes support for the largest x86-64 and x86-compatible servers and the highest levels of technical support, deployable on bare metal, as a guest on the major hypervisors, or in the cloud. Subscriptions are available with flexible guest entitlements of one, four, or unlimited guests per physical host. Pricing is based on the basis of the number of socket-pairs populated on the system motherboard, the number of guests supported, the level of support desired, and the length of subscription desired.

Red Hat Enterprise Linux for IBM POWER and *Red Hat Enterprise Linux for IBM System z* are similar variants intended for those system architectures.

- *Red Hat Enterprise Linux Desktop*: built for the administrator and end-user, Red Hat Enterprise Linux Desktop provides an attractive and highly productive environment for knowledge workers on desktops and laptops. Client installations can be finely tailored and locked down for simplicity and security for any workstation task.

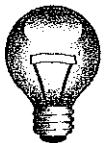
The basic *Desktop* variant is designed for task workers who have a limited amount of administrative control over the system, who primarily use productivity applications like Firefox Evolution/Thunderbird, OpenOffice.org, and Planner/TaskJuggler. The more sophisticated *Workstation* variant is designed for advanced Linux users who need a stand-alone development environment, and who are expected to have local super-user privileges or selected super-user privileges.

In addition, other variants exist such as *Red Hat Enterprise Linux for HPC Head Node* and *Red Hat Enterprise Linux for HPC Compute Node* (targeted at high-performance computing clusters), and *Red Hat Enterprise Linux for SAP Business Applications*. For more information please visit <http://www.redhat.com/>.

Additional Red Hat Enterprise Linux Software

Two additional software update channels are provided with Red Hat Enterprise Linux beyond the core software packages shipped:

- *Supplementary*: the "Supplementary" channel provides selected closed source packages, built for Red Hat Enterprise Linux as a convenience to the customer. These include things like Adobe Flash or proprietary Java JVMs.
- *Optional*: the "Optional" channel provides selected open source packages, as a convenience only. They are generally included in another Red Hat Enterprise Linux variant as a fully-supported package, or are a build requirement for the distribution. These packages are only available through a Red Hat Network child channel.



Important

Supplementary and *Optional* packages are provided with limited support, as a customer convenience only.

Red Hat also offers a portfolio of fully-supported *Add-Ons for Red Hat Enterprise Linux* which extend the features of your Red Hat Enterprise Linux subscription. These add-ons allow you to add capabilities and tailor your computing environment to your particular needs. These Add-Ons include support for high availability application clustering, cluster file systems and very large file systems, enhanced system management with Red Hat Network, extended update support, and more.



Note

Please visit <http://www.redhat.com/rhel/add-ons/> for more information about available *Add-Ons for Red Hat Enterprise Linux*.

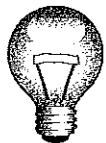
For information about other products which are provided by Red Hat, such as Red Hat Enterprise Virtualization, JBoss Enterprise Middleware, Red Hat Enterprise MRG, and various custom consulting and engineering services, <http://www.redhat.com/products/> also has useful information.

The Fedora Project also provides additional packages for Red Hat Enterprise Linux through *EPEL* (*Extra Packages for Enterprise Linux*). EPEL is a volunteer-based community effort to create a repository of high-quality add-on packages which can be used with Red Hat Enterprise Linux and compatible derivatives. It accepts legally-unencumbered free and open source software which does not conflict with packages in Red Hat Enterprise Linux or Red Hat add-on products. EPEL packages are built for a particular major release of Red Hat Enterprise Linux and will be updated by EPEL for the standard support lifetime of that major release.

Red Hat does not provide commercial support or service level agreements for EPEL packages. While not supported officially by Red Hat, EPEL provides a useful way to reduce support costs for unsupported packages which your enterprise wishes to use with Red Hat Enterprise Linux. EPEL allows you to distribute support work you would need to do by yourself across other organizations which share your desire to use this open source software in RHEL. The software packages themselves go through the same review process as Fedora packages, meaning that experienced Linux developers have examined the packages for issues. As EPEL does not replace or conflict with software packages shipped in RHEL, you can use EPEL with confidence that it will not cause problems with your normal software packages.

For developers who wish to see their open source software become part of Red Hat Enterprise Linux, often a first stage is to sponsor it in EPEL so that RHEL users have the opportunity to use it, and so experience is gained with managing the package for a Red Hat distribution.

Visit <http://fedoraproject.org/wiki/EPEL/> for more information about EPEL.



Important

EPEL is supported by the community-managed Fedora Project and not by Red Hat Support.

Contacting Red Hat Technical Support

One of the benefits of your subscription to Red Hat Enterprise Linux is access to technical support through Red Hat's customer portal at <http://access.redhat.com/>. If you do not have a Red Hat account on the customer portal or are not able to log in, you can go to <https://access.redhat.com/support/faq/LoginAssistance.html> or contact Customer Service for assistance.

You may be able to resolve your problem without formal technical support by searching Knowledgebase (<https://access.redhat.com/kb/knowledgebase/>). Otherwise, Red Hat Support may be contacted through a web form or by phone depending on your support level. Phone numbers and business hours for different regions vary; see <https://access.redhat.com/support/contact/technicalSupport.html> for current information. Information about the support process is available at https://access.redhat.com/support/policy/support_process.html.

Some tips on preparing your bug report to most effectively engage Red Hat Support:

- *Define the problem.* Make certain that you can articulate the problem and its symptoms before you contact Red Hat. Be as specific as possible, and detail the steps you can use (if any) to reproduce the problem.
- *Gather background information.* What version of our software are you running? Are you using the latest update? What steps led to the failure? Can the problem be recreated and what steps are required? Have any recent changes been made that could have triggered the issue? Were messages or other diagnostic messages issued? What exactly were they (exact wording may be critical)?
- *Gather relevant diagnostic information.* Be ready to provide as much relevant information as possible; logs, core dumps, traces, the output of **sosreport**, etc. Technical Support can assist you in determining what is relevant.
- *Determine the Severity Level of your issue.* Red Hat uses a four-level scale to indicate the criticality of issues; criteria may be found at https://access.redhat.com/support/policy/GSS_severity.html.



Warning

Bugzilla is not a support tool! For support issues affecting Red Hat Enterprise Linux, customers should file their bugs through the support channels discussed above in order to ensure that Red Hat is fully aware of your issue and can respond under the terms of your Service Level Agreement. Customers should *not* file bugs directly in the <http://bugzilla.redhat.com/> web interface.

For Red Hat Enterprise Linux, Bugzilla is used by engineering to track issues and changes, and to communicate on a technical level with Engineering partners and other external parties. Anyone, even non-customers, can file issues against Bugzilla, and Red Hat does monitor them and review them for inclusion in errata.

However, Red Hat does not guarantee any SLA for bugs filed directly in Bugzilla (bypassing normal support channels). A review might happen immediately, or after a time span of any length. Issues coming through Support are always prioritized above issues of similar impact and severity filed against Bugzilla. Also, workarounds and hotfixes if possible and appropriate may be provided to customers by Support even before a permanent fix is issued through Red Hat Network.

Red Hat considers issues directly entered into Bugzilla important feedback, and it allows us to provide efficient interaction with the open source development community and as much transparency as possible to customers as issues are processed. Nevertheless, for customers encountering production issues in Red Hat Enterprise Linux, Bugzilla is not the right channel.

About This Course

RHCSA Rapid Track Course

RHCSA Rapid Track Course (RH200) is designed for students who already have significant experience with Linux system administration. The *RHCSA Rapid Track Course* reviews the tasks covered in *Red Hat System Administration I* (RH124) and *Red Hat System Administration II* (RH135) at an accelerated pace. This course builds on the student's existing understanding of command-line based Linux system administration; alternatives based on graphical user interfaces are not a primary feature of this course.

Students should be able to execute common commands such as **cp**, **grep**, **sort**, **mkdir**, **tar**, **mkfs**, **ssh**, and **yum** from the shell prompt. Students should also be familiar with working with common command options and with accessing and reading man pages for help. Students lacking this knowledge are strongly encouraged to take *Red Hat System Administration I* (RH124).

Objectives

- Provide students who have some Linux system administration experience with an accelerated learning environment to round out their skill set
- To prepare students to validate their skills in the RHCSA exam

Audience and Prerequisites

- Students who have one to three years of full-time Linux system administration experience

Structure of the Course

Red Hat training courses are interactive, hands-on, performance-based, real world classes meant to engage your mind and give you an opportunity to use real systems to develop real skills. We encourage students to participate in class and ask questions in order to get the most out of their training sessions.

This course is divided up into a number of *Units* organized around a particular topic area. Each Unit is divided up into multiple *Sections* which focus on a specific skill or task. The unit will start with an introduction to the material, then move on to the first section.

In each section, there will be a *presentation* led by the instructor. During the presentation, it may be a good idea to take notes in your student workbook (this book), and the instructor may remind you to do so. The presentation is followed by a short activity or *assessment* to give you the opportunity to practice with the material or review procedures. After a review of the assessment, the instructor will move on to the next section. At the end of the unit, there will normally be a hands-on lab exercise of some sort (a "criterion test") which will give you an opportunity to learn by doing and review your understanding of the unit's content. Please feel free ask questions in class, or asking the instructor for advice and help during the end-of-unit exercise. We want the

classroom environment to be a "low risk" place where you feel comfortable asking questions and learning from things that work and things that do not at first.

Orientation to the Classroom Network

Two subnets may be used in this course. The primary classroom network is 192.168.0.0/24, and belongs to hosts in the DNS domain "example.com". This network will be used for most classroom activities. Some courses use a second subnet, 192.168.1.0/24, belonging to hosts in the DNS domain "remote.test". This network can be reached from hosts in example.com, and is used in lab exercises which require testing services or security settings from machines (theoretically) outside your administrative control.

Students are each assigned a physical machine (desktopX.example.com on 192.168.0.X) which may host two or more virtual machines for lab activities, serverX.example.com and hostX.example.com.

In some courses, students may also use a non-root account on a test machine in the remote.test domain, remoteX.example.com (192.168.1.X) to test access to network services on their example.com machines in lab activities.

The instructor controls a number of machines which students may see as well. The machine instructor.example.com (also known as instructor.remote.test) is the classroom utility server, providing default routing services, DHCP, DNS name service, one or more YUM repositories of software used by the class, and other network services. It is also connected to the classroom video projector to allow the instructor to display slides and demonstrations. It provides a virtual machine for the instructor, demo.example.com, which the instructor will use for in-class demonstrations.

DOS key? ^

| Machine name | IP addresses | Role |
|------------------------|-------------------|---|
| desktopX.example.com | 192.168.0.X | Physical student workstation |
| serverX.example.com | 192.168.0.(X+100) | Main student virtual machine |
| hostX.example.com | 192.168.0.(X+200) | Secondary student virtual machine |
| remoteX.remote.test | 192.168.1.X | Student test machine in remote.test domain (shared) |
| instructor.example.com | 192.168.0.254 | Physical instructor machine and utility server |
| instructor.remote.test | 192.168.1.254 | Identity of instructor.example.com on remote.test network |
| demo.example.com | 192.168.0.250 | Instructor virtual demonstration machine |

Table 1. Classroom Machines

Internationalization

Language Support

Red Hat Enterprise Linux 6 officially supports twenty-two languages: English, Assamese, Bengali, Chinese (Simplified), Chinese (Traditional), French, German, Gujarati, Hindi, Italian, Japanese, Kannada, Korean, Malayalam, Marathi, Oriya, Portuguese (Brazilian), Punjabi, Russian, Spanish, Tamil, and Telugu. Support for Maithili, Nepalese, and Sinhala are provided as Technology Previews.

System-wide Default Language

The operating system's default language is normally set to US English (en_US.UTF-8), but this can be changed during or after installation.

To use other languages, you may need to install additional package groups to provide the appropriate fonts, translations, dictionaries, and so forth. By convention, these package groups are always named ***language-support***. These package groups can be selected during installation, or after installation with PackageKit (System → Administration → Add/Remove Software) or **yum**.

A system's default language can be changed with **system-config-language** (System → Administration → Language), which affects the **/etc/sysconfig/i18n** file.

Per-user Language Selection

Users may prefer to use a different language for their own desktop environment or interactive shells than is set as the system default. This is indicated to the system through the **LANG** environment variable.

This may be set automatically for the GNOME desktop environment by selecting a language from the graphical login screen by clicking on the **Language** item at the bottom left corner of the graphical login screen immediately prior to login. The user will be prompted about whether the language selected should be used just for this one login session or as a default for the user from now on. The setting is saved in the user's **~/.dmrc** file by GDM.

If a user wants to make their shell environment use the same **LANG** setting as their graphical environment even when they login through a text console or over **ssh**, they can set code similar to the following in their **~/.bashrc** file. This code will set their preferred language if one is saved in **~/.dmrc** or will use the system default if one is not:

```
i=$(grep 'Language=' ${HOME}/.dmrc | sed 's/Language=//')
if [ "$i" != "" ]; then
    export LANG=$i
fi
```

Languages with non-ASCII characters may have problems displaying in some environments. Kanji characters, for example, may not display as expected on a virtual console. Individual commands can be made to use another language by setting **LANG** on the command-line:

```
[user@host ~]$ LANG=fr_FR.UTF-8 date
lun. oct. 24 10:37:53 CDT 2011
```

Subsequent commands will revert to using the system's default language for output. The **locale** command can be used to check the current value of **LANG** and other related environment variables.

Input Methods

IBus (Intelligent Input Bus) can be used to input text in various languages under X if the appropriate language support packages are installed. You can enable IBus with the **im-chooser** command (System → Preferences → Input Method).

Language Codes Reference

| Language | \$LANG value | Language package group |
|------------------------|--------------|------------------------|
| English (US) | en_US.UTF-8 | (default) |
| Assamese | as_IN.UTF-8 | assamese-support |
| Bengali | bn_IN.UTF-8 | bengali-support |
| Chinese (Simplified) | zh_CN.UTF-8 | chinese-support |
| Chinese (Traditional) | zh_TW.UTF-8 | chinese-support |
| French | fr_FR.UTF-8 | french-support |
| German | de_DE.UTF-8 | german-support |
| Gujarati | gu_IN.UTF-8 | gujarati-support |
| Hindi | hi_IN.UTF-8 | hindi-support |
| Italian | it_IT.UTF-8 | italian-support |
| Japanese | ja_JP.UTF-8 | japanese-support |
| Kannada | kn_IN.UTF-8 | kannada-support |
| Korean | ko_KR.UTF-8 | korean-support |
| Malayalam | ml_IN.UTF-8 | malayalam-support |
| Marathi | mr_IN.UTF-8 | marathi-support |
| Oriya | or_IN.UTF-8 | oriya-support |
| Portuguese (Brazilian) | pt_BR.UTF-8 | brazilian-support |
| Punjabi | pa_IN.UTF-8 | punjabi-support |
| Russian | ru_RU.UTF-8 | russian-support |

| Language | \$LANG value | Language package group |
|----------------------------|--------------|------------------------|
| Spanish | es_ES.UTF-8 | spanish-support |
| Tamil | ta_IN.UTF-8 | tamil-support |
| Telugu | te_IN.UTF-8 | telugu-support |
| <i>Technology Previews</i> | | |
| Maithili | mai_IN.UTF-8 | maithili-support |
| Nepali | ne_NP.UTF-8 | nepali-support |
| Sinhala | si_LK.UTF-8 | sinhala-support |

Table 2. Language Codes

iptables -I ROSTROUTING at nat -o eth1 -j MASQUERADE



UNIT ONE

NETWORK CONFIGURATION AND TROUBLESHOOTING

Introduction

Topics covered in this unit:

- Network Configuration Files
- Basic Troubleshooting Process
- Network Troubleshooting Toolkit

Understanding Network Configuration Files

Network Interface Names

The Linux kernel names interfaces with a specific prefix depending on the type of interface. For example, all Ethernet interfaces start with **eth**, regardless of the specific hardware vendor. Following the prefix, each interface is numbered, starting at zero. For example, **eth0**, **eth1**, and **eth2** would refer to the first, second, and third Ethernet interfaces. Other interface names include **wlan0** for the first wireless device, **virbr0** for the internal bridge set up for virtual hosts, **bond0** for the first bonded network device, and so on.

Network Interface Configuration

/sbin/ip is used to show or temporarily modify devices, routing, policy routing, and tunnels.

```
[root@demo ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:00:fa brd ff:ff:ff:ff:ff:ff
        inet 192.168.0.250/24 brd 192.168.0.255 scope global eth0
            inet6 fe80::5054:ff:fe00:fa/64 scope link
                valid_lft forever preferred_lft forever
[root@demo ~]# ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:00:fa brd ff:ff:ff:ff:ff:ff
        RX: bytes   packets   errors   dropped overrun mcast
        91449      520       0        0        0        0
        TX: bytes   packets   errors   dropped carrier collsns
        14020      99        0        0        0        0
[root@demo ~]# ip route
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.250 metric 1
default via 192.168.0.254 dev eth0 proto static
```



Note

ip -6 route shows the IPv6 routing table.

Hostname Resolution

The **hostname** command displays or temporarily modifies the system's fully-qualified hostname.

```
[root@demo ~]# hostname  
demo.example.com
```

The *stub resolver* is used to convert hostnames to IP addresses or the reverse. The contents of the file **/etc/hosts** is checked first.

```
[root@demo ~]# cat /etc/hosts
192.168.0.250    demo.example.com        demo      # Added by NetworkManager
127.0.0.1          localhost.localdomain  localhost
::1                demo.example.com        demo      localhost6.localdomain6 localhost6
```

If an entry is not found in that file the stub resolver looks for the information from a DNS nameserver. The **/etc/resolv.conf** file controls how this query is done:

- **nameserver**: the IP address of a nameserver to query. Up to three nameserver directives may be given to provide backups if one is down.
- **search**: a list of domain names to try with a short hostname. Both this and **domain** should not be set in the same file; if they are, the last instance wins. See **resolv.conf(5)** for details.

```
[root@demo ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 192.168.0.254
```

The **getent hosts hostname** command can be used to test hostname resolution.

ip addr is preferred over ipconfig

netstat -f

route -n

ip route

dig fgdn

nm-connection-editor is a GUI screen

service NetworkManager restart

or
service network restart

Modifying Network Configuration

NetworkManager may be installed on Red Hat Enterprise Linux 6. It consists of a core daemon, a GNOME Notification Area applet that provides network status information, and graphical configuration tools that can create, edit and remove connections and interfaces.

To change a NetworkManager-managed eth0 interface from using DHCP to using a static IP address:

1. Right-click the NetworkManager icon in the top Panel and select **Edit connections...**
2. On the **Wired** tab, select **System eth0** and click the **Edit...** button
3. Select the **IPv4 Settings** tab
4. On the **Method** drop-down menu, change **Automatic (DHCP)** to **Manual**
5. Under **Addresses** click **Add** and enter the IPv4 address, netmask (in VLSN or CIDR notation), gateway router, and DNS server to use
6. **IMPORTANT:** make sure that **Connect automatically** is checked so the interface starts at boot (rather than when the user logs in), and **Available to all users** is checked so that it is available system-wide
7. Click **Apply** to apply your changes.

It is also possible to configure the network by editing interface configuration files. Interface configuration files control the software interfaces for individual network devices. These files are usually named **/etc/sysconfig/network-scripts/ifcfg-<name>**, where <name> refers to the name of the device that the configuration file controls. The following are standard variables found in the file used for static or dynamic configuration.

| Static | DHCP | Any |
|--------------------------------------|-----------------------|---------------------------------|
| BOOTPROTO=static | BOOTPROTO=dhcp | DEVICE=eth0 |
| IPADDR=192.168.0.250 | | ONBOOT=yes |
| PREFIX=24 or 255.255.255.0 | | HWADDR=52:54:00:00:00:FA |
| GATEWAY=192.168.0.254 | | NM_CONTROLLED=yes |
| DNS1=192.168.0.254 | | |
| onboot = yes | | |

Network manager controlled instead of using manual control

Table1.1.Configuration Options for **ifcfg** file



Note

If **NetworkManager** is running, any changes made to the **ifcfg-*** files take affect immediately.



Note

If you need to configure static routes, the configuration is stored per interface in **/etc/sysconfig/network-scripts/route-<name>**. Details can be found in the Red Hat Enterprise Linux Deployment Guide, see below.



Important

NetworkManager runs by default on Red Hat Enterprise Linux 6 and may cause conflicts with your network configuration. If you want to permanently manage the network settings manually, add **NM_CONTROLLED=no** to the **ifcfg-*** file for each network interface.

/etc/sysconfig/network is used to specify the fully-qualified hostname and may specify a static default route if DHCP is not in use:

This is the global default

```
[root@demo ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=demo.example.com
GATEWAY=192.168.0.254
```

As we saw above, **/etc/resolv.conf** specifies the IP addresses of DNS servers and the search domain.



Important

If DHCP is in use, **/etc/resolv.conf** is automatically rewritten as interfaces are started unless you specify **PEERDNS=no** in the relevant interface configuration files.

If I am running my own DNS server

Network interfaces can be brought down with the **ifdown eth0** command and brought back up with the **ifup eth0** command, whether managed by NetworkManager or by unmanaged configuration files.

When changing the system configuration you must remember to:

1. Modify a configuration file
2. Restart a service
3. Verify the change

AMP Configuration - ed. for



References

Red Hat Enterprise Linux Deployment Guide

- Section 4.1: Network Configuration Files

Red Hat Enterprise Linux Deployment Guide

- Section 4.2: Interface Configuration Files

Red Hat Enterprise Linux Deployment Guide

- Section 4.4: Configuring Static Routes

Red Hat Enterprise Linux Deployment Guide

- Chapter 5: Network Configuration

/usr/share/doc/initscripts-*/sysconfig.txt

vim /etc/sysconfig/network-scripts/route-eth0
via interface
192.168.1.0/24 via 192.168.1.254
service network restart

cat /var/lib/dhcpclient/dhclient-eth0.leases
to dhc info on client side



Practice Quiz

Viewing and Modifying Network Configuration

1. Fill in the below table with the commands, utilities and filenames

| Setting Category | View Current Configuration | Change Configuration |
|----------------------------|-------------------------------|--|
| IP Address and Subnet Mask | <code>ip addr</code> | <code>NetworkManager</code> or <code>vi /etc/sysconfig/network-scripts/ifcfg-eth0</code> |
| Routing/Default Gateway | <code>ip route</code> | <code>vi /etc/sysconfig/network-scripts/ifcfg-eth0</code> or vi /etc/sysconfig/route-eth0 |
| System Hostname | <code>hostname</code> | <code>vi /etc/sysconfig/network</code> |
| Name Resolution | <code>/etc/resolv.conf</code> | <code>vi /etc/resolv.conf</code> |

Table 1.2. Network Configuration from the Command-Line

2. What would a dynamic `ifcfg-eth0` contain?

`bootproto=dhcp`

3. What would a static `ifcfg-eth0` contain?

`bootproto=none`
`ipaddr=`
`netmask=`
`gateway=`
`dns1=`

Basic Troubleshooting Process

Troubleshooting Steps

1. TEST

- Reproduce/verify/characterize the problem
- Monitor the issue
- Gather background information (hardware/software version, etc.)
- Gather diagnostic information (logs, error messages, etc.)
- Determine severity level

2. CHECK

- Look at configuration for evidence of a misconfiguration
- Compare expected settings to intended settings
- Verify proper operation and attachment of local hardware and external resources

3. FIX

- Modify and activate configuration
- Verify by re-running the TEST phase to ensure the problem has been resolved

4. DOCUMENT

- The final wrap-up step that should also be performed



Practice Exercise

Document Network Settings

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Before we break our network configuration, let us document the current settings on our serverX system.

1. Log in to serverX as root.
2. a. What is its current IP Address? 192.168.0.102
- b. What is its current CIDR subnet mask? /24
- c. What is its current default gateway? ip route
- d. What is its current hostname? hostname
- e. What are its current DNS servers? /etc/resolv.conf

/etc/nsswitch.conf
sets the order of using file or DNS first

can use mtr instead of traceroute

Network Troubleshooting Toolkit

The below table shows how to TEST, CHECK, and FIX each of the network troubleshooting categories:

| Category | TEST | CHECK | FIX |
|----------------------------|--------------------------------------|---|--|
| IP Address and Subnet Mask | <code>ping</code> , access a service | <code>ip addr</code> | <code>/etc/sysconfig/network-scripts/ifcfg-*</code> |
| Routing/Default Gateway | <code>traceroute</code> | <code>ip route</code> | <code>/etc/sysconfig/network-scripts/ifcfg-*</code> or provided via DHCP |
| Name Resolution | <code>host</code> | <code>/etc/hosts</code> and <code>/etc/resolv.conf</code> | <code>/etc/sysconfig/network-scripts/ifcfg-*</code> |

Table1.3.Network Troubleshooting from the Command-Line

Useful commands and files

- `ping`

```
[root@demo ~]# ping -c 2 instructor.example.com
PING instructor.example.com (192.168.0.254) 56(84) bytes of data.
64 bytes from instructor.example.com (192.168.0.254): icmp_seq=1 ttl=64 time=0.697 ms
64 bytes from instructor.example.com (192.168.0.254): icmp_seq=2 ttl=64 time=0.538 ms

--- instructor.example.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.538/0.617/0.697/0.083 ms
```

- `ip addr show eth0`

```
[root@demo ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:00:00:fa brd ff:ff:ff:ff:ff:ff
        inet 192.168.0.250/24 brd 192.168.0.255 scope global eth0
            inet6 fe80::5054:ff:fe00:fa/64 scope link
```

```
valid_lft forever preferred_lft forever
```

- **/etc/sysconfig/network-scripts/ifcfg-<name>**

```
[root@demo ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="dhcp"
HWADDR="52:54:00:00:00:FA"
NM_CONTROLLED="yes"
ONBOOT="yes"
```

- **traceroute**

```
[root@demo ~]# traceroute -Tn www.redhat.com
traceroute to www.redhat.com (184.85.80.112), 30 hops max, 60 byte packets
 1  192.168.0.254  0.641 ms  0.606 ms  0.590 ms
 2  172.31.35.1  9.829 ms  9.531 ms  9.237 ms
 3  204.60.4.40  27.954 ms  27.726 ms  27.385 ms
 4  66.159.184.226  27.128 ms  49.156 ms  48.291 ms
 5  151.164.92.147  43.256 ms  42.995 ms  42.155 ms
 6  12.122.81.57  60.897 ms  60.041 ms  54.531 ms
 7  75.149.230.169  54.143 ms  75.149.231.45  46.412 ms  192.205.37.34  40.208 ms
 8  68.86.86.45  67.587 ms  54.599 ms  53.381 ms
 9  68.86.86.234  65.540 ms  62.189 ms  53.777 ms
10  68.86.87.166  57.084 ms  55.752 ms  57.154 ms
11  184.85.80.112  55.707 ms  58.702 ms  57.996 ms
```

- **ip route**

```
[root@demo ~]# ip route
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.250 metric 1
default via 192.168.0.254 dev eth0 proto static
```

- **host**

```
[root@demo ~]# host i
i.example.com is an alias for instructor.example.com.
instructor.example.com has address 192.168.0.254
```

- **dig**

```
[root@demo ~]# dig i.example.com

; <>> DiG 9.7.0-P2-RedHat-9.7.0-5.P2.el6 <>> i.example.com
; global options: +cmd
; Got answer:
;_>>HEADER<- opcode: QUERY, status: NOERROR, id: 17644
; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;i.example.com.           IN      A

;; ANSWER SECTION:
i.example.com.      86400   IN      CNAME   instructor.example.com.
instructor.example.com. 86400   IN      A       192.168.0.254
```

```
;; AUTHORITY SECTION:  
example.com.          86400   IN      NS      instructor.example.com.  
  
;; Query time: 2 msec  
;; SERVER: 192.168.0.254#53(192.168.0.254)  
;; WHEN: Mon Dec 13 15:50:21 2010  
;; MSG SIZE  rcvd: 86
```

- **/etc/hosts**

```
[root@demo ~]# cat /etc/hosts  
192.168.0.250 demo.example.com demo # Added by NetworkManager  
127.0.0.1 localhost.localdomain localhost  
::1 demo.example.com demo localhost6.localdomain6 localhost6
```

- **/etc/resolv.conf**

```
[root@demo ~]# cat /etc/resolv.conf  
# Generated by NetworkManager  
domain example.com  
search example.com  
nameserver 192.168.0.254
```



References

- Red Hat Enterprise Linux Deployment Guide
 - Chapter 4: Network Interfaces

- Red Hat Enterprise Linux Deployment Guide
 - Chapter 5: Network Configuration



Test

Criterion Test

Exercise

Troubleshooting Network Configuration from the Command-line

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of the following should be performed on your virtual server, serverX. You will start by running a script that will "break" your network configuration. You will have ten minutes to resolve each of the three problems. Be sure to document what you have found, as we will review at the end.

1. Run the first script to misconfigure your networking:

lab-break-net 1

2. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
3. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
4. Document what you have found

Use this space for notes

DNS addr is wrong
use /etc/sysconfig/network-scripts/ifcfg-eth0
NetworkManager restart
service NetworkManager restart

5. Run the second script to misconfigure your networking:

lab-break-net 2

6. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
7. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
8. Document what you have found

problem w/ route
change gw in /etc/sysconfig/network-scripts/ifcfg-eth0

Use this space for notes

9. Run the third script to misconfigure your networking:

lab-break-net 3

10. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
11. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
12. Document what you have found

Use this space for notes

wrong IP
fix ifcfg-eth0
.restart service



Personal Notes



Unit Summary

Understanding Network Configuration Files

In this section you learned how to:

- Change network configuration with command-line tools
- Make network configuration changes persistent by editing files

Basic Troubleshooting Process

In this section you learned how to:

- Employ a systematic approach to troubleshooting system problems

Network Troubleshooting Toolkit

In this section you learned how to:

- Diagnose and correct network problems so that network access is restored



UNIT TWO

ADMINISTERING USERS AND GROUPS

Introduction

Topics covered in this unit:

- Manage Local Users/Groups
- Password Expiration
- LDAP client configuration
- Automounter metacharacters

Guru - system - config - user

Managing Local Users and Groups

A number of command-line tools can be used to manage local user and group accounts.

useradd Creates Users

- **useradd *username*** sets reasonable defaults for all fields in **/etc/passwd** when run without options
- **useradd** does not set any valid password by default, the user cannot log in.
- **useradd --help** will display the basic options that can be used to override the defaults

userdel Deletes Users

- **userdel *username*** removes the user from **/etc/passwd**, but, by default, leaves the home directory intact.
- **userdel -r *username*** removes the user and the user's home directory



Warning

When removing a user account with **userdel**, if you do not use the **-r** option to remove the user's home directory or if the user owns files outside their home directory on the system, you will end up with files owned by an unassigned user ID number.

This can lead to information leakage and other security issues.

This happens because, when a new user is created with **useradd**, unless the new user's UID number is specified (with the **-u *UID*** option), **useradd** will pick one. It will assign the "first free UID number" in its range and assign it to the new user.

If the "first free UID number" had been previously assigned to a user account which has since been removed from the system, the old user's UID number will get reassigned to the new user, and the new user ends up with ownership of the old user's remaining files! It is not hard to demonstrate this situation:

```
[root@serverX ~]# useradd prince
[root@serverX ~]# ls -l /home
drwx----- 4 prince prince 1024 Dec 23 12:30 prince
[root@serverX ~]# userdel prince
[root@serverX ~]# ls -l /home
drwx----- 4 500 500 1024 Dec 23 12:30 prince
[root@serverX ~]# useradd bob
[root@serverX ~]# ls -l /home
drwx----- 4 bob bob 1024 Dec 23 12:31 bob
drwx----- 4 bob bob 1024 Dec 23 12:30 prince
```

Notice that **bob** now owns all files that **prince** once owned. The best solution to this problem is to remove all "unowned" files from the system when the user that created them is deleted.

id Displays User Information

- **id** will display user information for the current process, including the user's UID number and group memberships.
- **id username** will display user information for *username*, including the user's UID number and group memberships.

passwd Sets Passwords

- **passwd username** can be used to either set the user's initial password or change that user's password.

Use this space for notes

echo password | passwd --stdin prince

About Local Users and Groups

UID Ranges

- *UID 0* is **root** and has special privileges
- *UID 1-499* are "system users" by convention - generally non-interactive service accounts
- *UID 500+* are "regular users" for people to use for interactive access to the machine

Primary Groups

- Every user has exactly one *primary group*.
- For local users, the primary group is defined by the GID number of the group listed in the third field of **/etc/passwd**.
- Normally, the primary group owns new files which are created by the user.
- Normally, the primary group of a newly created user is a newly created group with the same name as the user. The user is the only member of this *User Private Group* (UPG).

Supplementary Groups

- Users may be a member of zero or more *supplementary groups*.
- The users that are supplementary members of local groups are listed in the last field of the group's entry in **/etc/group**: For local groups, user membership is determined by a comma-separated list of users found in the last field of the group's entry in **/etc/group**:

groupname:password:GID:list,of,users,in,this,group

- Supplementary group membership is used to help ensure that users have access permissions to files and other resources on the system.

groupadd -r music
add music to uid <500

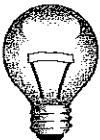


Note

Given the automatic creation of User Private Groups (GID 500+), it is generally recommended to set aside a range of GID numbers to be used for Supplementary Groups. Here we will use the range of 200 to 400, though this does risk a collision with a System Group (GID 0-499).

Managing Supplementary Groups

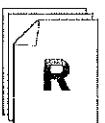
1. **groupadd -g 201 groupname** to create a supplementary group named **groupname** with a GID of **201**.
2. **usermod -aG groupname username** will add the user **username** to the group **groupname**.



Important

The use of the **-a** option makes **usermod** function in "append" mode. Without it, the user would be removed from *all other* supplementary groups.

*logs out
logs in to get the new assignment*



References

Red Hat Enterprise Linux Deployment Guide

- Chapter 15: Users and Groups

useradd(8), usermod(8), userdel(8), groupadd(8), groupdel(8) man pages



Practice Exercise

Create Users Using Command-Line Tools

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Create a number of users on your serverX system, setting an initial password (recording in the blanks below).

For the users created, put them into groups as listed here.

| Group | groupname | user_list |
|-------------------|------------------|---------------------------------------|
| Professors | <i>profs</i> | <i>faraday, juliet, elvis</i> |
| Graduate Students | <i>grads</i> | <i>jack, kate, james, elvis</i> |
| Summer Interns | <i>interns</i> | <i>walt, ben, claire, hugo, elvis</i> |

Table 2.1. User and Group Assignments

Notice that there is one additional user that you will need to create named **elvis**, who should be placed in all three groups.

1. Log into serverX as *root*.
2. Add the user *juliet*.
3. Confirm that *juliet* has been added using the **id** command.
4. Confirm that *juliet* has been added by examining the **/etc/passwd** file.
5. Use the **passwd** command to initialize *juliet*'s password and write down the password here:

password

6. Continue adding the remaining users from the list below, remembering to set an initial password and writing them down next to each username:

- faraday _____
- jack _____
- kate _____
- james _____
- walt _____
- ben _____
- claire _____
- hugo _____
- elvis _____

7. Create the groups specified in the table above, and assign the appropriate group members.

Managing Passwords

Historically, passwords were stored in **/etc/passwd**, but that file must be world readable to support username to UID mappings needed by utilities like **ls** to display the username rather than the UID number.

Passwords were migrated to a more secure **/etc/shadow** file where several different password encryption algorithms are supported. As long as encrypted passwords are being stored in a dedicated file, password aging policy and data can be stored as well.

What 3 pieces of information are stored in a password hash?

\$1\$gCjLa2/Z\$6Pu0EK0AzfCjxjv2hoLOB/

1. **1** - The hashing algorithm (1 indicates MD5 hash)
2. **gCjLa2/Z** - The salt used to encrypt the hash
3. **6Pu0EK0AzfCjxjv2hoLOB/** - The encrypted hash



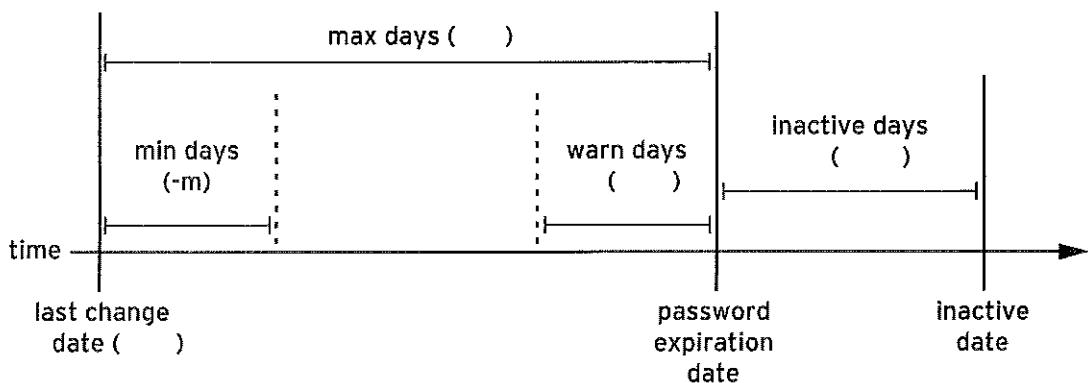
Note

Red Hat Enterprise Linux 6 supports two new strong password hashing algorithms, SHA-256 (algorithm **5**), and SHA-512 (algorithm **6**). These may be enabled as the default for **/etc/shadow** using **system-config-authentication** to select it from the **Password Hashing Algorithm** drop-down menu on the **Advanced Options** tab.

/etc/shadow Fields

1. Username
2. Password hash
3. Date of last password change (number of days since 1970.01.01)
4. Minimum password age (in days, 0 = no minimum age requirement)
5. Maximum password age (in days)
6. Password warning period (in days, 0 = no warning given)
7. Password inactive period (in days)
8. Account expiration (number of days since 1970.01.01)

The following diagram relates the relevant password aging parameters which can be adjusted using **chage** to implement a password aging policy.



As your instructor discusses these parameters, fill in the parenthesis in the above diagram with the relevant (short) **chage** command line switch.

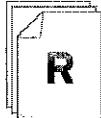
As an example, **-m** has been added to the *min days* parameter to get you started.

```
# chage -m 0 -M 90 -W 7 -I 14 username
```

chage -d 0 username will force a password update on next login.

chage -l username will list a username's current settings.

usermod can modify an account, including "locking" with the **-L** option.



References

Red Hat Enterprise Linux Deployment Guide

- Section 15.6: Shadow Passwords

chage(1), **shadow(5)**, **crypt(3)** man pages

vim /etc/login.defs
PASS_MAX_DAYS



Practice Quiz

Account Maintenance

1. What command would lock **elvis**'s account?
2. What command would then unlock it?
3. What command would cause **elvis**'s account to expire on March 15th, 2012?

usermod -L elvis

usermod -U elvis

chage -E 3/15/2012 elvis

Network Authentication Using an LDAP Server

So far in this class, we have looked at local user accounts managed through local files on each machine, **/etc/passwd**. But it is difficult to coordinate local user accounts to be the same on many systems.

In this section, we will look at how to set up a machine as a client, to use network user accounts that are provided by an existing LDAP directory service. This allows the LDAP directory to be our central authority for all network users and groups in our organization.

User account information determines the characteristics and configuration of the account. *Authentication methods* are used to determine if someone trying to log in should get access to the account. *Network directory services* can provide both user account information and authentication methods.

LDAP directory servers can be used as a distributed, centralized, network user management service. Directory entries are arranged in a tree structure that can be searched. The *base DN (Distinguished Name)* is the base of the tree that will be searched for directory entries for users and groups.

Key Elements for LDAP Client Configuration

1. Server's fully-qualified hostname
2. Base DN to search for user definitions
3. The certificate authority ("CA") certificate used to sign the LDAP server's SSL certificate

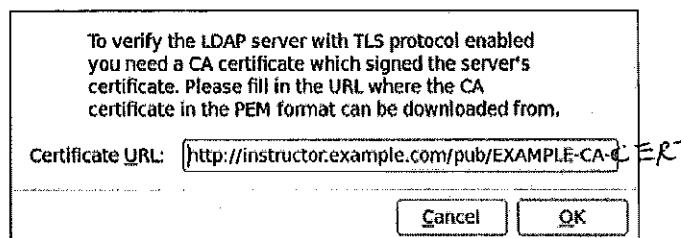
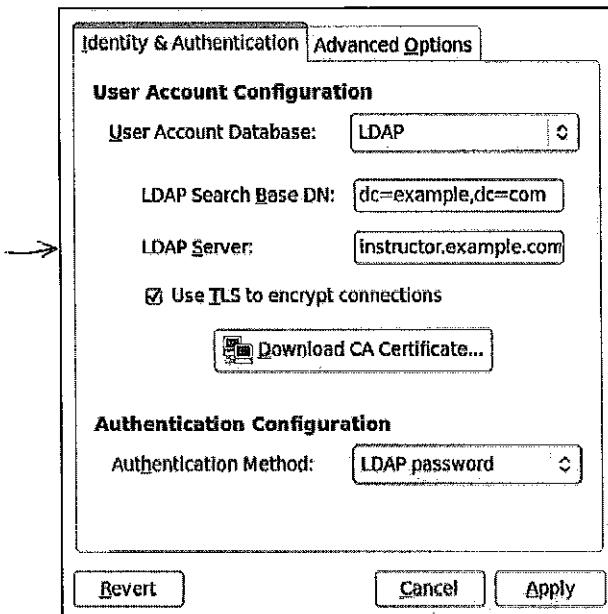
Use this space for notes

You should ensure that the **directory-client** yum package group is installed, which includes the packages **sssd**, **authconfig-gtk**, and **oddjob-mkhomedir**, before you begin.

System → Administration → Authentication or **system-config-authentication** can be used to modify the configuration of *Identity & Authentication*.

yum groupinstall directory-client

DN
DC=example, DC=com



system-config-authentication will automatically turn on the **sssd** service which will look up and cache LDAP user information and authentication credentials for the client. If the LDAP server is unavailable but **sssd** is working, the system may be able to authenticate and get information about network users from the **sssd** cache.

Run user 2

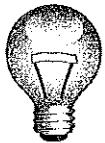
Use **getent passwd username** to verify the account information being used. This works whether the user is a local user defined in **/etc/passwd** or a network user from an LDAP service. The command will always show the definition that is actually being used by the system if there is any duplication between local users and network users. By default, the local user definition overrides the network user definition.



Note

In Red Hat Enterprise Linux 6, **getent passwd** (without specifying a username) will only dump out local usernames by default, it will not dump out the list of all LDAP users as it did in Red Hat Enterprise Linux 5. This is done for performance reasons; see **sssd.conf(5)** under the **enumerate** option for details. (This behavior can be changed by setting **enumerate = True** in the **[domain/default]** section of **/etc/sssd/sssd.conf**.)

*authconfig - textbased version of
system-config-authentication*
*authconfig-tui - Ncurses based
d/cert to /etc/openldap/cacerts*



Important

When using LDAP password as your authentication method, you *must* select and configure **Use TLS to encrypt connections**. This is to prevent clear-text passwords from being sent to the LDAP server over the network for authentication.

This is a change from Red Hat Enterprise Linux 5, which would allow the insecure use of LDAP password authentication without TLS. In RHEL 6, you may still use LDAP without TLS if you are using LDAP to get user information only. (For example, you may be using Kerberos for password authentication.) It is better practice to always use TLS.



References

Red Hat Enterprise Linux Deployment Guide

- Chapter 8: Authentication Configuration

system-config-authentication(8), **sssd(8)**, and **sssd.conf(5)** man pages



Practice Quiz

LDAP Client Configuration

1. What seven pieces of information are typically provided by *User account information services?*
account, uid, GID, GECOS, home, shell
password, same as getent
2. What "other" type of information can be provided by a *network directory service?*
authentication
3. What are the three pieces of information a client machine needs to be configured to get user information from an LDAP directory service?
DN, FQDN LDAP, URL of CA cert
4. What does the command `getent passwd ldapuser1` do? Why is this useful?
lookup nsswitch.conf
connect to LDAP

Network Mounting Home Directories

Recall that mounting network shares requires four pieces of information: hostname, sharename, mountpoint and mount options.

1. Use **showmount -e nfsserver.domain** to get the exported path that when combined with the hostname gives us the *sharename*.

```
[root@serverX ~]# showmount -e instructor.example.com
Export list for instructor.example.com:
/home/guests 192.168.0.0/255.255.255.0
/var/nfs     192.168.0.0/255.255.255.0
/kickstart   192.168.0.0/255.255.255.0
/var/ftp/pub 192.168.0.0/255.255.255.0
```

2. Use **getent passwd username** to get the needed home directory *mountpoint*.

```
[root@serverX ~]# getent passwd ldapuser1
ldapuser1:*:1701:1701:LDAP Test User 1:/home/guests/ldapuser1:/bin/bash
```



Note

getent passwd only shows the local accounts by default. You can display a single account if you explicitly add the name (as above). If you want to show all of the available accounts (including LDAP accounts), add the following to **/etc/sssd/sssd.conf** beneath the **[domain/default]** section:

```
enumerate = True
```

Restart the **sssd** service:

```
[root@serverX ~]# service sssd restart
[root@serverX ~]# getent passwd
...
ldapuser10:*:1710:1710:LDAP Test User 10:/home/guests/ldapuser10:/bin/bash
ldapuser11:*:1711:1711:LDAP Test User 11:/home/guests/ldapuser11:/bin/bash
...
```

3. As home directories, we probably want to use **rw** as the *mount option*.

Configuring indirect maps in **autofs** would look something like this:

```
# cat /etc/auto.master
/home/guests    /etc/auto.guests
# cat /etc/auto.guests
ldapuser1  -rw  instructor.example.com:/home/guests/ldapuser1
ldapuser2  -rw  instructor.example.com:/home/guests/ldapuser2
ldapuser3  -rw  instructor.example.com:/home/guests/ldapuser3
ldapuser4  -rw  instructor.example.com:/home/guests/ldapuser4
```

Each time a new LDAP user is created, **/etc/auto.guests** would need to be updated to include that additional user. However, notice the "pattern" to the lines. We want to support logging in as **any** username, so we could replace the first column with an "asterisk (*)", a wildcard, matching any subdirectory name that the login process may try to **cd** to. Then, we use the metacharacter, "ampersand (&)", to replace the username in the share which carries over the mapname matched by the wildcard:

```
# cat /etc/auto.master  
/home/guests /etc/auto.guests  
# cat /etc/auto.guests  
* -rw instructor.example.com:/home/guests/&  
    &vols=3
```



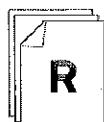
Important

At the time of writing, there is a bug in the **autofs** init script (bugzilla #624444) that causes **service autofs restart** to sometimes fail. Until this is fixed, use

service autofs reload

or

service autofs stop && service autofs start.



References

Red Hat Enterprise Linux Storage Administration Guide

- Section 10.3: **autofs**

autofs(5), auto.master(5) man pages

ndr



Practice Case Study

Automounting NFS Directories

These steps should be performed on serverX.

Your company is now taking on several new clients:

1. The Organization of Secret Hidden Undertakings (or OSHU)
2. Race Along the Lake Investments, Inc. (or RALII)

Your company has a new NFS server with shares for storing files related to these "special" clients named **instructor.example.com**, with currently two shares: **/var/nfs/oshu** and **/var/nfs/ralii**, with the expectation that more will be added as new clients are signed.

The workstations need to use **autofs** to automatically mount these shares to: **/special/oshu** and **/special/ralii**, respectively with read-only permissions.

Given the expectation that additional clients will be signed shortly, implement this using **autofs** wildcards and metacharacters.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Test

Criterion Test

Exercise

Get Network User Information from an LDAP Directory Service

Carefully perform the following steps. Ask your instructor if you have problems or questions.

You will now configure serverX to get information about network users from an LDAP directory server available to all machines in the classroom.

Here is information that was provided to you about the LDAP server:

- Hostname: *instructor.example.com*
- Search Base DN: *dc=example,dc=com*
- CA Certificate: *http://instructor.example.com/pub/EXAMPLE-CA-CERT*¹

You will then configure serverX to automatically mount the home directories of your LDAP-based network users when they log in.

Here is information which was provided to you about the NFS storage server that contains the home directories:

- Hostname: *instructor.example.com*
 - Exported Directory: **/home/guests/**
1. Login to serverX as *root*. If you use **ssh**, include the **-X** option to forward graphical interfaces to your workstation.
 2. Use **system-config-authentication** to configure serverX to get network user information from the classroom LDAP server.
 3. Use the **id** command to confirm that the user **ldapuserX** (replace X with your station number) is defined on the system.
 4. Use the **getent** command to look up the user information for **ldapuser1**.
 5. Login to serverX as **ldapuserX** or any of the other network users, using the password "password". (Do not worry if you get an error on login about a non-existent home directory, that is expected.)
 6. Modify the configuration of the automounter on serverX so that if any directory **ldapuserX** is accessed in /home/guests, the automounter will attempt to mount the equivalent NFS exported directory from **instructor.example.com:/home/guests/ldapuserX** (Hint: Use wildcard syntax.)

7. Log into serverX as ldapuserX, where "X" is your station number, with the password "password". The user's home directory should be automatically mounted.

```
[student@desktop1 ~]$ ssh ldapuser1@server1
The authenticity of host 'server1 (192.168.0.101)' can't be established.
RSA key fingerprint is 33:fa:a1:3c:98:30:ff:f6:d4:99:00:4e:7f:84:3e:c3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1,192.168.0.101' (RSA) to the list of known hosts.
ldapuser1@server1's password: password
Last login: Thu Dec 16 14:59:49 2010 from instructor.example.com
[ldapuser1@server1 ~]$ pwd
/home/guests/ldapuser1
[ldapuser1@server1 ~]$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
...
instructor.example.com:/home/guests/ldapuser1
      1032192     36864    943104    4% /home/guests/ldapuser1
```



Personal Notes



Unit Summary

Managing Local Users and Groups

In this section you learned how to:

- Add and remove user accounts from the command-line
- Set and/or change user passwords
- Identify primary versus supplementary groups
- Create a new group
- Delete a group
- Add a user account to a group

Managing Passwords

In this section you learned how to:

- Customize password aging policies for the users to meet organizational security requirements

Network Authentication Using an LDAP Server

In this section you learned how to:

- Configure the system to authenticate users managed in a central LDAP directory service

Network Mounting Home Directories

In this section you learned how to:

- Automount existing NFS home directories for remote users using indirect map metacharacters



UNIT THREE

COMMAND-LINE PROCESS MANAGEMENT

Introduction

Topics covered in this unit:

- Launching graphical commands as root
- Monitoring processes
- Terminating processes
- Schedule periodic tasks

Launching Graphical Tools from Bash

It may seem bizarre to run a graphical command from the command line, but you will understand the usefulness when we discuss connecting to remote systems.

How to run graphical commands as **root**:

1. Open a terminal window
2. Use **su -** to become root (you must use the **-**)
3. Run **command &**

Example:

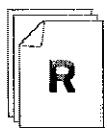
```
[student@serverX ~]$ su -
Password: redhat
[root@serverX ~]# nautilus &          nautilus --browser &
```

Job control functions provided by Bash:

- **Ctrl+c** (often written ^C): Terminate the foreground process
- **Ctrl+z** (often written ^Z): Suspend the foreground process
- **jobs**: List backgrounded and stopped processes
- **fg**: Send a job to the foreground. Only one process can run in the foreground in a shell. If no argument is given, it will foreground the current job (shown with a + in **jobs** output). Pass **fg** the job ID to manage jobs other than the current job.

```
[student@serverX ~]$ sleep 3000 &
[1] 22252
[student@serverX ~]$ sleep 4000 &
[2] 22253
[student@serverX ~]$ sleep 5000 &
[3] 22254
[student@serverX ~]$ jobs
[1]  Running                  sleep 3000 &
[2]- Running                  sleep 4000 &
[3]+ Running                  sleep 5000 &
[student@serverX ~]$ fg
sleep 5000
Ctrl+c
[student@serverX ~]$ jobs
[1]- Running                  sleep 3000 &
[2]+ Running                  sleep 4000 &
[student@serverX ~]$ fg 1
sleep 3000
```

- **bg**: Send a job to the background. Many jobs can run in the background in a single shell. If no argument is given, **bg** will background the current job (as with **fg**) as though it had been started with **&**.



References

GNU Bash Reference Manual

- Chapter 7: Job Control

<http://www.gnu.org/software/bash/manual>

If **bash-doc** is installed from the **Optional** RHN channel:
file:///usr/share/doc/bash-4.1.2/doc/bashref.pdf

jobs(1), **fg(1)** and **bg(1)** man pages



Practice Performance Checklist

Launching Graphical Tools from Bash

- Log in to your serverX host graphically as **student**.
- Open a terminal window.
- Within the window switch to a **root** shell.
- Launch **nautilus** in the foreground from the command line.
- Use the keyboard shortcut to get your shell prompt back without terminating the process.
- Put **nautilus** in the background.
- List your current shell jobs.
- Exit the root shell.

Monitoring Processes

A process is an instance of a running program. The **ps** command can be used to list processes. By default, it gives you very little useful information. It is only showing us processes started from this terminal (and in an X session, every window is a terminal...). If asked for help, however, the **ps** command has more command line switches than you want to know about, and can be tailored to provide very concise information. Everyone has their favorite options, but here is one we suggest you use:

```
# ps aux
USER     PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1  0.0  0.0   4132   888 ?        Ss   17:44  0:01 /sbin/init
root      2  0.0  0.0     0    0 ?        S    17:44  0:00 [kthreadd]
root    1469  0.0  0.0  63572  1232 ?        Ss   17:45  0:00 /usr/sbin/sshd
rlocke  2348  0.0  0.0 107956  1852 pts/0    Ss   17:46  0:00 bash
rlocke  2711  0.0  0.1 150652  3628 pts/1    S+   17:54  0:00 vim Processes.xml
rlocke  3507  2.0  0.0 107564  1032 pts/0    R+   19:00  0:00 ps aux
...
```

The **top** utility displays an automatically updating list of current processes. The following keys can be used within **top** to manage the display:

- M: Sort processes by memory utilization
- P: Sort processes by processor utilization
- h: Display help about more commands
- q: Quit

When ordering processes by memory use, **top** considers **RES** more important than **VIRT**. **RES** is the amount of physical memory currently used by a process (the “resident set”). **VIRT** is the space in its virtual memory map currently reserved by the process for possible use, and is much less meaningful. Even **RES** can be an over-estimate, as it includes memory shared with other processes. In **ps**, **VIRT** is labeled **VSZ**, and **RES** is **RSS**.



References

ps(1) and **top(1)** man pages

Terminating and Governing Processes

Processes communicate using messages called signals. Signals arrive at any time (asynchronously). Signals do not carry information beyond their "signal number" -- what kind of signal it is. Process can react ("handle the signal") in different ways depending on the signal number: it can exit, exit and dump a memory image, ignore the signal, or do something else. Most often, when a user wants to signal a process, they want it to terminate (exit). System events can send signals, or a user can send an arbitrary signal to a process with the `kill` command.

Signals

1. `top` and `kill` can both be used to send a signal to a process.
2. `kill -l`: Displays a table of the defined signal numbers.
3. `kill -9 3254`: Sends signal number 9 to the process with PID 3254.

K:\all pk

| Number | Name | Function |
|--------|------|--|
| 1 | HUP | Reinitialize a daemon |
| 9 | KILL | Force a process to terminate immediately |
| 15 | TERM | Request a process to terminate after cleanup (DEFAULT) |

Table 3.1. Good Signals to Know

In the table above, circle the signal that a programmer cannot override.

Process Scheduling (Niceness)

- A Linux system can have as many processes running at the same time as it has CPU cores.
- But a system appears to have more processes running by making them take turns at running on available cores ("time slicing").
- By default, every process has equal access to CPU time.
- The niceness of a process can be changed to adjust the priority of the process, giving it a larger or smaller share of CPU time compared to other processes.
- Niceness is a value which ranges from -20 (very greedy) through a default of 0 to 19 (very nice to other processes). *-20 ————— 19*
- Users can increase the niceness of their processes (requesting a smaller share of time).
- Only root can decrease the niceness of processes (requesting a larger share of time).
- `renice` and `top` can both be used to change the niceness of a running process.
- `nice` can be used to set the niceness of a new process.
- Niceness only affects the share of CPU time a particular process gets relative to other processes; if other processes are not running, a low priority process will still get all the CPU time on one core.



Workshop

Using top to Manage Processes

Follow along with the instructor as you complete the steps below.

1. Open two terminal windows from a graphical session on serverX.
2. In the first terminal, run **top**.
3. In the second terminal, run:

```
cat /dev/zero > /dev/null &
```

4. Start four (4) more **cat** processes as above.
5. In **top**, note that the **cat** processes each have a roughly equal share of CPU time.
6. Using **top**, determine the PID of one **cat** process.

7. Use **renice** in the second terminal to adjust the niceness of the process to **10**:

```
renice -n 10 PID
```

8. Choose a different **cat** process, and make it greedy by using renice to change its priority to **-5**. Again observe the relative CPU utilization.



Note

You will need to have **root** privileges to give it a higher priority (lower nice value).

9. Use **nice** to start a new **cat** process with a lower priority:

```
nice -n 5 cat /dev/zero > /dev/null &
```

10. Use the **r** key in **top** to renice some of the **cat** processes.
11. Use the **k** key in **top** to kill all of the **cat** processes.



References

kill(1), **signal(7)**, **nice(1)**, **renice(1)** man pages



Practice Exercise

Managing Processes

Before you begin...

On serverX run the **lab-setup-processes** command.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Change the priority of the process that is using the most CPU resources to 5.
2. Terminate the process that is using the most memory resources.
3. Execute **lab-grade-processes** on serverX to confirm you identified and managed the correct processes.
4. Run the **lab-cleanup-processes** command to clean up.

Managing Periodic Tasks

cron manages programs that must be run on a repeating, periodic schedule. A daemon, **crond**, wakes up once a minute to run any tasks that have been scheduled. Users schedule personal tasks with the **crontab** command. The system administrator can set tasks in system-wide configuration files.

Individual users register tasks using a text file referred to as a crontab ("cron table") file. **crontab -l** lists the file. **crontab -r** deletes the file. **crontab -e** edits the file. The default editor used with **crontab -e** is **vi**.

crontab -e will open a blank crontab file. Syntax documented in **crontab(5)** man page. Each line defines either a scheduled task or an "environment variable" that affects task execution. Task lines have six fields: the first five fields define the minute, hour, day of month, month, and day of week, while the remainder of the line specifies a command to run.



Important

The command will run when *either* day of month *or* day of week match! (Not just when both criteria match, which is the behavior most people expect.)

The following example shows a cron entry that runs the **ls** command every minute:

* 1 * 2 * 3 * 4 * 5 * 6 ls

man 5 crontab

- ➊ Minute. Valid values are 0-59
- ➋ Hour. Valid values are 0-23. 20 means 8 PM.
- ➌ Day of month. Valid values are 1-31
- ➍ Month. Valid values are 1-12 or the first three letters of the name (e.g., Jan)
- ➎ Day of week. Valid values are 0-7 (0 or 7 is Sunday) or the first three letters of the name (e.g., Sun)
- ➏ Command. Add the command as you would type it on the command-line. Use the semi-colon (;) to separate multiple commands to run at the same time. Add a separate line to run a command at a different time.

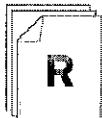
| Cron syntax | When the command will run |
|---------------|---------------------------------------|
| 05 * * * * | Every hour at 5 minutes past the hour |
| 05 02 * * * | Every day at 2:05 AM |
| 30 08 01 * * | 8:30pm on the first of every month |
| 00 07 25 12 * | December 25th at 7:00 AM |
| 30 16 * * 5 | Every Friday at 4:30pm |
| */5 * * * * | Every 5 minutes (0,5,10...45,50,55) |

| Cron syntax | When the command will run |
|---------------------------------|---|
| <code>*/10 9-16 1,15 * *</code> | Every 10 minutes between 9am and 5pm (it will not run at 5pm--the last instance will run at 4:50pm) on the first and fifteenth of the month |
| <code>0 0 1 jan 0</code> | January 1 at midnight <i>and</i> every Sunday in January (not just if January 1 is on Sunday) |

Table 3.2. Example cron table entries

Tasks that belong to the system normally go in system crontab files instead of personal ones. The main crontab file is `/etc/crontab` which can be edited normally (`crontab -e` is not used). A better practice is to create a crontab file in a normal text editor and drop it into `/etc/cron.d/` (this avoids issues when the `cronie` package is updated; this is what system packages do). `/etc/cron.d/` crontab files have an extra field after the date specification indicating the user which will run the job.

Scripts which need to run once a day, week, or month can simply be set executable and dropped into the appropriate directory in `/etc/cron.{daily,weekly,monthly}`. These tasks are run by the system `anacron` service, configured by `/etc/anacrontab`. In Red Hat Enterprise Linux 6, `anacron` is an integrated component of cron to better manage these jobs and ensure that they are run after boot if missed because the machine is off. For more information see `anacrontab(5)` man page.



References

`crontab(1)`, `crontab(5)`, `anacrontab(5)`, and `crontabs(4)` man pages



Practice Quiz

cronie Scheduling

1. When will the following jobs run?

- a. 00 07 25 12 * /usr/local/bin/open_presents
- b. */5 * * * * /usr/local/bin/take_stats
- c. 07 03 * * * /sbin/service xend restart
- d. 30 16 * * 5 /usr/local/bin/mail_checks

12/25 @ 7:00am

every 5 min

every day @ 3:07

every Friday @ 16:30

2. Devise a cron entry which would run the script **/usr/local/bin/vacuum_db** once a month on the first day of the month.

0 0 1 * * /usr/local/bin/vacuum_db

3. What if the machine in the previous question was down for maintenance on February first? What would be a better way to insure the database doesn't operate 2 (or more) months between vacuuming?

set it up in an acron monthly schedule



Test

Criterion Test

The following are to be performed on your serverX machine.

Exercise

Managing Processes

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. On your serverX, execute the script `manage_processes_start`.
2. The server station should now be very sluggish.
3. Determine the process which is using excessive amounts of memory, and terminate the process.
4. Determine the processes which are using excessive amounts of CPU, and renice them to a niceness of 15.
5. Create a system cron job which once every half hour readjusts all processes owned by the user `elvis` to a niceness of 10.
may need to use absolute path



Personal Notes



Unit Summary

Launching Graphical Tools from Bash

In this section you learned how to:

- Temporarily switch to another account from Bash without logging out
- Describe the role and privileges of the root user
- Launch graphical commands from Bash as root

Monitoring Processes

In this section you learned how to:

- Identify processes which consume the most CPU resources

Terminating and Governing Processes

In this section you learned how to:

- Terminate processes
- Change the priority of an existing process
- Launch a process with a non-default priority

Managing Periodic Tasks

In this section you learned how to:

- Schedule recurring jobs using **cron**



UNIT FOUR

GET HELP FROM RED HAT

Introduction

Topics covered in this unit:

- On-line Red Hat documentation
- Contacting Red Hat

Research On-line Documentation

An installed Red Hat Enterprise Linux system has documentation to go with the software you have installed. Traditionally, the **man** and **info** commands access information about commands, file formats and devices that are useful from the command line.

Other documentation is stored in the **/usr/share/doc** directory. Look for the name of a software package that interests you as a subdirectory (folder) within the **/usr/share/doc** directory to find documentation related to that package.

If you can't find what you are looking for on the Red Hat Enterprise Linux system, there are plenty of online resources, described below, to help you.

1. Red Hat Customer Portal

The Red Hat Customer Portal (<http://access.redhat.com/>) consolidates information you need to manage your Red Hat Enterprise Linux system. The site tells how to manage subscriptions and get telephone and online support. Full site access is provided through the same Red Hat login you may already use for hosted Red Hat Network access, and is one of the benefits of your software subscription.

On the **Knowledge** tab, you can select the **Knowledgebase** item to search the Red Hat Knowledgebase to find articles on hardware and software questions and issues.

Portions of the Red Hat Customer Portal are accessible without a current Red Hat login.

2. Documentation

Red Hat produces its own set of manuals to go with each release of Red Hat Enterprise Linux. Those manuals are available from <http://access.redhat.com/docs>. Manuals include:

- Release Notes - Describes features and issues associated with the Red Hat Enterprise Linux release.
- Migration Planning Guide - Describes how to migrate from earlier Red Hat Enterprise Linux releases.
- Installation Guide - Covers installation issues, including how to do bare-metal and unattended installs.
- Deployment Guide - Includes topics for configuring your Red Hat Enterprise Linux systems.

Other online guides cover topics such as security, resource management, clustering and virtualization.

3. Google search

Using Google searches for Red Hat Enterprise Linux, along with search terms representing a specific issue you want to learn more about, can often yield the best results. If you are running into an issue with some hardware or software component, that someone else has had an issue as well, a Google search (<http://www.google.com>) will often turn up good results. You can limit your search to a specific site by appending

site:web_site, where **web_site** is the web site to which to limit the search. For example, **site:www.redhat.com/docs** would limit the Google search to the Red Hat documentation site.



References

Red Hat Customer Portal

<http://access.redhat.com/>

Red Hat documentation

<http://access.redhat.com/docs/>

<http://www.redhat.com/docs/>

Knowledgebase

<http://access.redhat.com/kb/knowledgebase/>

Linux-related searches

<http://www.google.com/linux/>

To limit your Google search to the Red Hat documentation site, enter your search terms followed by **site:www.redhat.com/docs**

Getting the Most from Red Hat Global Support Services

This section is designed to help you approach Red Hat technical support.

There are many ways of getting help with Red Hat Enterprise Linux. As a subscriber to Red Hat Enterprise Linux, one of your ultimate resources is official Red Hat technical support. To get the best result when you contact Red Hat Global Support Services, you should understand a bit about how the process works and what information you need to bring with you.

The following are the seven steps to take when interacting with Red Hat Global Support Services (GSS):

1. *Define the problem*

Articulate the issue. Reproduce the problem and list the steps taken to do so.

2. *Search documentation and kbase articles*

Do your homework. Search for documentation from others who have dealt with similar issues.

3. *Gather background information*

What versions of relevant software are you running? Are there updates that may fix your issue? What error messages and symptoms are produced? Be as specific as possible.

4. *Gather relevant diagnostic information (**sosreport**)*

Gather log files, core dumps, output from commands and anything else you may think is relevant. Run **sosreport** to gather information.

5. *Determine the severity level*

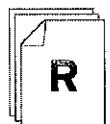
Try to accurately determine the severity level. The severity levels (1-4) are described at https://access.redhat.com/support/policy/GSS_severity.html

6. *Contact Red Hat via phone or web*

Make sure you have your Red Hat account number available. Contact information is available from https://access.redhat.com/support/policy/support_process.html by clicking on the *submit a support case online or contact us by phone* link.

7. *Ensure the support ticket is transferred to a technician in your region*

After you contact Red Hat, ask that your support ticket be transferred to a technician in your geographic region to avoid time zone issues.



References

Overview of Red Hat support process

<https://www.redhat.com/support/process/>

Netiquette

<http://www.catb.org/~esr/faqs/smart-questions.html>

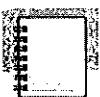


Practice Resequencing Exercise

Working with Red Hat Global Support Services

Below are the steps taken when interacting with Red Hat Global Support Services. Mark the order the steps should be taken:

- Gather relevant diagnostic info (log information, core dumps, etc.)
- Have support ticket be transferred to a technician in your region
- Define the problem
- Contact Red Hat via phone or web
- Determine the severity level
- Gather background information
- Search documentation and kbase articles



Personal Notes



Unit Summary

Research On-line Documentation

In this section you learned how to:

- Use Red Hat on-line documentation to answer questions

Getting the Most from Red Hat Global Support Services

In this section you learned how to:

- Describe the process for getting additional help from Red Hat Global Support Services



UNIT FIVE

MANAGE SYSTEM RESOURCES

Introduction

Topics covered in this unit:

- System Log Destinations
- Log Summary Reports
- Redirect Log Reports

Determine Log Destinations

Many programs use a standard protocol to send messages to `rsyslogd`. Each message is described by a *facility* (the type of message it is) and a *severity* (how important it is). The names of available facilities and levels of severity are standardized. The `/etc/rsyslog.conf` file uses the facility and severity of a log message to determine where it should be sent (to a log file, for example).

The `rsyslog.conf` file is documented by the `rsyslog.conf(5)` man page and by extensive HTML documentation in `/usr/share/doc/rsyslog-*/manual.html`. The ##### RULES ##### section of `/etc/rsyslog.conf` contains directives that define where log messages are saved. The left-hand side of each line indicates the facility and severity of the log message the directive matches. The right-hand side of each line indicates what file to save the log message in. Note that log messages are normally saved in files in the `/var/log` directory.

Sample `rsyslog.conf` file

```
##### MODULES #####
$ModLoad imuxsock.so      # provides support for local system logging (e.g. via logger
                           command)
$ModLoad imklog.so        # provides kernel logging support (previously done by rklogd)
#$ModLoad immark.so       # provides --MARK-- message capability

# Provides UDP syslog reception
#$ModLoad imudp.so
#$UDPServerRun 514

# Provides TCP syslog reception
#$ModLoad imtcp.so
#$InputTCPServerRun 514

##### GLOBAL DIRECTIVES #####
# Use default timestamp format
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

##### RULES #####
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                     /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none      /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                    /var/log/secure
                                                writes asynchronously

# Log all the mail messages in one place.
mail.*                                         -/var/log/maillog

# Log cron stuff
cron.*                                         /var/log/cron
```

```
# Everybody gets emergency messages
*.emerg

# Save news errors of level crit and higher in a special file.
uucp.news.crit                                /var/log/spooler

# Save boot messages also to boot.log
local7.*                                       /var/log/boot.log
```

(*) Censuke

Default Log Files

Fill in the name of the log file as you review the contents of `/etc/rsyslog.conf` on serverX.

1. All authentication-related messages go to

authpriv — /var/log/secure

2. Anything e-mail related goes to

/var/log/maillog

3. Messages related to cron go to

/var/log/cron

follow or see update

4. All other messages sent at **info**

priority or higher are saved in

/var/log/messages

fail(-f) /var/log/cron

Rotating Logs

- Logs are "rotated" to keep them from filling up the file system containing `/var/log/`.
- When a log file is rotated, it is renamed with an extension indicating the date on which it was rotated: the old `/var/log/messages` file may become `/var/log/messages-20101030` if it is rotated on October 30, 2010.
- Once the old log file is rotated, a new log file is created and the service that writes to it is notified.
- After a certain number of rotations (typically after four weeks), the old log file is discarded to conserve disk space.
- A cron job runs the logrotate program daily to see if any logs need to be rotated.
- Most log files are rotated weekly, but logrotate rotates some faster, or slower, or when they reach a certain size.

/etc/cron.daily/logrotate



References

rsyslog Manual

• /usr/share/doc/rsyslog-*/manual.html

rsyslog.conf(5) and **logrotate(8)** man pages



Practice Quiz

Review rsyslog

Answers the following questions:

1. Which two fields are used to match log events?

facility , priority

2. What is the effect of a wildcard in the first field?

all facility

3. What is the effect of a wildcard in the second field?

all priority

4. Is it possible for the same log event to be recorded in more than one log?

yes.

Locate and Analyze a Log Summary Report

A program called **logwatch** can be installed which will automatically analyze the standard log files and send a summary e-mail to **root**. A quick way to check the status of a system is to read this log summary report.

logwatch runs as a daily cron job to generate its report of interesting log information. This report is e-mailed to the local **root** account, by default.

If you want to generate a logwatch email manually, simply run **logwatch**.



References

`/usr/share/doc/logwatch-*`

logwatch(8) man page

`logwatch --range=today --print`



Practice Performance Checklist

Analyze a Log Summary Report

Determine the amount of free space for the root filesystem from the latest logwatch report on serverX.

- Open the email of root
- Locate and read the most recent logwatch report. If there is no logwatch report, run the **logwatch** command to generate one manually.
- Record the amount of free space for the / filesystem:

Change the Log Summary e-mail Address

/etc/logwatch/conf/logwatch.conf is an empty file which contains local logwatch settings. The system wide default settings for **logwatch** are kept in **/usr/share/logwatch/default.conf/logwatch.conf**. An excerpt is shown below:

```
# Default Log Directory
# All log-files are assumed to be given relative to this directory.
LogDir = /var/log

# You can override the default temp directory (/tmp) here
TmpDir = /var/cache/logwatch

# Default person to mail reports to. Can be a local account or a
# complete email address. Variable Print should be set to No to
# enable mail feature.
MailTo = root

# Default person to mail reports from. Can be a local account or a
# complete email address.
MailFrom = Logwatch

# if set, the results will be saved in <filename> instead of mailed
# or displayed.
#Save = /tmp/logwatch

# The default time range for the report...
# The current choices are All, Today, Yesterday
Range = yesterday

# The default detail level for the report.
# This can either be Low, Med, High or a number.
# Low = 0
# Med = 5
# High = 10
Detail = Low

# The 'Service' option expects either the name of a filter
# (in /usr/share/logwatch/scripts/services/*) or 'All'.
# The default service(s) to report on. This should be left as All for
# most people.
Service = All

# By default we assume that all Unix systems have sendmail or a sendmail-like system.
# The mailer code Prints a header with To: From: and Subject:.
mailer = "sendmail -t"

# By default the cron daemon generates daily logwatch report
# if you want to switch it off uncomment DailyReport tag.
# The implicit value is Yes
#
# DailyReport = No
```

Normally you edit the file in **/etc**, not the defaults in **/usr/share**. The **MailTo** element takes an e-mail address to send logwatch reports to; by default logwatch uses **root@localhost**. If you wanted **student@serverX.example.com** to receive these mailings, add the following line to the **/etc/logwatch/conf/logwatch.conf** file:

MailTo = student@serverX.example.com



References
logwatch(8) man page



Practice Case Study

Redirect Log Summary e-mails

Change the configuration of **logwatch** on serverX to send log summary reports to user **student** rather than user **root**.

Bonus Question: How would you send the **logwatch** reports to both **student** and **root**?

How would you address the case study described above? Take notes on your process in the space below and then implement it.

MailTo = student, root



Test

Criterion Test

Case Study

Log Debugging Messages

Your company wants to include a debug log on your server for analysis. Redirect all debugging level messages (and higher priority) to a file named `/var/log/debug.log`.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Determine Log Destinations

In this section you learned how to:

- Manage the dispatching of software logs with `rsyslog`
- Protect filesystem with `logrotate`

Locate and Analyze a Log Summary Report

In this section you learned how to:

- Utilize log summary emails

Change the Log Summary e-mail Address

In this section you learned how to:

- Manage the dispatching of log summaries



UNIT SIX

INSTALLING AND MANAGING SOFTWARE

Introduction

Topics covered in this unit:

- Registration with Red Hat Network (RHN)
- Using yum to manage software packages
- Using rpm to get package information
- Third-Party/Private Repositories

Register with Red Hat Network (RHN)

What is Red Hat Network?

Red Hat Network is a centrally-managed service that makes it easy to deploy software and software updates to Red Hat Enterprise Linux systems and to remotely manage and monitor those systems. You can use the "hosted" RHN service managed by Red Hat, or you can set up and manage your own RHN Satellite in your organization. Either way, to get package updates for your clients from RHN and to have them show up in your web management interface, you need to start by registering those systems with the RHN server of your choice.

Using `rhn_register`

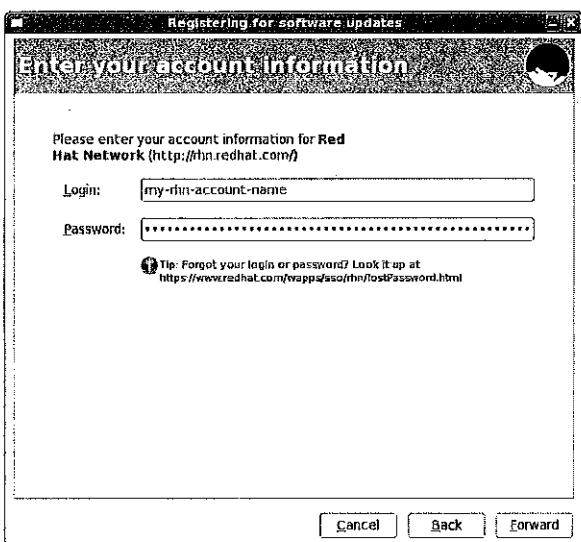
Start the Red Hat Network (RHN) registration process by running the `rhn_register` command from the command-line or choosing it from the GUI menu: **System → Administration → RHN Registration**

If you have a RHN Satellite or RHN Proxy server, choose the **I have access to a Red Hat Network Satellite...** button in the GUI. Fill in the DNS name of the RHN Satellite server or RHN Proxy server.

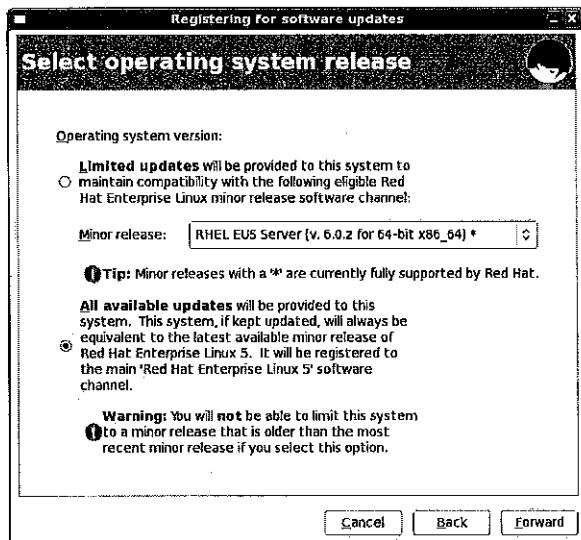
If you do not have a RHN Satellite or RHN Proxy server, or you want to register with Hosted RHN, choose the **I'd like to receive updates from Red Hat Network** button.

If you need to set proxy setting for the connection, click on the **Advanced Network Configuration...** button and fill in the appropriate fields.

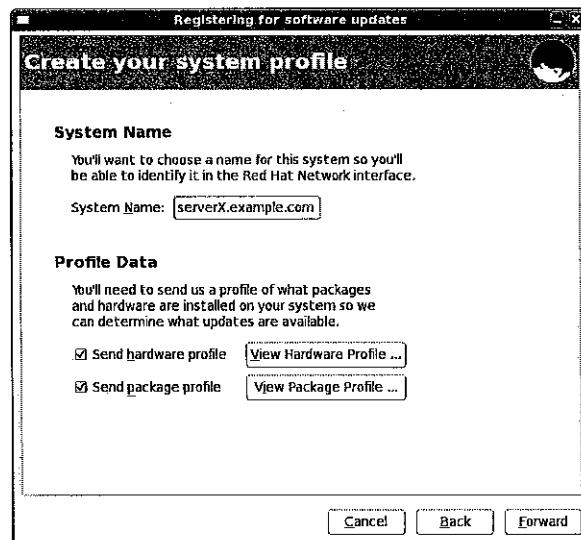
Fill in your Red Hat Network account information. If you have forgotten your account name or password, or you need to create a new account, go to <https://www.redhat.com/wapps/sso/login.html>



The next screen allows you to limit updates to maintain compatibility with Red Hat Enterprise Linux minor releases. If you want this ability choose **Limited updates**. If you want all the current updates, choose **All available updates**.

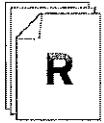


Enter the name for your system (it will use the current hostname by default), and optionally send the hardware and package profile to RHN.



Note

The **rhn_register** command works equally well in a graphical environment or a text environment. If you run **rhn_register** in a text-only environment, it will prompt for information much as the GUI does.



References

rhn_register(8) and **rhnplugin(8)** man pages

Knowledgebase: "What is the command rhn_register used for in Red Hat Enterprise Linux?"

<https://access.redhat.com/kb/docs/DOC-11217>

Knowledgebase: "I had to re-install my system. How do I re-register my system with Red Hat Network (RHN)?"

<https://access.redhat.com/kb/docs/DOC-8037>

Red Hat Enterprise Virtualization for Servers 2.2: 5.5-2.2 Hypervisor Deployment Guide

- Section 5.1.7: Register to RHN



Practice Quiz

Red Hat Network Registration

1. The command-line tool that begins the registration with the Red Hat Network is rhn_register.
2. The first registration choice determines whether a system registers with rhn or satellite proxy.
3. Optionally additional HTTP proxy server and authentication information may need to be provided.
4. An login name and its matching password must be provided for successful Red Hat Network registration.
5. The last questions to be answered during the registration process are system name and whether to upload hardware and package profile information.

Using yum

yum is a powerful command-line tool that can be used to more flexibly manage (install, update, remove, and query) software packages. Official Red Hat packages are normally downloaded from Red Hat Network (RHN). When you register your machine with RHN, **yum** is automatically configured to use it. You can also configure **yum** to get packages from third-party package repositories over the network.

Basic yum Commands

1. **yum help** will display usage information
2. **yum list** displays installed and available packages *installed, updates, etc*
3. **yum search KEYWORD** lists packages by keywords
4. **yum info PACKAGE NAME** gives detailed information about a package
5. **yum install PACKAGE NAME** obtains and installs a software package, including any dependencies
6. **yum remove PACKAGE NAME** removes an installed software package, including any supported packages
7. **yum update PACKAGE NAME** obtains and installs a newer version of the software package, including any dependencies. Generally the process tries to preserve configuration files in place, but in some cases they may be renamed if the packager thinks the old one will not work after update. With no PACKAGE NAME specified, it will install all relevant updates.
8. **yum provides PATHNAME** displays packages that match the pathname specified (which often include wildcard characters)

Use this space for notes

Example yum Commands:

To search for packages that have "web server" in their description, summary, or package name:

```
[root@serverX ~]# yum search 'web server'
=====
Matched: web server =====
mod_auth_mysql.x86_64 : Basic authentication for the Apache web server using a
                         : MySQL database
webalizer.x86_64 : A flexible Web server log file analysis program
freeradius.x86_64 : High-performance and highly configurable free RADIUS server
hsqldb.x86_64 : Hsqldb Database Engine
htdig.x86_64 : htd://Dig - Web search engine
htdig-web.x86_64 : Scripts and HTML code needed for using ht://Dig as a web
                         : search engine
httpd.x86_64 : Apache HTTP Server
```

To get information on the Apache HTTP Server:

```
[root@serverX ~]# yum info httpd
Available Packages
Name        : httpd
Arch       : x86_64
Version    : 2.2.15
Release    : 5.el6
Size       : 811 k
Repo       : base
Summary    : Apache HTTP Server
URL        : http://httpd.apache.org/
License    : ASL 2.0
Description: The Apache HTTP Server is a powerful, efficient, and extensible
             web server.
```

To install, update and remove the **httpd** package:

```
[root@serverX ~]# yum install httpd
[root@serverX ~]# yum update httpd
[root@serverX ~]# yum remove httpd
```

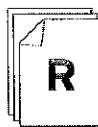


Warning

yum remove will remove the package(s) listed and *any package that requires the package(s) being removed* (and package(s) which require those packages, and so on). This can lead to unexpected removal of packages, so carefully check the list of packages to be removed.

yum Component Groups

yum also has the concept of *component groups*, which are collections of related software grouped around a particular solution. Review **yum help** filtering on lines with the phrase "group".



References

Red Hat Enterprise Linux Deployment Guide

- Chapter 1: Yum

yum(1), yum.conf(5) man pages

yum -v group list
 yum group info
 yum groupinstall

yum -plugin-remove-with-leaves

/etc/yum.conf

look into ksplice.com



Practice Exercise

Searching For and Installing Packages

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Login as **root** on serverX and perform the following tasks:

1. Attempt to run the command **gnuplot**. You should find that it is not installed.
2. Search for plotting packages.
3. Find out more information about the **gnuplot** package.
4. Install the **gnuplot** package.
5. Attempt to remove the **gnuplot** package, but say no.

How many packages would be removed? 1

6. Attempt to remove the **gnuplot-common** package, but say no.

How many packages would be removed? 2

Handling Third-Party Software

The **rpm** utility is a low-level tool that can get information about the contents of package files and installed packages. It gets its information from a local database or the package files themselves.

The general form of a query is:

- **rpm -q [select-options] [query-options]**
- **rpm --query [select-options] [query-options]**

RPM Queries: Select Options

- **-q -a** - all installed packages
- **-q *PACKAGENAME*** - currently installed *PACKAGENAME*
- **-q -p *PACKAGEFILE.rpm*** - package file named *PACKAGEFILE.rpm*
- **-q -f *FILENAME*** - what package provides *FILENAME*

*-q = --last by time
-a shows reverse of cat
-p package namefile*

RPM Queries: Information About Content of Packages

- **-q -l** - lists the package's name and version; compare to **yum list**
- **-q -i** - package information; compare to **yum info**
- **-q -l** - list of files installed by the specified package
- **-q -c** - list just the configuration files
- **-q -d** - list just the documentation files
- **-q --scripts** - list shell scripts that may run after the package is installed or uninstalled
- **-q --changelog** - list change information for the package



Note

The **repoquery** command can also be used to get information about packages and their contents. It differs from **rpm** by looking up that information in yum's repositories and RHN instead of the local database of installed packages.

Example rpm Query Commands:

Querying installed packages:

```
[root@serverX ~]# rpm -q samba-client
samba-client-3.5.4-68.el6.x86_64
[root@serverX ~]# rpm -q zlib -l
```

```
/lib64/libz.so.1
/lib64/libz.so.1.2.3
/usr/share/doc/zlib-1.2.3
/usr/share/doc/zlib-1.2.3/ChangeLog
/usr/share/doc/zlib-1.2.3/FAQ
/usr/share/doc/zlib-1.2.3/README
[root@serverX ~]# rpm -q httpd --scripts
preinstall scriptlet (using /bin/sh):
# Add the "apache" user
getent group apache >/dev/null || groupadd -g 48 -r apache
getent passwd apache >/dev/null || \
    useradd -r -u 48 -g apache -s /sbin/nologin \
        -d /var/www -c "Apache" apache
exit 0
postinstall scriptlet (using /bin/sh):
# Register the httpd service
/sbin/chkconfig --add httpd
preuninstall scriptlet (using /bin/sh):
if [ $1 = 0 ]; then
    /sbin/service httpd stop > /dev/null 2>&1
    /sbin/chkconfig --del httpd
fi
posttrans scriptlet (using /bin/sh):
/sbin/service httpd condrestart >/dev/null 2>&1 || :
```

Querying and installing package files:

```
[root@serverX ~]# cd /net/instructor/var/ftp/pub/materials/
[root@serverX ~]# rpm -q -p wonderwidgets-1.0-4.x86_64.rpm -l
/etc/wonderwidgets.conf
/usr/bin/wonderwidgets
/usr/share/doc/wonderwidgets-1.0
/usr/share/doc/wonderwidgets-1.0/README.txt
[root@serverX ~]# rpm -q -p wonderwidgets-1.0-4.x86_64.rpm -i
Name        : wonderwidgets                    Relocations: (not relocatable)
Version     : 1.0                           Vendor: Red Hat, Inc.
Release     : 4                            Build Date: Fri 03 Dec 2010 05:42:55 AM EST
Install Date: (not installed)           Build Host: station166.rosemont.lan
Group       : GLS/Applications          Source RPM: wonderwidgets-1.0-4.src.rpm
Size        : 4849                         License: GPL
Signature   : (none)
Summary     : Demonstration package for use in GLS training.
Description :
A demonstration package that installs an executable, and a config file.
[root@serverX ~]# rpm -q -p wonderwidgets-1.0-4.x86_64.rpm -c
/etc/wonderwidgets.conf
[root@serverX ~]# rpm -q -p wonderwidgets-1.0-4.x86_64.rpm -d
/usr/share/doc/wonderwidgets-1.0/README.txt
[root@serverX ~]# yum localinstall wonderwidgets-1.0-4.x86_64.rpm
...
Package wonderwidgets-1.0-4.x86_64.rpm is not signed
[root@serverX ~]# yum localinstall --nogpgcheck wonderwidgets-1.0-4.x86_64.rpm
[root@serverX ~]# rpm -q wonderwidgets
wonderwidgets-1.0-4.x86_64
```

Using yum to Install Local Package Files

yum localinstall *PACKAGEFILE.rpm* can be used to install package files directly. It automatically downloads any dependencies the package has from RHN and any configured **yum** repositories. Packages are normally digitally signed to ensure they are legitimate; if the package

is not signed by a key trusted by your system, it will be rejected. The **--nogpgcheck** option can disable the signature check if you are certain the package is legitimate.



Note

rpm -ivh PACKAGEFILE.rpm can also be used to install package files. However, using **yum** helps maintain a transaction history kept by **yum** (see **yum history**).



Important

Be careful when installing packages from third parties, not just because of the software that they may install, but because the RPM may run arbitrary scripts as **root** as part of the installation process.

RPM package files are essentially **cpio** archives. The **cpio** command is an archiving tool like **zip** or **tar**, but differs in that it reads the list of files to be archived from STDIN rather than as command line arguments. Pipe the output of **rpm2cpio PACKAGEFILE.rpm** into **cpio -id**, and it will extract all the files stored in the RPM to the current directory.

*-imuf
-imud*



References

Red Hat Enterprise Linux Deployment Guide

- Section 3.2: Using RPM

rpm(8), **repoquery(1)**, and **rpm2cpio(8)** man pages

file-roller - Gui filemanager



Practice Exercise

Handling Third-Party Software

Carefully perform the following steps. Ask your instructor if you have problems or questions.

In this exercise you will gather information about a third-party package, extract files from it, and install it as a whole on your desktopX system.

1. Download *wonderwidgets-1.0-4.x86_64.rpm* from <http://instructor/pub/materials>.
- 81P
2. What files does it contain?
- - scripts
3. What scripts does it contain?
- - scripts
4. How much disk space will it use when installed? ~~240~~ 81P
5. Use **yum localinstall** to install the package.

Using Third-Party Repositories

epel

Third-party repositories are network-accessible directories of software package files which can be accessed by **yum**, provided outside of Red Hat Network. Yum repositories are used by non-Red Hat distributors of software, or for small collections of local packages. (For example, Adobe provides some of its free software for Linux through a yum repository.) The **instructor** classroom server actually hosts yum repositories for this class.

Put a file in the **/etc/yum.repos.d/** directory to enable support for a new third-party repository. Repository configuration files must end in **.repo**. The repository definition contains the URL of the repository, a name, whether to use GPG to check the package signatures, and if so the URL pointing to the trusted GPG key.

Examples of /etc/yum.repos.d/*.repo Configuration Files:

An example with a single repository, with security checks of downloaded packages disabled:

```
[GLS]
name=Instructor GLS Repository
baseurl=ftp://instructor.example.com/pub/gls
gpgcheck=0
```

An example with multiple repository references in a single file:

```
[base]
name=Instructor Server Repository
baseurl=http://instructor.example.com/pub/rhel6/dvd
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

# Optional rhel6
[optional]
name=Instructor Optional Repository
baseurl=http://instructor.example.com/pub/rhel6/Optional
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[client]
name=Instructor Client Repository
baseurl=http://instructor.example.com/pub/rhel6/Client
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
enabled=0

[kernel-extras]
name=Instructor Kernel Extras Repository
baseurl=http://instructor.example.com/pub/rhel6/Kernel-Extras
gpgcheck=1
```



Note

Note that some repositories, such as EPEL (Extra Packages for Enterprise Linux), provide this configuration file as part of an RPM package that can be downloaded from the web and installed with **yum localinstall**.

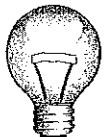
Installing the Red Hat Enterprise Linux 6 EPEL repo package:

```
[root@serverX ~]# rpm --import http://download.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-6
[root@serverX ~]# yum install http://download.fedoraproject.org/pub/epel/beta/6/x86_64/epel-release-6-5.noarch.rpm
[root@serverX ~]# cat /etc/yum.repos.d/epel.repo

[epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6

[epel-debuginfo]
name=Extra Packages for Enterprise Linux 6 - $basearch - Debug
#baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch/debug
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-6&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
gpgcheck=1

[epel-source]
name=Extra Packages for Enterprise Linux 6 - $basearch - Source
#baseurl=http://download.fedoraproject.org/pub/epel/6/SRPMS
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-source-6&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
gpgcheck=1
```



Important

Install the RPM GPG key before installing signed packages. This will verify that the package belongs to a key you have imported. Otherwise, **yum** will complain about the missing key. (You can use the **--nogpgcheck** option to ignore missing GPG keys, but this could cause forged or insecure packages to be installed on your system.)



References

Red Hat Enterprise Linux Deployment Guide

- Section 1.3: Configuring Yum and Yum Repositories

yum(1) and **yum.conf(5)** man pages



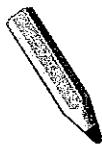
Practice Exercise

Using yum Repositories

Carefully perform the following steps. Ask your instructor if you have problems or questions.

You will configure your server to use a separate **yum** repository to obtain updates, and update your machine.

1. Create the file **/etc/yum.repos.d/errata.repo**, to enable the "Updates" repository found on the instructor machine. It should access content found at the following URL: `ftp://instructor.example.com/pub/rhel6/Errata`
2. Update all relevant software provided by the repository, using **yum update**.



Test

Criterion Test

Case Study

Update and Install Software

Before you begin...

Reset your serverX lab system with the **lab-setup-server** command.

You have a new server, serverX, to administrate that has very specific software requirements. It must have the latest version of the following packages installed:

kernel (existing package w/ an update)

xsane-gimp (new package)

bzip2 (updated package)

For security reasons it should not have the **xinetd** package installed.

Do not install all updates. Only install updates for the packages listed above if they are available.

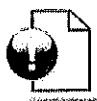
Updated packages can be found at the following URL: <ftp://instructor.example.com/pub/rhel6/Errata>

When you finish, run the **lab-grade-packages-2** evaluation script to make sure that you have done everything correctly.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Register with Red Hat Network (RHN)

In this section you learned how to:

- Register a system with Red Hat Network

Using yum

In this section you learned how to:

- List packages by name, keyword, or file
- Get the version and description of a package
- Install, update and remove packages with **yum**

Handling Third-Party Software

In this section you learned how to:

- Query third-party packages for files before installation
- Query rpm package internals

Using Third-Party Repositories

In this section you learned how to:

- Manage repository definition files in **/etc/yum.repos.d/**



UNIT SEVEN

ADMINISTER REMOTE SYSTEMS

Introduction

Topics covered in this unit:

- Remote shell access
- Remote file transfers
- SSH keys
- SSH hardening
- Archives and compression

Remote Shell Access

The Secure Shell (SSH) is one of the most versatile system administration tools. It allows login and execution of commands on remote systems. It uses strong encryption and host keys as a protection against network sniffing. It is the only network service which is enabled by default and is remotely accessible. The OpenSSH server configuration usually does not require modification.

Fill in the blanks as your instructor demonstrates the use of **ssh** and covers these key points.

1. SSH is more secure than telnet because all communication between hosts is encrypted.
2. **ssh -X user@host.fqdn** initiates a remote connection to host.fqdn as **user**.
3. The first time an SSH connection is made to a system, the public key of the remote system is stored locally so its identity can be verified each time a future connection is started.
4. The exit command is used to finish an SSH session and return to the local shell.



References

Red Hat Enterprise Linux Deployment Guide

- Section 9.3.1: Using the ssh Utility



Practice Quiz

Remote Shell Access Quiz

Connect to serverX from desktopX using a remote shell. Answer the following questions running commands from that remote shell:

1. The Disk Utility command is **palimpsest**.

/dev/vda is the name of the hard drive on serverX.

2. santiago is the name of the OS release according to **/etc/redhat-release**.

3. Run **nautilus** or use the command-line in the remote shell on serverX to perform the following:

- Create a file named **a1.txt** in **/root**
- Create a directory named **b2** in **/home/student** which is owned by the **student** user and the **student** group.

Remote File Transfers

Compare and Contrast: Local vs. Remote File Copy

Fill in the open fields.

| | | Local File Copy | Remote File Copy |
|--------------------|--|---|------------------|
| Command | <code>cp</code> | | |
| Syntax | <code>cp original-file new-file</code> | | |
| Arguments | Can use pathnames for arguments | In addition to pathnames, the files can have the following syntax: target:pathname , where target = [user@]host.fqdn. Specify user@ when the remote username is different. | |
| Scope of operation | Only works with local files | | |

Table 7.1. Local vs. Remote File Copy Comparison

Use this space for notes

rsync -aP target: —
 ↑
 archive
 parent
 print out
 rsync can only handle one remote machine



References

`rsync(1)` man page



Practice Performance Checklist

Remote File Transfers

- Use **rsync** to backup **student**'s home directory on desktopX to the **/tmp** directory on serverX.
- Create a new file named **z.txt** in **student**'s home directory.
- Use the same **rsync** command to backup **student**'s home directory on desktopX to the **/tmp** directory on serverX.
- Remove the **Desktop** directory from the backup on serverX. Run the same **rsync** command.

Using SSH Keys

The Secure Shell, **ssh**, allows you to authenticate using a private-public key scheme. This means that you generate two keys, called your private key and your public key. The private key should, as the name implies, be kept private. The public key can be given to anyone. An ssh server that has your public key can issue a challenge that can only be answered by a system holding your private key. As a result, you can authenticate using the presence of your key. This allows you to access systems in a way that does not require typing a password every time but is still secure.

Key generation is done using the **ssh-keygen** command. You can use a key type of DSA or RSA with SSH version 2. SSH protocol version 1 is known to have a security flaw, and therefore its use is not recommended unless you need to connect to legacy ssh servers.

During key generation, you will be given the option to specify a passphrase, which must be provided in order to access your private key. This way, even if the key is stolen, it is very difficult for someone other than you to use it. This gives you time to make a new key pair and remove all references to the old ones, before the private key can be used by an attacker who has cracked it.

It is always wise to passphrase-protect your private key since the key allows you to access other machines. However, this means that you must type your passphrase whenever the key is used, making the authentication process no longer password-less. This can be avoided using **ssh-agent**, which can be given your passphrase once at the start of your session (using **ssh-add**) so it can provide it as necessary while you stay logged in.

Once your SSH keys have been generated, they are stored by default in the **.ssh/** directory of your home directory. Default modes should be 600 on your private key and 644 on your public key.

Before you can use key-based authentication, you will need to copy your public key to the destination system. This can be done with **ssh-copy-id**.

```
[student@desktopX ~]$ ssh-copy-id -i .ssh/id_rsa.pub root@desktopY
```

When you copy your key to another system via **ssh-copy-id**, it uses the **~/.ssh/id_rsa.pub** file by default. If you use a different key, or give your key a different name, it will have to be specified with the **-i** option when using **ssh-copy-id**.

SSH Key Demonstration

- Use **ssh-keygen** to create a public-private keypair.
- Use **ssh-copy-id** to copy the public key to the correct location on a remote system. For example:

```
[root@serverX]# ssh-copy-id root@serverY.example.com
```



References

Red Hat Enterprise Linux Deployment Guide

- Section 9.2.4: Using a Key-Based Authentication

ssh-keygen(1), ssh-copy-id(1), ssh-agent(1), ssh-add(1) man pages

eval \$(ssh-agent)

ssh-add

ssh - - -



Practice Performance Checklist

Securely Transferring Backups

- Create an SSH key-pair as **student** on desktopX.
- Send the SSH public key to the **student** account on serverX.
- Run the **rsync** command used before to backup **student**'s home directory on desktopX to the **/tmp** directory on serverX.

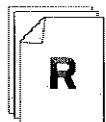
Securing SSH Access

While OpenSSH server configuration usually does not require modification, additional security measures are available.

In this activity, we are going to "discover" how to disable remote root logins and the use of passwords (require use of keys).

Securing SSH Search & Learn

1. Use **yum** or **rpm** to determine which package provides the SSH service.
2. Use **rpm -q -l** or **rpm -q -c** to determine the primary configuration file for the service.
3. Reviewing the man page for the configuration file, which directive disables root login?
4. Which directive in that configuration file disables password login?



References

- Red Hat Enterprise Linux Deployment Guide
- Section 9.2.4: Using a Key-Based Authentication



Practice Performance Checklist

Securing SSH

- If not done earlier, generate SSH keys on desktopX and copy the public key to the **student** account on serverX and verify that the keys are working:
*PermitRoot...
PermitRoot...*
- Configure SSH on serverX to prevent root logins.
- Restart the SSH service.
- Confirm that **root** cannot log in with SSH, but **student** is permitted to log in.
- Configure SSH on serverX to prevent password authentication.

Archives and Compression

Archiving and compressing files are useful when creating backups and transferring data across a network. We will explore the command-line tools to archive and compress files.

Key tar Options

1. c = create
2. x = extract -C /
3. t = test
4. v = verbose
5. T = file name
6. Z = gzip tar.gz
tar.bz2
7. j = bzip2

Example tar Syntax

Create a gzip-compressed archive backup of the **/etc** directory.

```
tar cvzf /tmp/etc.tar.gz /etc
```

Get a "long" listing of files in a bzip2-compressed archive.

```
tar tvjf /tmp/archive.tar.bz2
```

Extract all files from a gzip-compressed archive to the current directory.

```
tar xvzf /tmp/etc.tgz
```

Additionally, **gzip** and **bzip2** can be used independently to compress files. **gunzip** and **bunzip2** are the corresponding decompress commands.

Use this space for notes



References

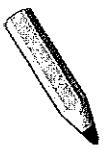
tar(1), **bzip2(1)**, and **gzip(1)** man pages

**Practice Exercise**

Archive and Compress Files

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Login to serverX as **root**.
2. Create an archive of **/etc** using **gzip** compression. Save the file as **/tmp/etc.tar.gz**.
3. Copy the **/tmp/etc.tar.gz** file to **/backups** on your desktopX machine.
4. Extract the compressed archive to **/backups** on desktopX.



Test

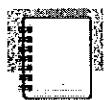
Criterion Test

Exercise

SSH Keys and File Archives

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Reset serverX by running **lab-setup-remote**.
2. Install the SSH public key generated previously on desktopX to the **student** account on serverX.
3. Archive **student**'s home directory on desktopX into **/tmp/student.tar.bz2**.
4. Copy the **/tmp/student.tar.bz2** file on desktopX to **/tmp** on serverX.
5. Remove **student**'s home directory on serverX.
6. Restore **student**'s home directory from the **/tmp/student.tar.bz2** archive.
7. Install the SSH public key from the backup to desktopX and verify you can use the SSH keys to get from serverX to desktopX without typing a password.
8. Verify you have accomplished the tasks by running **lab-grade-remote**.



Personal Notes



Unit Summary

Remote Shell Access

In this section you learned how to:

- Describe the steps taken by SSH to initiate a secure link
- Use SSH to access a remote shell prompt

Remote File Transfers

In this section you learned how to:

- Copy files securely to/from a remote server

Using SSH Keys

In this section you learned how to:

- Create a user SSH key pair and will deploy the public key on a remote system

Securing SSH Access

In this section you learned how to:

- Configure SSH to prohibit root login
- Configure SSH to prohibit password login, but allow access with ssh keys

Archives and Compression

In this section you learned how to:

- Archive and compress files using command-line tools (**tar**, **gzip**, **bzip2**)



UNIT EIGHT

DEPLOY AND SECURE SERVICES

Introduction

Topics covered in this unit:

- System clock management
- Manage service startup
- Desktop server (VNC) configuration
- FTP server deployment
- FTP server configuration
- Web server deployment
- Firewall

Manage the System Clock

The PC hardware clock is not accurate enough for many applications. It tends to drift over time. Many applications require exact timing and may react poorly if the clock is reset suddenly to a new time.

Network Time Protocol (NTP) counters the drift by modifying the length of a second, much like tuning the pendulum of an old fashioned clock. If the system's time is behind the combined reference time of the time servers the second is made shorter so that the system clock races towards the correct time. Thus the time difference is reduced gently without disturbing other applications. However if the time differs too greatly, NTP ceases to work.

Having three central time servers allows clients to reject bogus synchronization messages if one of the servers' NTP daemons or clocks malfunctions. It is possible for a client to synchronize with fewer time servers if necessary but it is less secure.

system-config-date

Configure NTP Service

1. Launch the Date & Time management tool.
2. Click the Time Zone tab.
3. Set the timezone for your locale.
4. Select UTC.
5. Click the Date and Time tab.
6. Enable NTP, point to **instructor.example.com**.
7. Synchronize clock immediately.

update /etc/ntp.conf

To test what systems you are connected to:

```
[root@serverX ~]# ntpq -c peers
      remote      refid    st t when poll reach offset jitter
=====
*instructor.exam LOCAL(0) 11 u    24    64  377  0.198 -10.882 25.501
```



References

- Red Hat Enterprise Linux Deployment Guide
 - Chapter 13: Date and Time Configuration

NTP: Network Time Protocol
<http://www.ntp.org/>



Practice Performance Checklist

Manage the System Clock

- Configure desktopX to synchronize with instructor.example.com using NTP.
- Set the timezone to the appropriate setting for your locale.
- Make the hardware clock store UTC time.

Manage Services

In Unix (and Linux), daemons are processes that "wait or run in the background" performing various tasks. Generally, daemons start automatically at bootup and continue to run for the life of the system. Conventionally, daemon executables end in the letter "d".

Daemons are managed by service scripts which reside in the `/etc/rc.d/init.d/` directory. Service scripts expect to be called with a single **start**, **stop**, **restart**, **status**, or **reload** argument. Service scripts should be invoked using the front-end **service** command. Services are enabled (configured to start automatically at boot time) with **chkconfig service on**. Services are disabled (configured not to start automatically at boot time) with **chkconfig service off**. While the list of "runlevels" may seem intimidating, for most purposes we can just use **on** and **off** and not worry about them. Without arguments, **chkconfig** lists the current configuration of all services. The state of a single service can be observed with **chkconfig --list service**. Running the **chkconfig** command does not immediately affect the state of a daemon started from a service script.

Installing New Services

What command is commonly used to perform each of these steps of deploying a new service on a Red Hat Enterprise Linux system?

- **Install** the software: `yum install rpm -qf tigervnc-server & rpm -qf init.d`, configure the service
- **Start** the service: `service sshd start`
- **Enable** the service at bootup: `chkconfig sshd on`
- **Test** the services: will be covered on the next pages

Use this space for notes



References

Red Hat Enterprise Linux Deployment Guide

- Chapter 7: Controlling Access to Services

service(8) and **chkconfig(8)** man pages

Configuring a VNC Server

While many data centers will standardize on **ssh** for remote administration of Unix and Linux systems, some will use Virtual Network Computing (VNC) for remote administration of Windows servers. Red Hat Enterprise Linux 6 supports the implementation of a VNC server that can allow one or more remote graphical desktops.

Configure a VNC Server Demonstration

1. Install the VNC server package

```
[root@demo ~]# yum install tigervnc-server
```

2. Edit **/etc/sysconfig/vncservers**:

```
VNCSEVERS="2:root"  
VNCSEVERARGS[2]="-geometry 800x600 -nolisten tcp -localhost"
```

The **-localhost** option will prevent remote VNC clients connecting except when doing so through a secure tunnel, for example, when using **vncviewer** and its **-via** option:

```
vncviewer -via user@remotehost localhost:2
```

3. Set a VNC password.

```
[root@demo ~]# vncpasswd  
Password: password  
Verify: password
```

4. Start and enable the service:

```
[root@demo ~]# service vncserver start  
[root@demo ~]# chkconfig vncserver on
```



References

Red Hat Enterprise Linux Deployment Guide

- Section 18.1.23: **/etc/sysconfig/vncservers**

vncviewer(1), **vncpasswd(1)** man pages



Practice Exercise

Enabling a VNC Server

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Install the **tigervnc-server** package on serverX.
2. Configure VNC display 1 for student.
3. Set **redhat** as the VNC password for student.
4. Start and enable the VNC service.
5. You will verify the connection in the next section.

Secure Access to a Remote GNOME Desktop

The **vncviewer** command is a viewer (client) used to connect to a VNC server running on a remote system. VNC is a clear text network protocol; there is no security against eavesdropping, interference, or hijacking of the communication.

Therefore, a more secure way to use VNC is to wrap all VNC traffic in a layer of encryption. The easiest way to do this is to tunnel the traffic over an SSH tunnel, assuming **sshd** is running on the remote system. Once the remote **sshd** service decrypts the VNC traffic, it can be passed clear text over its local loopback interface to the machine's VNC service without exposing the clear text traffic over the network.

This is such a useful approach that the **vncviewer** command has an option, **-via user@host**, which connects to the SSH server on *host* as *user* before attempting to connect to the VNC server from there. Note that the hostname given for *host* is resolved by the remote side of the connection, so if you specify *localhost* it will point at *host*, not the local client machine.



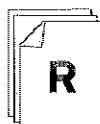
Warning

Use the **-via** option to tunnel VNC traffic over an SSH tunnel whenever possible. VNC is a cleartext protocol and your passwords and desktop session will be vulnerable to eavesdropping and interference if you do not tunnel it over a secure connection.

Connect to VNC Server Securely Demonstration

1. Connect to a VNC server using SSH:

```
[root@instructor ~]# vncviewer -via visitor@demo localhost:1
```



References

vncviewer(1) man page

ssh -f -L 5903:server3:5903 root@server3 sleep 300
 ↑ ↑ Authentication
 local port Remote Remoteport



Practice Exercise

Connect to VNC Securely

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Configure the VNC server on serverX to allow local connections only (unless you already did this in the previous exercise).
2. Connect to the VNC server on serverX securely from desktopX using an SSH tunnel
3. Verify everything is completed as specified.

Deploy an FTP Server

FTP, the File Transfer Protocol, is one of the oldest network protocols still in common use on the Internet. It provides a simple way for systems to transfer files to and from a remote server over the network.

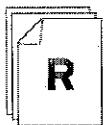
The name of the FTP server package in Red Hat Enterprise Linux 6 is **vsftpd**, which stands for Very Secure File Transfer Protocol Daemon. The service name is also called **vsftpd**.

The default configuration file supports **anonymous** download-only access to a **chrooted** tree located at **/var/ftp/**. This means that a remote FTP client can connect to the server as user **anonymous** or **ftp** with no password, and download files from the **/var/ftp** directory on the FTP server which are readable by its local **ftp** user. It also permits users on the system to connect with their password, download any files on the system they can read, and upload files to any location on the system they can write. If you plan to use the FTP server for this purpose, the default configuration settings are reasonable and do not need to be changed.

Use this space for notes

Four Steps for Deploying a Network Service

1. Install: `yum install vsftpd`
2. Start: `service vsftpd start`
3. Enable: `chkconfig vsftpd on`
4. Test: Use an FTP client such as **lftp**, **Firefox**, or **Nautilus** to see if the service is working (can you download a file from the server?)



References

- Red Hat Enterprise Linux Deployment Guide
- Chapter 7: Controlling Access to Services



Practice Performance Checklist

Deploy an FTP server

Deploy an FTP server on serverX. Verify it is working and enabled.

- Install the **vsftpd** package.
- Start the **vsftpd** service.
- Enable the **vsftpd** service.
- Publish a copy of **/etc/hosts** to the anonymous FTP document root.
- Test the FTP server on desktopX with an ftp client (**lftp** or **Nautilus**) to connect to the server `ftp://serverX.example.com`. Download the **hosts** file to student's home directory.

FTP Server Configuration

Another common FTP server configuration is an anonymous-only FTP server that only allows the anonymous client to download files, and disables all local users and uploads. In order to configure this, some changes will need to be made to the **vsftpd** configuration.

One reason turning off local user access is desirable is that FTP is a clear text network protocol, and if users use it to upload and download files from the system, they may expose their account name and password to compromise by an eavesdropping attacker. (If secure file transfer is needed by users, the SFTP service provided by **sshd** is a better choice.) Anonymous FTP by its nature is public, and files provided through anonymous FTP are assumed to be public and not sensitive in the same way as files provided by a public web site.

The **vsftpd** configuration file is found in **/etc/vsftpd/vsftpd.conf** and the document root is found in **/var/ftp/**. When you make changes to the FTP server, do not forget to restart the service.

Make sure you understand the following options:

- **anonymous_enable=YES**: enables the anonymous FTP user
- **local_enable=NO**: disables all non-anonymous local user accounts
- **write_enable=NO**: disables any user from uploading files to the FTP server

Use this space for notes



References
vsftpd.conf(5) man page



Practice Performance Checklist

Restrict FTP Access

Because FTP is an insecure protocol, it is a security risk to allow normal users to connect and authenticate. Configure your FTP server to permit anonymous connections only.

- Use **lftp** to connect to serverX and authenticate as **student** to confirm it allows non-anonymous users.
- Which file is the main vsftpd configuration file?
- Which configuration file directive controls non-anonymous access to the system?
- Configure vsftpd to deny access by local, non-anonymous users.
- Retest your server and confirm student no longer has authenticated access to your FTP server.

Deploy a Web Server

As we have seen, there is a pattern to deploying a network service: Install, Start, Enable and Test.

In this section, the class will work in small groups to walk through the deployment of an Apache web server in its default configuration. You will set it up to serve out a web page with a link to your FTP server's `/var/ftp/pub` directory.

The Apache web server package is named `httpd`, which is also the name of the service once installed.

The `httpd` configuration file is found in `/etc/httpd/conf/httpd.conf` and the document root which contains HTML pages it serves is `/var/www/html/` by default.

If you make changes to the web server's configuration file, do not forget to restart the `httpd` service.

The walk through exercise begins on the next page.

Use this space for notes



References

Red Hat Enterprise Linux Deployment Guide

- Chapter 11: The Apache HTTP Server

Apache HTTP Server Version 2.2 Documentation (*if httpd-manual is installed*)

<http://localhost/manual/>

Apache HTTP Server Version 2.2 Documentation

<http://httpd.apache.org/docs/2.2/>



Practice Performance Checklist

Deploy a Web Server

The instructor will split up the class into groups. Once you are in your group, do the following:

Given that the name of the web server package is **httpd**, deploy a web server on serverX. It should provide HTTP file services. It should be active when your server is rebooted.

- Install the **httpd** package.
- Start the **httpd** service.
- Enable the **httpd** service.
- Create a symbolic link in your web server document root to the **/pub** directory in your FTP server and call it **pub**.
- Create an **index.html** file in the document root of your web server with the following contents:

```
<h1>Classroom Web Services</h1>
<p>
<a href="pub">Click here</a> to view public files.
```

- Reboot and verify this content is available through your web browser before you notify the public to ensure your customers can access it as well.
- Test the web server using the **Firefox** browser.

Protect Services with a Firewall

The Linux kernel has a built-in firewall which can selectively allow or deny packets coming in to or going out from the system.

Red Hat Enterprise Linux 6 comes with an improved graphical management tool for the system firewall, **system-config-firewall**. You can run this program to configure the system firewall by choosing **System → Administration → Firewall** or by running **system-config-firewall**. The command **system-config-firewall-tui** starts up a text-based interface to the same tool.

In the graphical interface, at the top of the window is a button with a green circle marked **Enable**. Selecting this turns on the firewall. Likewise, the button with a red circle marked **Disable** turns it off and allows all traffic to pass. To make your changes take effect, click the button with a green checkmark marked **Apply**.

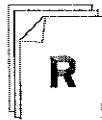
You may customize the default firewall policy to allow trusted well-known services or specific ports passage in to the system. The firewall blocks inbound traffic to your machine by default, and allows outbound traffic. You can easily add rules to allow inbound traffic for common network services in the interface; be sure to read the instructions in the window carefully to make sure you understand what the effect of selecting an item in the interface is.



Important

If you are modifying firewall rules on a remote system, take great care when modifying firewall policy not to block the network traffic you are using to remotely communicate with the system.

Use this space for notes



References

Red Hat Enterprise Linux Security Guide

- Section 2.5.2: Basic Firewall Configuration

Service iptables restart
iptables -L



Practice Performance Checklist

Allow HTTP and FTP through the Firewall

For this activity, the instructor will break you up into small groups. One student in each group will determine the steps needed to perform the activity while the others watch, make suggestions, and take notes. Once the activity has been successfully completed by the group, the students taking notes should return to their own computers and repeat the activity there.

Virtual Training students: Complete the activity on your own, and ask the instructor if you have any questions.

- If not done already, deploy a default configuration FTP server on serverX.
- If not done already, deploy a default configuration HTTP server on serverX.
- Enable the firewall on serverX
- Allow connections to the FTP service through the firewall on serverX
- Allow connections to the HTTP service through the firewall on serverX
- Allow connections to the SSH service through the firewall on serverX
- Test that each service is accessible from desktopX

Test

Criterion Test

Performance Checklist

Deploy File Sharing Services

Before you begin...

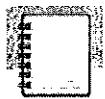
Reset your serverX lab system with the **lab-setup-server** command.

Nickel and Copper Cutlery want to publish an on-line catalog to their customers. Deploy FTP and HTTP services and confirm they are working and enabled at boot.

- Create a file called **index.html** with exactly two lines that contain the following content:

```
NICKEL AND COPPER CUTLERY  
On-line catalog coming soon!
```

- Configure serverX to provide both FTP and web services. Disable non-anonymous FTP access.
- Configure your serverX machine to serve identical file content to both anonymous FTP and HTTP users. The following URLs should both display the file you created above:
 - <ftp://serverX/pub/index.html>
 - <http://serverX/index.html>
- Grading: Reboot your serverX machine. Use a web browser to confirm your services are functioning correctly.



Personal Notes



Unit Summary

Manage the System Clock

In this section you learned how to:

- Set the system time and time zone using a graphical tool
- Configure the system to use NTP to synchronize its clock

Manage Services

In this section you learned how to:

- Start and stop services
- Enable and disable automatic starting of services
- List which services start automatically at boot time

Configuring a VNC Server

In this section you learned how to:

- Configure a VNC server

Secure Access to a Remote GNOME Desktop

In this section you learned how to:

- Connect securely to a VNC server

Deploy an FTP Server

In this section you learned how to:

- Install and activate an FTP server

FTP Server Configuration

In this section you learned how to:

- Configure the FTP server to only provide anonymous download service

Deploy a Web Server

In this section you learned how to:

- Install and activate an HTTP server

Protect Services with a Firewall

In this section you learned how to:

- Activate and deactivate the system firewall using tools
- Use a tool to modify the firewall rules to allow or deny access to specific predefined services



UNIT NINE

SELINUX MANAGEMENT

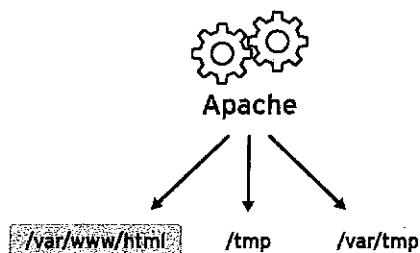
Introduction

Topics covered in this unit:

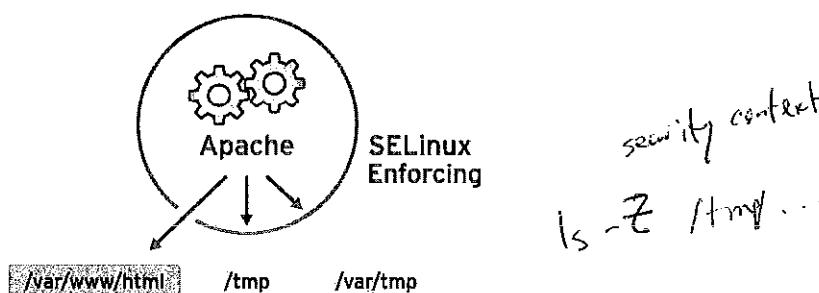
- Review basic SELinux concepts
- Displaying and setting SELinux modes
- Displaying and setting SELinux file contexts
- Tuning policy behavior with SELinux booleans
- Monitoring SELinux policy violations

Basic SELinux Security Concepts

SELinux, Security-Enhanced Linux, is an additional method to protect your system.



Presuming we want to allow remote anonymous access to a web server, we must open the ports through the firewall. However, that means that malicious people can try to crack into the system through a security exploit and, if they compromise the web server process, gain its permissions: the permissions of the apache user and the apache group. That user/group has read access to things like the document root (`/var/www/html`), as well as write access to `/tmp`, `/var/tmp` and any other files/directories that are world writable.



SELinux is a set of security rules that determine which process can access which files, directories, ports, etc. Every file, process, directory and port have special security label called SELinux contexts. A context is simply a name that is used by the SELinux policy to determine whether or not a process can access a file, directory or port. By default, the policy does not allow any interaction, so explicit rules grant access. If there is no allow rule, no access is allowed.

SELinux labels have several contexts, but we are most interested in the third context: the type context. Type context names usually end with `_t`. The type context for the web server is `httpd_t`. The type context for files and directories normally found in `/var/www/html` is `httpd_sys_content_t`. The type contexts for files and directories normally found in `/tmp` and `/var/tmp` is `tmp_t`. The type context for web server ports is `http_port_t`.

There is a rule in the policy that permits Apache (the web server process running as `httpd_t`) to access files and directories with a context normally found in `/var/www/html` and other web server directories (`httpd_sys_content_t`). There is no allow rule in the policy for files normally found in `/tmp` and `/var/tmp`, so access is not permitted. With SELinux, a malicious user could not access the `/tmp` directory, let alone write files to it. SELinux even has rules for remote filesystems such as NFS and CIFS, although all files on these filesystems are labeled with the same context.

One of the goals of SELinux is to protect the user data from system services that have been compromised.



References

- Red Hat Enterprise Linux Security-Enhanced Linux
 - Chapter 2: Introduction

Is (E) httpd...

user:role:type:sensitivity:category

targeted user type security

strict

MLS

can use user

role

type

sensitivity

ps -eZ | grep httpd

semanage port -l | grep http



Practice Quiz

Basic SELinux Concepts

1. To which of the following does SELinux apply security context (check all that apply)?

(select one or more of the following...)

- a. Ports
- b. Processes
- c. Files
- d. Directories
- e. Remote file systems

2. SELinux can be used to:

(select one or more of the following...)

- a. Protect a service from running on other ports.
- b. Protect user data from applications like the web server
- c. Block remote systems from accessing local ports
- d. Keep the system updated
- e. Access a web server

rhnsd

3. Which of the following are standard SELinux context types?

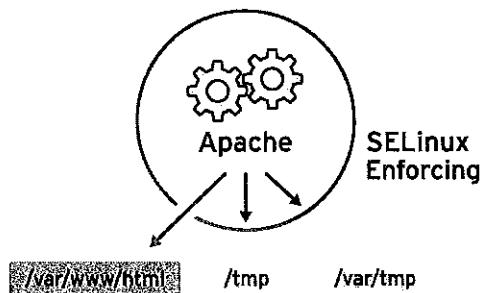
(select one or more of the following...)

- a. selinux_type
- b. object_r
- c. httpd_sys_content_t
- d. tmp_t
- e. user_u

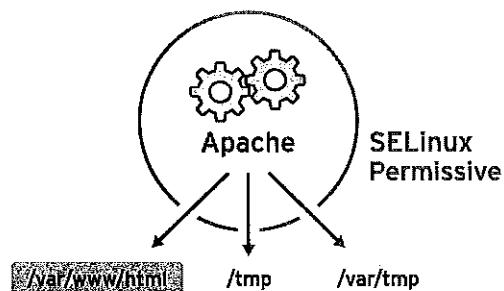
SELinux Modes

For troubleshooting purposes, we can temporarily disable SELinux protection, using SELinux modes.

*getenforce to show what mode it is
in
setenforce 0 → permissive mode
1 → back to enforcing*



In *enforcing mode*, SELinux actively denies access to the web server attempting to read files with **tmp_t** type context. In enforcing mode, SELinux both logs and protects.



Permissive mode is often used to troubleshoot issues. In permissive mode, SELinux allows all interactions, even if there is no explicit rule, and it logs all of the denied interactions. This mode can be used to determine if you are having an SELinux issue. No reboot is required to go from enforcing to permissive or back again.

A third mode, *disabled*, completely disables SELinux. You must reboot to disable SELinux entirely, or to get from disabled mode to enforcing or permissive.



Important

If you plan to re-enable SELinux restrictions, it is better to use permissive mode than to turn off SELinux entirely. One reason for this is that even in permissive mode, the kernel will automatically maintain SELinux file system labels as needed, avoiding the need for an expensive relabeling of the file system when you reboot with SELinux re-enabled.



References

Red Hat Enterprise Linux Security-Enhanced Linux

- Section 5.5: SELinux Modes



Practice Quiz

SELinux Modes

1. SELinux permissive mode allows logging, but not protection.
2. SELinux enforcing mode protects the system.
3. Which of the following are valid SELinux modes?

(select one or more of the following...)

- a. enforcing
- b. testing
- c. permissive
- d. disabled
- e. logging

Display and Modify SELinux Modes

Notice that **/etc/sysconfig/selinux** contains some useful comments:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Use **/etc/sysconfig/selinux** to change the default SELinux mode at boot time. In the example above, it is set to enforcing mode.

To display the current SELinux mode, use **getenforce**. To modify the current SELinux mode, use **setenforce**:

```
[root@serverX ~]# getenforce
Enforcing
[root@serverX ~]# setenforce
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[root@serverX ~]# setenforce 0
[root@serverX ~]# getenforce
Permissive
[root@serverX ~]# setenforce Enforcing
[root@serverX ~]# getenforce
Enforcing
```



References

Red Hat Enterprise Linux Security-Enhanced Linux

- Section 5.5: SELinux Modes

selinux(8), getenforce(1), setenforce(1) man pages

**Practice Exercise**

Changing Enforcing and Permissive Modes

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. On serverX, change the default SELinux mode to permissive and reboot.
2. After reboot, verify the system is in permissive mode.
3. Change the default SELinux mode to enforcing.
4. Change the current SELinux mode to enforcing.

Display and Modify SELinux File Contexts

Many commands that deal with files have an option (usually **-Z**) to display or set SELinux contexts. For instance, **ps**, **ls**, **cp**, and **mkdir** all use the **-Z** option to display or set SELinux contexts.

```
[root@serverX ~]# ps axZ
LABEL PID TTY STAT TIME COMMAND
system_u:system_r:init_t:s0 1 ? Ss 0:00 /sbin/init
system_u:system_r:kernel_t:s0 2 ? S 0:00 [kthreadd]
system_u:system_r:kernel_t:s0 3 ? S 0:00 [migration/0]
...
[root@serverX ~]# service httpd start
[root@serverX ~]# ps -ZC httpd
LABEL PID TTY TIME CMD
unconfined_u:system_r:httpd_t:s0 27672 ?
unconfined_u:system_r:httpd_t:s0 27675 ?
...
[root@serverX ~]# ls -Z /home
drwx-----. root root system_u:object_r:lost_found_t:s0 lost+found
drwx-----. student student unconfined_u:object_r:user_home_dir_t:s0 student
drwx-----. visitor visitor unconfined_u:object_r:user_home_dir_t:s0 visitor
[root@serverX ~]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
```

What determines a file's initial SELinux context? Normally, it is the parent directory. The context of the parent directory is assigned to the newly-created file. This works for commands like **vim**, **cp**, and **touch**, however, if a file is created elsewhere and the permissions are preserved (as with **mv** or **cp -a**), it will preserve the SELinux context as well. There are some special rules in the policy, called type transition rules, that may change the type context from the default. These rules are beyond the scope of this course.

```
[root@serverX ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
[root@serverX ~]# touch /var/www/html/index.html
[root@serverX ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/
index.html
```

semanage fcontext can be used to display or modify the rules that **restorecon** uses to set default file contexts. It uses extended regular expressions to specify the path and file names. The most common extended regular expression used in **fcontext** rules is **(/.*?)?** which means *optionally, match a / followed by any number of characters*. In essence, it will match the directory listed before the expression and everything in that directory recursively.

restorecon is part of the **policycoreutil** package, and **semanage** is part of the **policycoreutil-python** package.

```
[root@serverX ~]# touch /tmp/file1 /tmp/file2
[root@serverX ~]# ls -Z /tmp/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
```

```
[root@serverX ~]# mv /tmp/file1 /var/www/html/
[root@serverX ~]# cp /tmp/file2 /var/www/html/
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
[root@serverX ~]# semanage fcontext -l
...
/var/www(/.*)?                                all files
    system_u:object_r:httpd_sys_content_t:s0
...
[root@serverX ~]# restorecon -Rv /var/www/
restorecon reset /var/www/html/file1 context unconfined_u:object_r:user_tmp_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

The following example show how to use **semanage** to add a context for a new directory.

```
[root@serverX ~]# mkdir /virtual
[root@serverX ~]# touch /virtual/index.html
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html
[root@serverX ~]# semanage fcontext -a -f "" -t httpd_sys_content_t '/virtual(/.*)?'
[root@serverX ~]# restorecon -RFvv /virtual   ← reload/restore context
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```



References

Red Hat Enterprise Linux Security-Enhanced Linux

- Section 5.7: SELinux Contexts - Labeling Files

restorecon(8) and **semanage(8)** man pages

man -k _selinux
man httpd-selinux



Practice Exercise

Correcting SELinux File Contexts

Carefully perform the following steps. Ask your instructor if you have problems or questions.

You have been asked to adjust your remote machine's DNS configuration to exactly match the configuration from your desktop machine. You decide the easiest way is to copy the file **/etc/resolv.conf** from the local machine to the remote machine.

1. Transfer the **/etc/resolv.conf** file from your desktop machine to **root**'s home directory on serverX.
2. Shell into serverX as **root**. All of the following steps should occur on your server.
3. Observe the SELinux context of the initial **/etc/resolv.conf**.

Original **/etc/resolv.conf** context:

4. Move **resolv.conf** from **root**'s home directory to **/etc/resolv.conf**.
5. Observe the SELinux context of the newly copied **/etc/resolv.conf**.

New **/etc/resolv.conf** context:

6. Restore the SELinux context of newly positioned **/etc/resolv.conf**.
7. Observe the SELinux context of the restored **/etc/resolv.conf**.

Restored **/etc/resolv.conf** context:

Managing SELinux Booleans

SELinux booleans are switches that change the behavior of the SELinux policy. SELinux booleans are rules that can be enabled or disabled. They can be used by security administrators to tune the policy to make selective adjustments. Many packages have man pages ***_selinux(8)** which may detail some of the booleans which they use; **man -k '_selinux'** can find these man pages easily.

getsebool is used to display the booleans and **setsebool** is used to modify the booleans. **setsebool -P** modifies the SELinux policy to make the modification persistent. **semanage boolean -l** will show whether or not a boolean is persistent.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
allow_console_login --> on
allow_corosync_rw_tmpfs --> off
...
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
[root@serverX ~]# setsebool httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs --> off Allow httpd to read home directories
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on persistent
[root@serverX ~]# setsebool -P httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs --> on Allow httpd to read home directories
```



References

Red Hat Enterprise Linux Security-Enhanced Linux

- Section 5.6: Booleans

booleans(8), getsebool(8), setsebool(8), semanage(8) man pages

setfacl -m u:apache:x /home/student

Monitoring SELinux Violations

The **setroubleshoot-server** package must be installed to send SELinux messages to **/var/log/messages**. **setroubleshoot-server** listens for audit messages in **/var/log/audit/audit.log** and sends a short summary to **/var/log/messages**. This summary includes unique identifiers (*UUIDs*) for SELinux violations that can be used to gather further information. **sealert -l *UUID*** is used to produce a report for a specific incident. **sealert -a /var/log/audit/audit.log** is used to produce reports for all incidents in that file.

```
[root@serverX ~]# touch /root/file3
[root@serverX ~]# mv /root/file3 /var/www/html
[root@serverX ~]# service httpd start
[root@serverX ~]# elinks -dump http://serverX/file3
                                         Forbidden
    You don't have permission to access /file3 on this server.
[root@serverX ~]# tail /var/log/audit/audit.log
...
type=AVC msg=audit(1292526292.144:952): avc: denied { getattr } for
: pid=27675 comm="httpd" path="/var/www/html/file3" dev=dm-1 ino=54545
scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:admin_home_t:s0
tclass=file
...
[root@serverX ~]# tail /var/log/messages
...
Dec 16 14:04:59 serverX setroubleshoot: SELinux is preventing /usr/sbin/httpd "getattr"
access to /var/www/html/file3. For complete SELinux messages, run sealert -l e6e1d1d6-
d716-4e2e-863c-bba4d2b2407a
[root@serverX ~]# sealert -l e6e1d1d6-d716-4e2e-863c-bba4d2b2407a
Summary:
SELinux is preventing /usr/sbin/httpd "getattr" access to /var/www/html/file3.
```

Detailed Description:

SELinux denied access requested by httpd. **/var/www/html/file3** may be a mislabeled. **/var/www/html/file3** default SELinux type is **httpd_sys_content_t**, but its current type is **admin_home_t**. Changing this file back to the default type, may fix your problem.

Allowing Access:

You can restore the default system context to this file by executing the **restorecon** command. **restorecon '/var/www/html/file3'**, if this file is a directory, you can recursively restore using **restorecon -R '/var/www/html/file3'**.

Fix Command:

```
/sbin/restorecon '/var/www/html/file3'
...
```



Note

The "Allowing Access" section suggests **restorecon /var/www/html/file3**. If there may be other files that need to be adjusted, **restorecon** can recursively reset the context: **restorecon -R /var/www/**.



References

Red Hat Enterprise Linux Security-Enhanced Linux

- Chapter 8: Troubleshooting

sealert(8) man page

sealert -a /var/108/messages
sealert -b . gui interface



Practice Quiz

Monitoring SELinux Violations

1. What file contains log entries providing unique identifiers for SELinux violations?
/var/log/messages
2. Given the UUID of an SELinux violation, what command generates a text report of the problem?
sealert -l

Test

Criterion Test

Exercise

Managing SELinux

Before you begin...

Before you begin, run the **lab-setup-selinux** command on desktopX

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Login to serverX as **student**. Open a terminal and switch to the **root** user.
2. Copy the **web_content.tgz** archive from *instructor:/var/ftp/pub/materials* to **/tmp**.
3. Extract the archive into **/tmp**.
4. Move the extracted directory to **/var/www/html**.
5. Start the web service.
6. Try to observe the new directory with your web browser by visiting the URL http://serverX/web_content.
7. Search your system for the UUIDs of any SELinux violations your attempt to browse the newly installed content might have generated.
8. Generate text reports for the violations.
9. Follow the report's advice to restore the SELinux contexts of the newly installed content.
10. Confirm that you can view the material from your web browser by visiting the URL http://serverX/web_content.

Use **restorecon -Rv -m web-content**
instead of just **index.html**



Personal Notes



Unit Summary

Basic SELinux Security Concepts

In this section you learned how to:

- Identify basic SELinux security concepts such as context, user/role/type, and policy

SELinux Modes

In this section you learned how to:

- Describe the functional differences between SELinux enforcing and permissive modes when SELinux security is enabled

Display and Modify SELinux Modes

In this section you learned how to:

- View and change the current SELinux mode of a system
- Set the default SELinux mode of a system

Display and Modify SELinux File Contexts

In this section you learned how to:

- View the SELinux security context of processes and files
- Set the SELinux security context of files in the policy
- Restore the SELinux security context of files

Managing SELinux Booleans

In this section you learned how to:

- Use SELinux booleans to make adjustments to policy behavior

Monitoring SELinux Violations

In this section you learned how to:

- Deploy SELinux log analysis tools



UNIT TEN

MANAGING SIMPLE PARTITIONS AND FILE SYSTEMS

Introduction

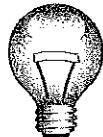
Topics covered in this unit:

- Adding file system space
- Encrypting partitions
- Adding swap space

Simple Partitions and File Systems

Storage is a basic need of every computer system. Red Hat Enterprise Linux includes powerful tools for managing many types of storage devices in a wide range of scenarios.

fdisk is a utility to manage disk partitions. You can view disks and their partitioning by running the utility with the **-l** option and the name of the disk (**fdisk -cu1 /dev/vda**). Changes can be made by running the utility interactively and choosing appropriate menu options (**fdisk -cu /dev/vda**). **-c** disables legacy DOS-compatibility mode and **-u** displays output in sectors (not cylinders, which are obsolete).



Important

Red Hat Enterprise Linux 6 automatically aligns the first partition to start at sector 2048 instead of sector 63 (the "traditional" start of cylinder 1). This is to ensure maximum performance on new 4 KIB sector hard drives as well as legacy 512 byte sector hard drives, and is compatible with the behavior of other recent operating systems that use the MBR partitioning scheme. Partition misalignment can lead to significant performance loss, so be careful adjusting these settings.

For your virtual server, **serverX**, verify the current storage configuration. Look for information in the output of the following command: **fdisk -cu1 /dev/vda**

Primary Disk:

1. Name: **/dev/vda**
2. Size: **6442 MB**
3. Total sectors: **12582912**
4. Last used sector: **9914367**

```
[root@serverX ~]# fdisk -cu1
Disk /dev/vda①: 6442 MB②, 6442450944 bytes
16 heads, 63 sectors/track, 12483 cylinders, total 12582912 sectors③
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000a9b12

      Device Boot      Start        End      Blocks   Id  System
  /dev/vda1    *        2048     526335     262144   83  Linux
  /dev/vda2          526336    9914367④    4694016   8e  Linux LVM
```

- ①** Name of disk
- ②** Total size of disk
- ③** Total sectors

- ④ Last used sector

Create a New Partition

```
[root@serverX ~]# fdisk -cu /dev/vda
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (9914368-12582911, default 9914368): Enter
Using default value 9914368
Last sector, +sectors or +size{K,M,G} (9914368-12582911, default 12582911): +1G

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@serverX ~]# reboot
```

vda1 can only have 4 primary
partitions
sda1 extended disk container for
multiple logical
partitions inside.

File System Comparison

- **ext4** is the standard file system for Red Hat Enterprise Linux 6. It is very robust and reliable, and has many features to improve performance for modern workloads.
- **ext2** is an older file system commonly used in Linux; it is simple and reliable, and works well for small storage devices, but is not as efficient as **ext4**.
- **vfat** support covers a family of related file systems (VFAT/FAT16, FAT32) developed for older versions of Microsoft Windows and supported on a wide variety of systems and devices.

Creating and Using a New File System

1. **mkfs -t filesystem /dev/partition** creates the type of file system requested.
2. **blkid** displays information about the contents of block devices (partitions and logical volumes) including the UUID of the file system.
3. **mkdir /mountpoint** creates a directory to link the new file system to.
4. Add an entry to **/etc/fstab** using the obtained UUID from the **blkid** command:

UUID=uuid /mountpoint ext4 defaults 1 (2) check order

man fstab

5. Mount the new file system with **mount /mountpoint**.

mount -o rw, remount /

if there's a
problem at
boot time



Warning

When adding new file systems to **/etc/fstab**, you should use **blkid** to determine its' UUID and mount by UUID. You should *not* mount file systems on simple partitions by standard device name (such as **/dev/sda3**). Disk device names may change depending on the devices visible at boot time, which may cause your system to attempt to mount the wrong file system for the wrong purpose, which at worst could lead to data loss. This is especially important when SAN devices (iSCSI, Fiber Channel) are involved which may be detected by the system in a different order from boot to boot depending on SAN traffic, but it can also matter when removable media such as USB devices may be in use.

Note that Red Hat Enterprise Linux 6 uses UUID instead of LABEL in **/etc/fstab** to reduce the likelihood of naming collisions. The installer no longer uses **e2label** to set labels on Red Hat Enterprise Linux 6 file systems by default.

Example of File System Creation

```
[root@serverX ~]# mkfs -t ext4 /dev/vda3
[root@serverX ~]# blkid /dev/vda3
/dev/vda3: UUID="a11fadb0-2f5b-49e8-ba43-13de7990d3b9" TYPE="ext4"
[root@serverX ~]# mkdir /test
```

Add an entry to **/etc/fstab**:

```
UUID="a11fadb0-2f5b-49e8-ba43-13de7990d3b9" /test ext4 defaults 1 2
```

Test the mount:

```
[root@serverX ~]# mount /test
```

Remove an Existing File System

1. Unmount the file system by using **umount /mountpoint**.
2. Remove the corresponding entry in **/etc/fstab**.
3. Remove the mount point directory: **rmdir /mountpoint**.



References

fdisk(8), fstab(5), mkfs(8), blkid(8), mount(8) man pages

Knowledgebase: "How can I create a disk partition on a disk that is greater than 2 TB in size?"
<https://access.redhat.com/kb/docs/DOC-4282>



Practice Quiz

Add a New File System

1. Identify a disk that has some free space

fdisk -cu

2. Create a new partition on that disk

fdisk -cu /dev/vda

3. Update the kernel partition table

reboot

4. Create a file system on the partition

mkfs -t ext4 /dev/vda3

5. Determine the UUID of the file system

blkid

6. Create a mount point

mkdir /data

7. Add an entry to the file system table file

vi /etc/fstab

8. Mount the file system

mount /data

Enabling Data Privacy with Partition Encryption

LUKS ("Linux Unified Key Setup") is a standard format for device encryption. LUKS encrypts the partition or volume; the volume must be decrypted before the file system in it can be mounted.

Create a New Encrypted Volume

1. Create a new partition with **fdisk**
2. **cryptsetup luksFormat /dev/vdaN** encrypts the new partition and sets the decryption password
3. **cryptsetup luksOpen /dev/vdaN name** unlocks the encrypted volume **/dev/vdaN** as **/dev/mapper/name** after you enter the correct decryption password
4. Create an **ext4** file system on the decrypted volume: **mkfs -t ext4 /dev/mapper/name**
5. Create the directory mountpoint and mount the file system: **mkdir /secret ; mount /dev/mapper/name /secret**
6. When finished, **umount /dev/mapper/name** and run **cryptsetup luksClose name** to lock the encrypted volume



Note

When creating an encrypted partition, it is best to write random data to the raw device before using LUKS to initialize the volume. This may help make attacks on the disk encryption more difficult. Copying data from **/dev/urandom** to the device file for the partition will accomplish this, but it can take a long time.

dd if=/dev/urandom of=/dev/vda3

Persistently Mount Encrypted Partition

1. **/etc/crypttab** contains a list of devices to be unlocked during system startup.

❶name **❷**/dev/vdaN **❸**/path/to/password/file

/etc/crypttab lists one device per line, with the following space separated fields:

- ❶** Name device mapper will use for the device
- ❷** The underlying "locked" device
- ❸** Password file to use to unlock the device. If this field is left blank (or set to **none**), the user will be prompted for the decryption password during startup

2. Create an entry in **/etc/fstab** like the following:

/dev/mapper/name /secret ext4 defaults 1 2



Warning

The device listed in the first field of **/etc/fstab** must match the name chosen for the local name to map in **/etc/crypttab**. This is a common configuration error.

3. Create the key file that includes the password. Make sure it is owned by root and the mode is 600. Add the key for LUKS using the following command:

```
[root@serverX ~]# cryptsetup luksAddKey /dev/vdaN /path/to/password/file
```

Encrypted File System Creation Example

Create a new partition as previously done. We will assume the device is **/dev/vda5**.

```
[root@serverX ~]# cryptsetup luksFormat /dev/vda5
WARNING!
=====
This will overwrite data on /dev/vda5 irreversibly.

Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: testing123
Verify passphrase: testing123
[root@serverX ~]# cryptsetup luksOpen /dev/vda5 encdisk
Enter passphrase for /dev/vda5: testing123
[root@serverX ~]# mkfs -t ext4 /dev/mapper/encdisk
[root@serverX ~]# mkdir /encdisk
[root@serverX ~]# mount /dev/mapper/encdisk /encdisk
```

To make the disk persistent, start by appending the following to **/etc/fstab**:

```
/dev/mapper/encdisk /encdisk ext4 defaults 1 2
```

Create **/etc/crypttab** and add the following line. This will ask the password every time the machine boots:

```
encdisk /dev/vda5
```

Automatic Entry of Encryption Password

If you want an automated boot, you must place the password in a text file (which has obvious security implications).

/etc/crypttab:

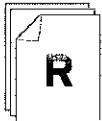
```
encdisk /dev/vda5 /root/encdisk
```

```
[root@serverX ~]# echo -n "testing123" > /root/encdisk
[root@serverX ~]# chown root /root/encdisk
[root@serverX ~]# chmod 600 /root/encdisk
```

Uim /etc/crypttab

change from none to /etc/crypttab

```
[root@serverX ~]# cryptsetup luksAddKey /dev/vda5 /root/encdisk
Enter any passphrase: testing123
```



References

Red Hat Enterprise Linux Security Guide

- Section 3.8: LUKS Disk Encryption

cryptsetup(8) and **crypttab(5)** man pages



Practice Resequencing Exercise

Create Encrypted File System

For each of the file or directory names below, write down the number of its definition from the list at the bottom.

- 1 Create a new partition
- 4 Create an **ext4** file system
- 2 Format the new partition for encryption
- 6 Mount the file system on the unlocked device
- 8 Create an entry in **/etc/fstab**
- 3 Create a directory to use as a mount point
- 5 Unlock the encrypted partition
- 7 Create an entry in **/etc/crypttab**
- 9 Make LUKS aware of the password file

1. **fdisk**
2. **cryptsetup luksFormat /dev/vdaN**
3. **cryptsetup luksOpen /dev/vdaN secret**
4. **mkfs -t ext4 /dev/mapper/secret**
5. **mkdir /secret**
6. **mount /dev/mapper/secret /secret**
7. **secret /dev/vdaN /password/file**
8. **/dev/mapper/secret /secret ext4 defaults 1 2**
9. **cryptsetup luksAddKey /dev/vdaN /password/file**

Managing Swap Space

Swap space or a swap area is space on the disk drive used as overflow for parts of memory that are not currently being used. This allows the system to make room in main memory for data currently being processed, and provides emergency overflow if the system is at risk of running out of space in main memory.

Creating and Using an Additional Swap Partition

1. Create a new partition using **fdisk**. Additionally, change the partition type to "**0x82 Linux Swap**" before saving changes with **fdisk**.
2. **mkswap /dev/vdaN** will prepare the partition for use as a swap area.
3. **blkid /dev/vdaN** will determine the UUID.
4. Add the new swap space to **/etc/fstab**:

```
UUID=uuid swap swap defaults 0 0
```

5. **swapon -a** will activate the new swap area.

swapon -s will show status of current swap areas.

swapoff /dev/vdaN will de-activate that particular swap area.

Example of Swap Space Creation

Create a new partition and change the type to 82:

```
[root@serverX ~]# fdisk /dev/vda
Command (m for help): n
First sector (12539904-12582911, default 12539904): Enter
Using default value 12539904
Last sector, +sectors or +size{K,M,G} (12539904-12582911, default 12582911): Enter
Using default value 12582911

Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): 82
Changed system type of partition 6 to 82 (Linux swap / Solaris)

Command (m for help): w

[root@serverX ~]# reboot
```

Write the swap signature to the device and find the UUID:

```
[root@serverX ~]# mkswap /dev/vda6
[root@serverX ~]# blkid /dev/vda6
/dev/vda6: UUID="4903c440-ffcb-4404-bc09-505c79c7a412" TYPE="swap"
```

Add an entry to **/etc/fstab**:

```
UUID="4903c440-ffcb-4404-bc09-505c79c7a412" swap swap defaults 0 0
```

Activate the swap space, verify it is available and then deactivate the swap space:

```
[root@serverX ~]# swapon -a
swapon -s
/dev/dm-0          partition      557048  0      -1
/dev/vda6          partition     21496   0      -2
[root@serverX ~]# swapoff /dev/vda6
```

Sizing total swap space should really be based on the memory workload on the system, not the total amount of physical memory present. However, the table below provides some rough rules of thumb for sizing swap space. For more detailed guidance on sizing swap space, see the Knowledgebase article in the references.

| System RAM | Recommended Minimum Swap Space |
|-------------------|---------------------------------------|
| up to 4 GB | at least 2 GB |
| 4 GB to 16 GB | at least 4 GB |
| 16 GB to 64 GB | at least 8 GB |
| 64 GB to 256 GB | at least 16 GB |

Table 10.1. Basic Guidance on Swap Space Sizing

make swap at least as big as RAM if
you want to hibernate



References

Red Hat Enterprise Linux Storage Administration Guide

- Chapter 14: Swap Space

Knowledgebase: "If I add several hundred GB of RAM to a system, do I really need several hundred GB of swap space?"
<https://access.redhat.com/kb/docs/DOC-15252>

mkswap(8) and **swapon(8)** man pages

Can use a file as swap as well

do if = /dev/zero of = /swapfile bs=1M count = 250

mkswap is swapfile

swapon -a /swapfile

edit /etc/fstab



Practice Exercise

Create and Use a New Swap Partition

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Create and use a new 256 MB swap partition on your virtual server, serverX.

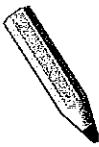
1. Start **fdisk** and create a new partition



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand

2. Change the partition type to **swap**.
3. Prepare the new partition for use as swap
4. Determine the UUID
5. Add the new partition to **/etc/fstab**
6. Determine current amount of swap
7. Activate the new swap
8. Verify newly activated swap



Test

Criterion Test

Case Study

Managing Simple Partitions and File Systems

Before you begin...

Make sure to run the **lab-setup-storage** from your desktopX system, which will prepare your serverX system for the lab.

When you are ready, run the **lab-grade-storage** script on serverX to check your work.

Your department wants to use some unallocated storage on the servers. Create additions to your system according to the following list:

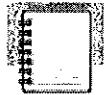
- Create a new partition and **ext4** file system that is 400 MB in size. The file system should persistently mount under **/data**.
- Persistently add a swap partition that is 200 MB in size.
- Create an encrypted device with an **ext4** file system that is 256 MB in size and uses the password **testing123**. The system should prompt for the password at boot and mount the file system to **/test**.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Simple Partitions and File Systems

In this section you learned how to:

- Create and format a simple partition for data storage

Enabling Data Privacy with Partition Encryption

In this section you learned how to:

- Enable data privacy with an encrypted partition from the command-line

Managing Swap Space

In this section you learned how to:

- Create and format a simple partition for swap



UNIT ELEVEN

CONTROLLING ACCESS TO FILES

Introduction

Topics covered in this unit:

- POSIX Access Control Lists for file permissions

Managing File System Access Control Lists

Access Control List Support

- Standard Linux file systems (ext2/3/4) support more complex file permissions to be set by using POSIX ACLs, provided the file system is mounted with the **acl** option.
- In Red Hat Enterprise Linux, if the last character of the permission string displayed by **ls -l** is a **+**, the file or directory has an ACL set.
- getfacl file** is used to display ACLs on a file

```
u:elvis:rwx      # applies to user elvis
u:3142:---      # applies to user id 3142
u::rwx          # applies to file user owner

g:music:rwx     # applies to group music
g:10:r-x        # applies to group id 11
g::rw-          # applies to file group owner

o::rwx          # applies to everyone else
```

- setfacl** is used to set or modify ACLs on a file

```
setfacl -m u:friend:rwx filename # grants rw to user friend
setfacl -m g:grads:rwx filename # grants rw to the group grads
setfacl -m g:profs:r filename   # grants r to the group profs

setfacl -x u:friend           # removes the existing ACL for friend
setfacl -m o::-- filename     # changes normal "other" permissions
```

Use this space for notes

Permission Precedence

When determining whether a process (that is, a running program) can access a file, file permissions and ACLs are applied as follows:

- If the process is running as the user that owns the file, then the file's *user* permissions apply
- Else, if the process is running as a user that is listed in a user ACL entry, then the *user ACL* applies (as long as it is permitted by the **mask**)
- Else, if the process is running as a group that matches the group that owns the file or as a group with an explicit group ACL entry, if the permission is granted by *any* matching group it applies (as long as it is permitted by the **mask**)
- Otherwise, the file's *other* permissions apply

The ACL Mask

- Files that have ACLs have a "mask" which limits the maximum permissions that both the group that owns the file, and that supplementary users and groups in ACLs, can have.
- **getfacl file** shows the current mask as **mask::permissions**.
- The group permissions displayed by **ls -ld file** also reflect the current mask (*not* the owning group's permissions!)



Important

Changing group permissions on a file with an ACL by using **chmod** does not change the owning group's actual permissions! It actually changes the *mask*, which limits the maximum permissions of all groups and supplementary users that have access to the file. (That is, the permissions for everyone but the owning *user* and users in the *other* category for the file.)

Use this space for notes

Default ACLs (Inheritance)

- Directories can have "default ACL" entries which are automatically set on new files created in that directory
- **setfacl -m d::u:elvis:rw directory** would set a default ACL entry granting read-write access to user **elvis** on all new files created in **directory**.
- This is similar to the way that the setgid permission, when set on a directory, causes new files created in that directory to be owned by the same group that owns the directory.



Note

When setting default ACLs on a directory, if you want to ensure that users will be able to access the contents of new subdirectories created in it, make sure you include executable permissions on their ACL:

```
[user@host ~]$ setfacl -m d::u:elvis:rx directory
```

Normally, users will not automatically get executable on newly created regular files because unlike new directories, the ACL *mask* of a new regular file is **rw-** by default.

Use this space for notes

ACL Mount Option

- Support for POSIX ACL entries must be enabled when the file system is mounted.
- The installer configures all **ext4** file systems it creates to automatically turn on ACL support.

```
[root@host ~]# dumpe2fs /dev/block-dev | grep 'Default mount'  
Default mount options: user_xattr acl
```

- If you manually formatted the file system, you will need to mount it with the **acl** mount option.
- You can set a manually-formatted **ext4** file system to turn on support at mount automatically by using **tune2fs** to set default mount options:

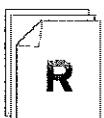
```
[root@host ~]# tune2fs -o acl,user_xattr /dev/block-dev
```



Note

The **user_xattr** mount option does not have anything to do with POSIX ACLs, but enables *user extended attributes*. These are used by a handful of programs to store custom information with files. It is a good idea to leave this default option set even though it is not needed to enable POSIX ACLs. See **attr(5)** for more information.

Use this space for notes



References

Red Hat Enterprise Linux Storage Administration Guide

- Chapter 16: Access Control Lists

acl(5), getfac1(1), and setfac1(1) man pages

Collaborative Directory Permissions

In the quiz below, use both POSIX ACLs and standard permissions, as appropriate, to solve these problems.

1. Given a normal directory, where the owning *user* has **rwx** permissions, the owning *group* has **rwx** permissions, and *other* has **---** permissions, what command would grant a second group **r-x** permissions without changing the permissions of the existing owning group or *other*?

2. What command would automatically grant that second group read-write access to any newly created regular files in that directory?

3. *Bonus question.* What command would automatically set the owning *group* as the owning group of any newly created files in that directory?

setgid
chmod g+rw _____
chmod g+sw _____

Test

Criterion Test

Case Study

Using ACLs to Grant and Limit Access

This lab uses users and groups created earlier on serverX. If you do not already have the users and groups defined, run **lab-add-users** on serverX.

Graduate students need a collaborative directory titled **/opt/research**, where they can store generated research results. Only members of the groups **profs** and **grads** should be able to create new files in the directory, and new files should have the following properties:

- The directory should be owned by user **root**.
- New files should be group owned by the group **grads**.
- Professors (members of the group **profs**) should automatically have read/write access to new files.
- Summer interns (members of the group **interns**) should automatically have read-only access to new files.
- Other users (not a member of groups **profs**, **grads**, or **interns**) should not be able to access the directory and its contents at all.

See the Solutions appendix when you are done to check your solution and to see some possible approaches.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Managing File System Access Control Lists

In this section you learned how to:

- Use an ACL entry to grant or block file access
- List the ACLs on a file
- Delete an ACL entry
- Assign an ACL or group ownership to new files created in a directory automatically



UNIT TWELVE

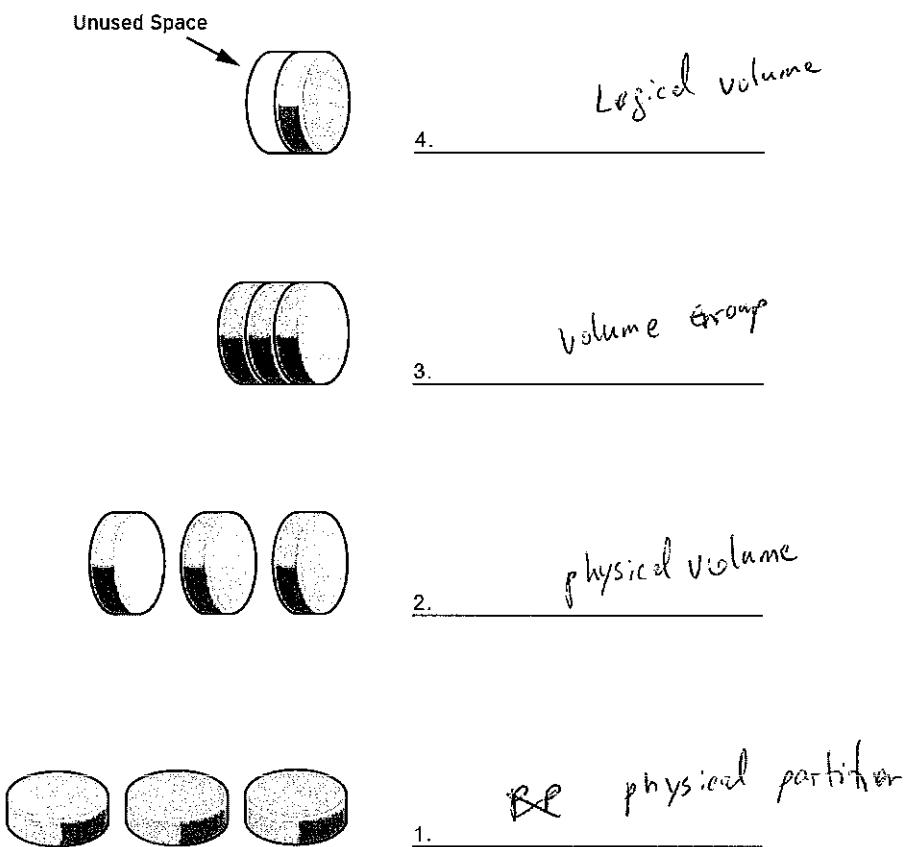
MANAGING FLEXIBLE STORAGE WITH LOGICAL VOLUME MANAGER

Introduction

Topics covered in this unit:

- Review LVM Components
- Implement LVM Storage
- Grow a File System
- Add a Disk
- Snapshot as Backup

Recognize the Components of LVM



Review LVM Definitions

- *Physical Partitions or Disks* are the first building block of LVM. These could be partitions, whole disks, RAID sets or SAN disks.
- *Physical Volumes* are the underlying "physical" storage used with LVM. This is typically a block device such as a partition or whole disk. A device must be initialized as an LVM Physical Volume in order to be used with LVM.
- *Volume Groups* are storage pools made up of one or more Physical Volumes.
- *Physical Extents* are small chunks of data stored on Physical Volumes that act as the back end of LVM storage.
- *Logical Extents* map to Physical Extents to make up the front end of LVM storage. By default, each Logical Extent will map to one Physical Extent. Enabling some options will change this mapping. Mirroring, for example, causes each Logical Extent to map to two Physical Extents.

- *Logical Volumes* are groups of Logical Extents. A Logical Volume may be used in the same manner as a hard drive partition.

Why Use Logical Volumes?

Logical volumes, and logical volume management make it easier to manage disk space. If a file system needs more space, it can be allocated to its logical volume from the free space in its volume group and the file system can be resized. If a disk starts to fail, a replacement disk can be registered as a physical volume with the volume group and the logical volume's extents can be migrated to the new disk.



References

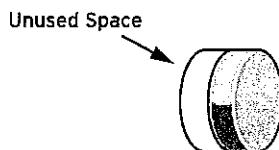
Red Hat Enterprise Linux Logical Volume Manager Administration



Practice Quiz

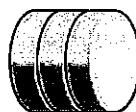
LVM Components

1. Fill in the following graphic with the names of the components.



4.

LV



3.

VG



2.

PV



1.

PF

2. What are the smallest pieces (chunks or blocks) of the physical volume?

Block phy ext

3. What is the smallest size you could make a logical volume?

1 ext

4. What references the physical extents of a logical volume?

logical ext

Implement LVM Storage with Command-line Tools

Prepare a Physical Volume

1. **fdisk** is used to create a new partition for use with LVM. Always set the Type to **Linux LVM** on a partition to be used with LVM.



Note

Alternatively, you can use a whole disk, a RAID array, or a SAN disk.

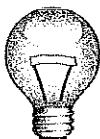
2. **pvcreate /dev/vdaN** is used to initialize the partition (or other physical device) for use with LVM as a Physical Volume. A header to store LVM configuration data is created directly in the Physical Volume.

Creating a Volume Group

1. **vgcreate vgname /dev/vdaN** will create a volume group named *vgname* made up of the physical volume */dev/vdaN*. You can specify additional space-delimited physical volumes at the time of creation or add new physical volumes later with **vgextend**.

Create and Use a New Logical Volume

1. **lvcreate -n lvname -L 2G vgname** creates a new 2 GB logical volume named *lvname* from the available physical extents on *vgname*.



Important

Different tools will display the logical volume name using either the traditional name, **/dev/vgname/lvname**, or the kernel device mapper name, **/dev/mapper/vgname-lvname**.

2. **mkfs -t ext4 /dev/vgname/lvname** will create an **ext4** file system on the new logical volume.
3. **mkdir /data** makes directory needed as a mount point.
4. Add an entry to the **/etc/fstab** file:

```
/dev/mapper/vgname-lvname /data ext4 defaults 1 2
```

5. Run **mount -a** to mount all the file systems in **/etc/fstab**, including the entry you just added.

Review LVM Status Information

1. **pvdisplay /dev/vdaN** will display information about the specific physical volume.
2. **vgdisplay vgname** will display information about the specific volume group.

Example **vgdisplay** output:

```
--- Volume group ---
VG Name ①          vg1
System ID
Format    lvm2
Metadata Areas   10
Metadata Sequence No 18
VG Access      read/write
VG Status       resizable
MAX LV         0
Cur LV          6
Open LV         6
Max PV          0
Cur PV          5
Act PV          5
VG Size ②        7.28 TB
PE Size ③        4.00 MB
Total PE ④       1907727
Alloc PE / Size ⑤ 1720587 / 6.56 TB
Free  PE / Size ⑥ 187140 / 731.02 GB
VG UUID        7FmScA-HJWa-<snip>
```

- ① The name of the Volume Group
- ② The total size of the “physical” storage in the Volume Group
- ③ Physical Extent size
- ④ Total Physical Extents in Volume Group
- ⑤ Total Physical Extents used by Logical Volumes
- ⑥ Physical Extents available

3. **lvdisplay /dev/vgname/lvname** will display information about the specific logical volume.



References

Red Hat Enterprise Linux Logical Volume Manager Administration

lvm(8) man page



Practice Exercise

Implement LVM and Create a Logical Volume

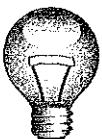
Before you begin...

Make sure to run the **lab-setup-lvm** from your desktopX system, which will prepare your serverX system for the practice exercise.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of these steps will be performed on serverX.

1. Create a new partition of 512 MB and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

2. Create a Volume Group named **shazam** using the Physical Volume created in the previous step.
3. Create and format with **ext4**, a new Logical Volume of 256 MB called **/dev/shazam/storage**.
4. Modify your system such that **/dev/shazam/storage** is mounted at boot time as **/storage**.

Extend a Logical Volume and Ext4 File System

One benefit of logical volumes is the ability to increase their size without experiencing downtime. Free physical extents in a volume group can be added to a logical volume to "extend" its capacity, which can then be used to extend the file system it contains.

Growing a Logical Volume Basic Steps

1. Verify available space in the
Volume group
2. Extend the
Logical volume
3. Extend the
file system

Extending the Logical Volume and File System

1. Verify the current size of the mounted file system `/data`:

```
# df -h /data
```

2. Verify that there are sufficient "Physical Extents available" for use:

```
# vgdisplay vgname
```

Refer to the sample output in the "Review LVM Status Information" section found earlier in this unit to see how to identify available free physical extents.

3. Extend the logical volume using some or all of the available extents:

```
# lvextend -l 128 /dev/vgname/lvname
```

4. Grow the associated file system mounted on `/data`:

```
# resize2fs -p /dev/vgname/lvname
```

The `-p` option displays progress during the operation.



Note

The file system can remain mounted and be used while `resize2fs` is being run.



Important

A common mistake is to run **lvextend** but forget to run **resize2fs**.

5. Verify the new size of the mounted file system **/data**:

```
# df -h /data
```

Reducing a File System and Logical Volume

This process is similar to extending, but *in reverse*: **resize2fs**, then **lvreduce**.



Warning

It is essential you have a solid backup before undertaking a reduction in the logical volume, as typographical errors in the command line can cause data loss.

1. While extending a logical volume can be done while the file system is in use, reducing an **ext4** file system must be done offline.
umount /data to unmount the file system you want shrink.
2. **fsck -f /dev/mapper/vgname-lvname** to verify that all file system data structures are clean prior to resizing.
3. **resize2fs -p /dev/mapper/vgname-lvname 512M** will resize the file system to be 512 MB, presuming that the logical volume is larger than 512 MB.
Note: If you omit the size from the **resize2fs** command, it defaults to the size of the logical volume, perfect for extending the logical volume like done previously.
4. **lvreduce -L 512M /dev/mapper/vgname-lvname** will shrink the logical volume to 512 MB.



Warning

lvreduce has no knowledge of your file system data structures, and, without warning, will discard elements of your file system if you did not first use **resize2fs** to make the file system *smaller* than the intended logical volume size.

5. **mount -a** will remount all the file systems listed in **/etc/fstab**, including your now smaller logical volume assuming it is listed in



References

Red Hat Enterprise Linux Logical Volume Manager Administration

lvm(8) man page



Practice Exercise

Extend a Logical Volume

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of these steps will be performed on serverX.

1. Determine the amount of free space in Volume Group **shazam**.
2. Extend the logical volume **/dev/shazam/storage** with *half* the available extents in the volume group using command-line tools.
3. Extend the file system mounted on **/storage** using command-line tools.

Extending and Reducing a Volume Group

When the logical volumes in a volume group use all of the volume group's free physical extents, they cannot be extended without adding additional space to the volume group. Thankfully additional physical volumes can be created and added to a volume group to "extend" its capacity.

Another benefit of using LVM is that data can be moved between physical storage devices without user downtime. For example, data can be moved from a slower disk drive to a new, faster disk drive. This allows a system administrator to remove the unused physical storage device from a volume group, in this case the slow disk drive.

Extending a Volume Group

- As in the creating a new volume group, a new partition must be created and prepared for use as an LVM Physical Volume.

Use **fdisk** to create a new partition and take care to set the Type to **0x8e Linux LVM**.

Use **pvccreate /dev/vdaN** to initialize the partition for use with LVM as a Physical Volume.

- vgextend vgname /dev/vdaN** is used to add the new Physical Volume, **/dev/vdaN**, to an existing Volume Group, **vgname**.
- Use **vgdisplay** to confirm additional "Physical Extents available".

Reducing a Volume Group

- pvmmove /dev/vdaN** is used to relocate any physical extents used on **/dev/vdaN** to other Physical Volumes in the Volume Group. This is only possible if there are enough available extents in the Volume Group and if all of those come from other Physical Volumes.



Warning

Before using **pvmmove**, it is recommended to back up data on logical volumes in the volume group. An unexpected power loss during the operation may leave the volume group in an inconsistent state.

- vgreduce vgname /dev/vdaN** is used to remove the Physical Volume **/dev/vdaN** from the Volume Group **vgname**.



References

Red Hat Enterprise Linux Logical Volume Manager Administration

lvm(8), **pvmmove(8)**, and **vgreduce(8)** man pages



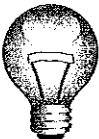
Practice Exercise

Extend a Volume Group

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of these steps will be performed on serverX.

1. Create a new 512 MB partition and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

2. Extend the Volume Group **shazam** by adding the Physical Volume created in the previous step.

Create a Snapshot to Facilitate Data Backup

Snapshot logical volumes are another flexible feature of LVM storage. An LVM snapshot is a logical volume that temporarily preserves the original data of a changing logical volume. The snapshot provides a static view of the original volume so its data can be backed up in a consistent state.

Determining Snapshot Size

1. Expected rate of _____
2. Required snapshot _____

The snapshot volume need only be large enough to store the data that will change while it exists.

If more data changes than the snapshot can hold, the snapshot will automatically become unusable. (The original volume will remain unharmed, and the dead snapshot will still need to be unmounted and removed from the volume group manually.)

Creating and Using a Snapshot for Backup

1. Create a new snapshot volume called *snaplvname* of */dev/vgname/lvname* that is **20 MB** in size.

```
# lvcreate -s -n snaplv -L 20M /dev/vgname/lvname
```

2. If your backup software requires it, mount the snapshot and point the backup program to the new mountpoint:

```
# mkdir /snapmount  
# mount -o ro /dev/vgname/snaplv /snapmount
```

3. Verify the status of the snapshot logical volume:

```
# lvs /dev/vgname/snaplv  
LV      VG      Attr   LSize   Origin Snap%  Move Log Copy%  Convert  
snaplv vgname swi-a- 224.00m origlv  0.26
```

4. When done with the snapshot, unmount and remove it:

```
# umount /snapmount  
# lvremove /dev/vgname/snaplv
```



References

Red Hat Enterprise Linux Logical Volume Manager Administration

lvm(8) man page

system-config-lvm

**Practice Exercise**

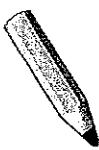
Creating an LVM Snapshot

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Compare the contents of our existing logical volume, **/dev/shazam/storage**, to a new snapshot volume, **/dev/shazam/storagesnap**, while making changes to the original volume.

All of these steps will be performed on serverX.

1. Copy the file **/usr/share/dict/linux.words** to **/storage** so you have some data to compare.
2. Create a new 20 MB snapshot logical volume of **/dev/shazam/storage** called **storagesnap**.
3. Manually mount **/dev/shazam/storagesnap** read only at **/storagesnap**
4. List the contents of **/storagesnap** and note that they are the same as **/storage**.
5. Delete the file **/storage/linux.words** and note that it still exists in **/storagesnap**.
6. Clean up: unmount **/storagesnap**, remove the directory, and delete the **storagesnap** logical volume.



Test

Criterion Test

Case Study

LVM Case Study

Before you begin...

Make sure to run the **lab-setup-lvm** from your desktopX system, which will prepare your serverX system for the lab.

When you are ready, run the **lab-grade-lvm** script on serverX to check your work.

Allison needs to store data for her business. Her customer database is currently 256 MB in size. The data in the database changes about 10 MB per hour on a typical day. The backup software takes 10 minutes to complete a full run.

Create a new Volume Group called **allison** with enough space for both a 512 MB volume and a snapshot of that volume for the backup software.

Once the volume group is created, create within it a 512 MB logical volume for Allison's customer database called **custdb**. Also create a snapshot volume of Allison's customer database called **custdbsnap** for her backup software.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Recognize the Components of LVM

In this section you learned how to:

- Identify the basic building blocks of Logical Volume Manager

Implement LVM Storage with Command-line Tools

In this section you learned how to:

- Create new physical volumes, volume groups and logical volumes via command-line tools
- Review LVM status information

Extend a Logical Volume and Ext4 File System

In this section you learned how to:

- Extend a logical volume and corresponding file system to satisfy growing data needs

Extending and Reducing a Volume Group

In this section you learned how to:

- Add new physical volumes to an existing volume group
- Remove an existing physical volume from a volume group

Create a Snapshot to Facilitate Data Backup

In this section you learned how to:

- Use temporary LVM snapshots to facilitate data backups, minimizing service downtime



UNIT THIRTEEN

CONTROL THE BOOT PROCESS

Introduction

Topics covered in this unit:

- Boot an alternate kernel
- Boot into a specific runlevel
- Overcome bootloader misconfigurations
- /boot/grub/grub.conf
- Kernel boot parameters
- /etc/inittab
- Recovery Shells

/etc/inittab

Booting an Alternate Kernel

The heart of the Linux operating system is the *kernel*, which acts as the interface between user code and system hardware. From time to time, a newer version of the kernel for Red Hat Enterprise Linux is released, which may enable new features or fix software bugs.

In order to use a new kernel, the system must be rebooted. Normally, the newest version of the kernel installed on the system is used. However, Red Hat Enterprise Linux allows multiple kernel versions to be installed at the same time. This allows you to test a kernel update, and if there is a critical regression or other problem with the update, you can easily fall back to a kernel that is known to work for your system.

In this section, we will look at how to manually select what kernel to boot when the system is started. Later, we will look at how you can make this selection permanent.

Write a definition for each of these key terms:

1. bootloader

2. GRUB

You can use the bootloader to:

- Boot into an older kernel if a new kernel is incompatible with your hardware
- Boot into single user mode when doing system maintenance or to get control of a machine with an unknown **root** password

Procedure To Boot an Alternate Kernel

1. Interrupt the GRUB countdown: **Esc** key
2. Use arrow keys to select alternate kernels
3. Hit **Enter** when the kernel you want to boot from is highlighted



References

Red Hat Enterprise Linux Installation Guide

- Appendix E: The GRUB Boot Loader



Practice Performance Checklist

Booting an Alternate Kernel

Perform all of the following steps on serverX.

- Configure **yum** to point to the **Errata** repository on the **instructor** machine with the following command:

```
[root@serverX ~]# wget http://instructor/pub/gls/errata.repo -O /etc/yum.repos.d/errata.repo
```

- Install the **kernel** update that is available. This will take over 3 minutes to install.
- Boot into the new kernel.
- Reboot and choose the old kernel.

Booting into a Different Runlevel

Runlevel Definitions

1. Write a definition for this key term:

runlevel

Who ->
uname -r

2. What are each of these runlevels typically used for?

runlevel 5 - Graphical desktop

runlevel 3 - _____

runlevel 1 - _____

Changing Runlevels

- Execute `init r1num` at the shell prompt, where `r1num` is the runlevel number. This will change the runlevel immediately.
- Pass the runlevel number as an argument to the kernel via GRUB at boot time. This will override the default runlevel.



References

Red Hat Enterprise Linux Installation Guide

- Technical Appendix E.8: Changing Runlevels at Boot Time



Practice Performance Checklist

Changing the root Password

This timed drill is designed to give you practice changing the root password on a system with an unknown root password.

Perform all of the following steps on serverX.

- Begin by running the **lab-setup-bootbreak-4** script. This will change the root password to something unknown and mark the current time.
- Get into the system and reset the root password to **redhat**.



Important

At the release of Red Hat Enterprise Linux 6, there was an SELinux bug which blocked the **passwd** command in single-user mode (#644820). If you have the original **selinux-policy** package installed, you must run the **setenforce 0** command in runlevel 1 before the **passwd** command for it to work. After changing the password you should run **setenforce 1** again to put SELinux back in enforcing mode.

- Once you have reset the password, change the system into runlevel 5 and run the **lab-grade-bootbreak-4** script.
- View the feedback from the script to ensure you completed the task correctly. The grading script will display a time, write it down.
- Repeat the process again at least five times.
- Circle your best time.

Resolve GRUB Issues

The GRand Unified Bootloader (GRUB) provides the bridge in the boot process between the hardware and the Linux kernel. When the system boots, the BIOS starts and normally loads GRUB in stages from the hard drive; from the first 446 bytes of the disk, then from the space between the first sector and the start of the first partition, then from files in **/boot**. GRUB then reads its configuration file, **/boot/grub/grub.conf**, which controls what operating systems and kernels are available to boot.

The GRUB Boot Screen

When GRUB starts up, a graphical splash screen can be accessed by pressing Return, Space or any other key. This screen has a list of menu entries, normally bootable images. You can select between the different images with the up and down arrow keys, and press Return to select a particular entry for booting. If you want to pass arguments to boot images through menu editing mode or access the GRUB command line, and a GRUB password is set, you will need to type "p" followed by your GRUB password.

Each menu entry which boots Red Hat Enterprise Linux typically has three GRUB directives:

- **root**, which indicates the location of the file system containing **/boot**
- **kernel**, which indicates the location of the kernel to boot relative to **root** and any command-line options or arguments to pass to the kernel
- **initrd**, which indicates the location of the "initial RAM disk" that is loaded early in the boot process by the kernel which contains critical device drivers needed at boot time

Temporary GRUB Correction

1. Interrupt the GRUB countdown: **Esc** key
2. Use "e" to edit current configuration
3. Select lines to correct with arrow keys
4. Type "e" again to edit the current line

↖ password --md5 md5 hash

Important

Typing **Esc** at this point takes you back to the menu, throwing your changes away.



5. Type "b" to boot with the current changes



References

- Red Hat Enterprise Linux Installation Guide
• Technical Appendix E.5: GRUB Interfaces

check kernel file version +
initrd ramfs version

cat /proc/cmdline is the current arguments passed to kernel
at this boot

info grub search for security

grub-md5-crypt

.....



Practice Performance Checklist

Getting Past a GRUB Misconfiguration

Perform all of the following steps on serverX.

- Run the **lab-setup-bootbreak-5** script to introduce an issue with the boot process.
- Fix the issue so the system can boot and you can log in.

Making Persistent GRUB Changes

The second stage of GRUB uses `/boot/grub/grub.conf` which has a format of global options followed by boot stanzas. Here is a sample `grub.conf` file:

```
[root@demo ~]# cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/mapper/vgsrv-root
#          initrd /initrd-[generic-]version.img
#boot=/dev/vda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux (2.6.32-71.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.el6.x86_64 ro root=/dev/mapper/vgsrv-root
    rd_LVM_LV=vgsrv/root rd_LVM_LV=vgsrv/swap rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
    SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us crashkernel=auto rhgb quiet
    initrd /initramfs-2.6.32-71.el6.x86_64.img
```

- Comment lines begin with a # character
- **default=number** - number is the default boot stanza (starting from 0)
- **timeout=number** - specifies how long the countdown occurs
- **hiddenmenu** - hides the menu display until a key is struck
- **rhgb quiet** - consider removing these kernel arguments to view more diagnostic information during boot



References

Red Hat Enterprise Linux Installation Guide

- Technical Appendix E.7: GRUB Menu Configuration File

Red Hat Enterprise Linux Deployment Guide

- Section 23.6: Verifying the Boot Loader

info grub



Practice Performance Checklist

Making Persistent GRUB Changes

Perform all of the following steps on serverX.

- Reboot and confirm the issue from the previous problem is not persistently fixed. You will need to apply the fix as before to boot the system.
- Edit the configuration file to fix the issue permanently.
- Revert to the older kernel. Ensure that when you reboot, the older kernel is the default kernel.

team 3

Passing Kernel Arguments

Kernel Arguments Search & Learn

1. Install the **kernel-doc** package.
2. Reference the material in **kernel-parameters.txt** found in the **/usr/share/doc/kernel-doc*/Documentation** directory.
3. Each team must research and summarize the following kernel parameters:

Team 1:

- console

specific to other console instead off HYP, then boot param
can be specify multiple times to go to other devices

Team 2:

- enforcing

change selinux enforcement mode audit-to-log

Team 3:

- init

from full path runs specific binaries instead of /sbin/init
can try emergency and just go into bash

Team 4:

- root

tells where is root file system

- ro

read-only

- rw

read-write



References

/usr/share/doc/kernel-doc-*/Documentation/kernel-parameters.txt



Practice Performance Checklist

Passing Kernel Arguments

Earlier we had to turn off SELinux enforcing mode to change the **root** password in runlevel 1. There is a kernel parameter that allows us to do that without using commands from the shell. Perform the following steps on serverX.

- Before you reboot your serverX machine, check its default SELinux status by executing the **getenforce** command. Confirm the system normally boots into Enforcing mode.
- Reboot your serverX machine and pass **enforcing=0** to the kernel when the system boots.
- Once serverX finishes booting, check its SELinux status. Confirm the system booted into Permissive mode.

Changing the Default Runlevel

The runlevel determines which services are started automatically on your Linux system. Most Linux desktop systems are set to boot to runlevel 5 (multi-user, networking, graphical interface). Many server systems boot to runlevel 3 (multi-user, networking, no graphical login), where the system comes up to a text-based interface.

The command **who -r** will return the runlevel the system is currently using, as will the right-hand number in the output of **runlevel**.

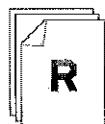
The default runlevel is read from the **/etc/inittab** file. For example, the line below would cause the system to boot to runlevel 5 by default.

```
id:5:initdefault:
```



Note

In Red Hat Enterprise Linux 6, the new **Upstart** boot system is configured to read the default runlevel from **/etc/inittab** for backward compatibility purposes. None of the other services formerly controlled from that file, including login prompts, can be set up in that file in RHEL 6. Those settings are kept in **/etc/init/** directory instead. For more information see the **init(8)** and **init(5)** man pages.



References

Red Hat Enterprise Linux Installation Guide

- Technical Appendix E.8: Changing Runlevels at Boot Time

Comments in **/etc/inittab**

startx to start x-window



Practice Performance Checklist

Changing the Default Runlevel

You are configuring a new system that you will be accessing remotely. The system is currently booting into runlevel 5 by default, but this machine will be housed in a data center where you will only log into it remotely. Perform the following steps on serverX.

- Change serverX to boot to runlevel 3 by default.
- Reboot serverX.
- You have successfully completed this lab if serverX boots into textual mode without human interaction.

Repairing Boot Issues

At boot time, if the system has difficulty correctly mounting file systems, the boot process may be interrupted to repair the problem. In this case, the system will automatically drop to a **sulogin** shell to allow repairs as root.

The **sulogin** shell is not the same as runlevel 1 or single-user mode. At this prompt, if any file systems are mounted they are mounted read-only and the system has not fully booted to any standard runlevel.

File systems most commonly fail to mount because the file system is in an inconsistent state due to a system crash. But there are other reasons they may fail to mount. For example, the system administrator may have changed file system configuration on the system and an error or typo in **/etc/fstab** may point to a device or file system that does not exist. Another possibility is that the system administrator has attempted to mount an LVM physical volume or an encrypted device by mistake, rather than the LVM logical volume or decrypted device containing the file system.

Repairing Damaged Filesystems

In normal operation, the kernel keeps frequently accessed file system information in memory, and only periodically commits the information to disk. If a file system becomes unexpectedly unavailable (such as due to a power outage or physical connectivity problem), the on-disk file system will contain inconsistencies. If these errors are not fixed, they are likely to lead to corrupt data.

The **fsck** command will attempt to restore a file system to a self-consistent state. The guarantee of **fsck** is not full data recovery, but consistency of the file system. On boot, the startup scripts will automatically **fsck** all file systems (if tagged in **/etc/fstab**). Minor problems will be resolved without interaction. If an automatic repair to a file system risks data loss, the boot process will drop to the **sulogin** shell to allow an administrator to run **fsck** interactively.

/etc/crypttab



Warning

Do not run **fsck** on a file system mounted read-write under any circumstances. This is likely to cause file system corruption and data loss.

1. Unmount the **/boot** file system on **/dev/vda1**.

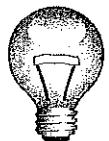
```
[root@demo ~]# umount /dev/vda1
```

2. Check the file system on **/dev/vda1**.

```
[root@demo ~]# fsck /dev/vda1
```

3. Remount the **/boot** file system.

```
[root@demo ~]# mount /dev/vda1
```



Important

While running, **fsck** may ask you whether or not certain changes should be made to the file system. Normally, you should answer *yes* to all questions, because **fsck** is asking for approval to complete a repair. You might answer *no* only if you are testing to see what errors **fsck** is finding or if you suspect the hardware is not responding to commands correctly, both rarer cases.

*Don't check on
extended or LVM
volume partition*



Warning

Running **fsck** can not be undone. Running **fsck** on something that does not contain a normal file system (like an LVM physical volume) may corrupt the contents of the device, causing data loss. Eliminate other possible causes of your problem before resorting to **fsck**.

If the file system is reporting errors because of a malfunctioning storage device, **fsck** will probably not be able to solve the issue, and may introduce further errors if the device is not performing correctly. (On the other hand, in that case your only other recourse are to restore from backup to a new storage device, or to send the malfunctioning device for data recovery—in which case you may not want to run **fsck** in order to avoid further risk of data loss.)

testdisk for better check than fsck



Workshop

Repairing Broken fstab with slogin Shell

Follow along with the instructor as you complete the steps below.

You will perform these steps on your virtual servers, first breaking the **/etc/fstab** file, experiencing the resulting symptoms (TEST), then look for misconfiguration (CHECK), and finally correct the problem (FIX).

1. Execute script, **lab-setup-bootbreak-1** on serverX to break **/etc/fstab** and reboot the server.

What happens?

dump to

2. Enter the root password to get to slogin shell
3. View the **/etc/fstab** file

What are the six fields?

Device, _____, File System Type, Mount Options, Dump Frequency, fsck Order

Hint: **fstab(5)** for details

4. Remount the root file system read-write

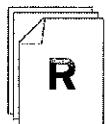
What command accomplishes this?

mount -o rw, remount /

5. Edit **/etc/fstab** to correct the error
6. Execute **mount -a** to confirm that no errors occur
7. Reboot the system and confirm that it boots normally

The slogin shell:

- System drops into an slogin shell if it encounters problems accessing a file system in **/etc/fstab**
- At the slogin shell, the root file system is mounted read-only.
- While the slogin shell might recommend it, running **fsck** is NOT always the preferred choice.



References

fsck(8), **fstab(5)**, and **sulogin(8)** man pages



Test

Criterion Test

Exercise

Bad Brian Blowup Recovery

Before you begin...

Run **lab-setup-bootbreak** on desktopX to reset serverX back to its original state.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Brian was a summer intern who acted as a system administrator for one of your critical servers, serverX. Your company's strained relationship with him finally blew up and resulted in his immediate firing. Sadly, when Bad Brian went out the door he took the root password for serverX with him.

You have been assigned the responsibility of getting control of serverX back:

1. Run the **lab-setup-bootbreak-6** script on serverX to prepare it for this lab exercise. This will assign your system with an unknown root password and reboot the system.
2. Set the root password to **redhat**.
3. Install the kernel update, but configure the system so the old kernel will continue to be used by default.
4. Pass the **selinux=1** argument to the kernel at boot time.
5. Set runlevel 3 as the default.
6. Once your system is booted, run the **lab-grade-bootbreak-6** script on serverX to determine how well you did.



Personal Notes



Unit Summary

Booting an Alternate Kernel

In this section you learned how to:

- Use the GRUB menu to select a different kernel to boot from

Booting into a Different Runlevel

In this section you learned how to:

- Use GRUB to boot the system into a specific runlevel

Resolve GRUB Issues

In this section you learned how to:

- Use GRUB to correct a broken GRUB configuration so the system will boot

Making Persistent GRUB Changes

In this section you learned how to:

- Persistently correct a GRUB misconfiguration
- Configure the system to boot from a different default kernel

Passing Kernel Arguments

In this section you learned how to:

- Pass additional kernel parameters to the booting kernel using GRUB

Changing the Default Runlevel

In this section you learned how to:

- Configure the system to boot into a specific runlevel

Repairing Boot Issues

In this section you learned how to:

- Fix problems with the boot process caused by a file system that will not mount properly
- Edit files from the read-only file system maintenance shell



UNIT FOURTEEN

TUNING AND MAINTAINING THE KERNEL

Introduction

Topics covered in this unit:

- Supported Architectures
- Kernel Modules
- Kernel Upgrades

Supported Architectures and Kernel Identification

As the interface between user programs and the system hardware, the kernel plays a key role in ensuring that Red Hat Enterprise Linux can be used on a wide variety of physical and virtual hardware environments.

Red Hat Enterprise Linux 6 Supported Architectures

Red Hat Enterprise Linux 6 is supported to run directly on four processor architectures:

- Intel and AMD 64-bit x86-64
- Intel and AMD 32-bit x86
- IBM POWER (64-bit POWER6 or later)
- IBM System z (System z9 or later)

This class is taught on machines using the Intel and AMD 64-bit x86-64 architecture (also known as *AMD64* and *Intel 64*).

Red Hat Enterprise Linux 6 and Virtualization

Red Hat Enterprise Linux 6 Supported as Virtualized Guest

Red Hat Enterprise Linux 6 is supported when running as a virtualized guest on the following hypervisors:

- KVM in Red Hat Enterprise Linux 5 and 6 (x86-64)
- Xen in Red Hat Enterprise Linux 5 (x86 and x86-64, paravirtualized and fully virtualized)
- VMware ESX Server and VMware ESXi Server
- Microsoft Windows Server 2008 Hyper-V

This means that Red Hat Enterprise Linux on these virtualization platforms is certified and supported independently of the underlying physical hardware (aside from any hardware pass-through). Please note that Red Hat supports the guest operating system. Direct support for a virtualization platform comes from the respective vendor. Please refer to <http://hardware.redhat.com/> for details on the verified products and versions, and to http://www.redhat.com/rhel/server/virtualization_support.html for more information.

Red Hat Enterprise Linux 6 Support in Hardware Partitions

Red Hat Enterprise Linux is also supported on hardware partitioning and virtualization solutions such as

- IBM POWER and System z
- Fujitsu PRIMEQUEST

These are certified for support with the physical hardware, not as a generic feature on non-certified hardware. See <http://hardware.redhat.com/> for details.

Red Hat Enterprise Linux 6 Support on the Public Cloud

For private in-house cloud computing deployments, standard support applies. Red Hat also certifies running Red Hat Enterprise Linux on the platform of various public cloud computing providers as part of our Cloud Partner Program. At the time of writing, Certified Cloud Providers included:

- Amazon EC2 (<http://www.redhat.com/solutions/cloud/amazon/>)
- IBM (<http://www.ibm.com/ibm/cloud/>)
- Savvis (<http://www.savvis.com/>)

System Limits

Supported system limits depend on architecture and product variant and version implemented. The URL <http://www.redhat.com/rhel/compare> is updated as new versions are released and new hardware is qualified.

Identifying the Running Kernel

1. `cat /etc/redhat-release` - installed Red Hat Enterprise Linux release
2. `uname -r` - Kernel version currently running
3. `yum list installed kernel*` - installed kernel versions
4. `uname -a` or `arch` - processor architecture currently running on

Occasionally, the kernel emits log messages. These messages are logged in the `/var/log/messages` file, labeled as the kernel service.



References

- Red Hat Server and Desktop Version Comparisons
<http://www.redhat.com/rhel/compare/>
- Virtualization Support in Red Hat Enterprise Linux
http://www.redhat.com/rhel/server/virtualization_support.html
- Red Hat Hardware Catalog
<http://hardware.redhat.com/>
- Cloud Partner Program
<http://www.redhat.com/solutions/cloud/partners/>
- uname(1) and arch(1) man pages**

Managing Kernel Modules

Kernel modules are object files, executable code that can be dynamically linked into a Linux kernel while it is running to extend its capabilities or provide device drivers. Dynamically loadable kernel modules are useful because they allow Linux to load only components of the kernel that are needed on a particular system in a particular configuration, saving memory space and system resource usage. They also allow the kernel to be extended without the need to recompile it and reboot the system.

Module Loading and Unloading

- The core kernel image, loaded at boot time, resides at **/boot/vmlinuz-VERSION**.
- While multiple kernels can be installed, only one is the current running kernel. In order to change kernels, the system has to be rebooted.
- Every kernel includes a collection of dynamically loaded modules compatible with that kernel, which are kept in **/lib/modules/VERSION/**.
- Generally, modules are loaded and unloaded on demand, with no user (or administrator) interactions.
- Currently loaded modules can be listed with **lsmod**. *modprobe -l*
- Occasionally, modules may need to be loaded manually with **modprobe MODULENAME**.
- Modules which are no longer in use can be removed with **modprobe -r MODULENAME**. *or from /usr/share/doc/Kernel ...*

Module Parameters

- Many modules accept parameters which can be specified as the module is loaded.
- The **modinfo** command reveals the parameters that a module supports.
- Parameters are specified as **name=value** pairs on the **modprobe** command line.

```
# modprobe encryptfs encryptfs_verbose=1
```

- Parameters can be applied automatically by configuring options in a **/etc/modprobe.d/local.conf** configuration file:

```
options encryptfs encryptfs_verbose=1
```

/etc/blacklist.conf

References

- Red Hat Enterprise Linux Deployment Guide
- Chapter 22: Working with Kernel Modules

lsmod(8) and **modprobe(8)** man pages

man modprobe.conf



Practice Exercise

Loading Modules and Setting Default Parameters

Carefully perform the following steps. Ask your instructor if you have problems or questions.

You have been asked to load the `nf_conntrack_ftp` kernel module, and configure it appropriately for an FTP server listening on TCP port 21 and on 8021.

The following commands are to be run on your serverX.

1. Use the locate command to convince yourself that the `nf_conntrack_ftp` module is supported by your kernel.
`modprobe -l | less`
2. Load the FTP connection track module.
`modprobe nf_conntrack_ftp`
3. Convince yourself it's loaded.
`lsmod | head`
4. Unload the FTP connection track module.
`modprobe -r nf_conntrack_ftp`
5. Convince yourself it's unloaded.
`lsmod | head`
6. In addition to the standard port 21, you are planning to run an FTP server on the non-standard port 8021. Examine options which might let you specify non-standard ports.
7. Configure a file `/etc/modprobe.d/local.conf` which implements this option upon loading
`ports=8021`

`modinfo nf_conntrack_ftp`

Upgrading Your Kernel

Fill in the blanks below as your instructor discusses the following topics. (Answers are also in the appendix.)

1. What (familiar) command performs a kernel update?
yum install kernel yum update kernel
2. New kernels are installed, not updated.
Because every file owned by the kernel package is versioned, or resides in a versioned directory, RPM is willing to have concurrent versions installed.
3. By default, when "updating" a kernel, **yum** will keep a total of 4 3 versions installed, automatically removing any older version.
/etc/yum.conf
4. In order to use your new kernel, you must reboot your machine.
5. While the machine will automatically reboot to your upgraded kernel, you may still choose an older kernel from the GRUB bootloader's menu.
6. If removing a kernel manually, you must specify not only the package name (`kernel`), but also the version and release.



Warning

Do not attempt to run `yum remove kernel` without further specifying *which* kernel package to remove from the system! The command will attempt to remove *all* kernel packages installed on the system as well as all packages which depend on `kernel`. This will result in a broken and unbootable system.



References

Red Hat Enterprise Linux Deployment Guide
• Chapter 23: Manually Upgrading the Kernel

lsmod(8) and **modprobe(8)** man pages

yum(1), **yum.conf(8)** man pages



Personal Notes



Unit Summary

Supported Architectures and Kernel Identification

In this section you learned how to:

- Discuss supported kernel architectures and limits
- Determine which kernel is running on the system

Managing Kernel Modules

In this section you learned how to:

- List current modules loaded in the kernel
- Load a module and its dependencies
- Remove a module
- Persistently configure a module to load with specified parameters
- Find documentation about a module

Upgrading Your Kernel

In this section you learned how to:

- Update a Red Hat compiled kernel



UNIT FIFTEEN

MANAGE VIRTUAL MACHINES

Introduction

Topics covered in this unit:

- KVM virtualization
- Virtual guest installation
- Autostart at boot

Introduction to KVM Virtualization

Virtualization is a feature that allows a single physical machine to be divided into multiple *virtual machines*, which can each run an independent operating system. Red Hat Enterprise Linux 6 for x86-64 supports *KVM*, which allows the kernel to function as a hypervisor supporting guest virtual machines, as long as certain requirements are met.

Facts about KVM virtualization:

- *Kernel-based Virtual Machine*: the virtualization system in Red Hat Enterprise Linux, built into the kernel as a module
- *VirtIO*: KVM supports *paravirtualized drivers* which can be used by KVM guests to obtain better IO performance

KVM benefits include:

- *Fast*: KVM is able to achieve high performance by taking advantage of x86-64 hardware virtualization support and by being closely integrated into the Linux kernel
- *Simple*: the design of KVM is simple, which makes it more robust, easier to support and optimize, and easier to use
- *Standard*: the KVM hypervisor is provided as a capability of the unmodified Linux kernel by the official "upstream" kernel team, which includes Red Hat engineers

KVM support requirements:

- *64-bit*: Red Hat supports KVM on 64-bit AMD or Intel processors running the x86-64 processor architecture
- *Extensions*: the 64-bit CPU, BIOS, and system hardware must also support the AMD Virtualization or Intel VT-x hardware-based virtualization extensions

To check whether a CPU claims to support hardware-assisted virtualization extensions, you can examine its *feature flags*. For example:

```
[user@host ~]$ grep flags /proc/cpuinfo
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
        dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx rdtscp lm constant_tsc arch_perfmon
        pebs bts rep_good xtopology nonstop_tsc aperfmpf perf_pni pclmulqdq dtes64 monitor ds_cpl
        vmx smx est tm2 ssse3 cx16 xtpr pdcm sse4_1 sse4_2 popcnt aes lahf_lm ida arat tpr_shadow
        vnmi flexpriority ept vpid
```

Relevant CPU feature flags include:

- **lm** = Long Mode (indicates 64-bit support)
- **svm** = Secure Virtual Machine (AMD basic virtualization support)
- **vmx** = Virtual Machine x86 (Intel basic virtualization support)

Only one of **svm** or **vmx** needs to (or is likely to) be present. Note the example above has the **lm** and **vmx** flags, so the CPU should support KVM.



Note

Red Hat Enterprise Linux 6 can not act as a Xen hypervisor, although it can run as a para-virtualized or fully-virtualized Xen guest on a RHEL 5 Xen host. See *Red Hat Enterprise Linux Virtualization* chapter 8, "Installing Red Hat Enterprise Linux 6 as a para-virtualized guest on Red Hat Enterprise Linux 5", for details.

Existing Xen guest machines from a Red Hat Enterprise Linux 5 host can be migrated to run as KVM guest machines on a Red Hat Enterprise Linux 6 host. See *Red Hat Enterprise Linux Virtualization* chapter 23, "Migrating to KVM from other hypervisors using virt-v2v", for details.

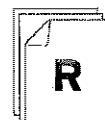


Important

There are two ways that the term *paravirtualization* is used in Linux virtualization which may lead to confusion.

In Red Hat Enterprise Linux 5, the Xen hypervisor supported *paravirtualized guests*. In this scenario, the drivers and kernel of the guests were modified to allow it to run on a Xen hypervisor running on a system that did not support full hardware virtualization extensions. This required that the operating system itself be modified to support Xen paravirtualized virtualization. KVM does not support paravirtualization in this sense.

KVM does support *paravirtualized drivers*. Paravirtualized drivers are special device drivers that can "cheat" by talking directly to the hypervisor. This removes the need for the guest to use a less efficient interface to the hypervisor that acts like some existing hardware device, like a disk controller or network card. These *virtio* paravirtualized drivers are faster than using normal drivers for the virtual hardware presented by KVM to the guest. Likewise, the operating system kernel does not need to be modified in order to take advantage of paravirtualized devices, you only need new drivers to be written which supports them.



References

Virtualization Support in Red Hat Enterprise Linux
http://www.redhat.com/rhel/server/virtualization_support.html

Virtualization Limits in Red Hat Enterprise Linux
<http://www.redhat.com/rhel/virtualization/compare/>

Red Hat Enterprise Linux Virtualization

- Part I: Requirements and Limitations

Virtual Guest Installation

When installing a virtual machine, there are several elements that must be chosen before proceeding with the rest of the installation via **Anaconda**.

Virtual Machine Specifications

1. A domain name must be specified
2. Specify the installation media for the first and second stages of **Anaconda**
3. Specify virtual hardware elements:
 - Number and type of CPU
 - Size of RAM
 - Virtual disk device (file or volume)
 - Network connection and MAC address

Virtual machines can be installed, managed, and accessed with **virt-manager**, a graphical tool. The instructor will demonstrate how to use **virt-manager** in class before you use it in the next practice exercise.

virsh



Note

Para-virtualized hard disks (that use the virtio drivers) appear to guests as **/dev/vd*** instead of **/dev/sd***.

Use this space for notes



References

Red Hat Enterprise Linux Virtualization

- Chapter 6: Virtualized guest installation overview

Red Hat Enterprise Linux Virtualization

- Chapter 7: Installing Red Hat Enterprise Linux 6 as a virtualized guest

virt-manager(1) man page



Practice Performance Checklist

Virtual Guest Installation

In this lab you will install a new virtual machine with Red Hat Enterprise Linux using **virt-manager** and the graphical installer. Once you have successfully completed the lab you will need to remove both the virtual machine and its logical volume to reclaim system resources needed for other labs.

Perform the following steps on desktopX:

- Gracefully shutdown your serverX virtual machine (**vserver**) to reclaim system CPU and RAM resources.
- Create a logical volume 10 GB in size from the **vol0** volume group and name it **guest**.
- Create a Red Hat Enterprise Linux 6 virtual machine with the following characteristics:
 - Name = guest
 - Install media = network install from <http://instructor.example.com/pub/rhel6/dvd>
 - Memory (RAM) = 768 MB
 - CPUs = 1
 - Storage device = the logical volume created in the previous step
- When the installation begins, choose your keyboard and language. Build your guest system according to the following specifications:
 - When asked about the Virtio Block Device, choose **Re-initialize all**.
 - Choose the appropriate time zone
 - Assign **redhat** as the root password
 - Choose the Desktop software set
 - Use the defaults for everything else

Configuring Guests to Start at Boot Time



Practice Group Exercise

Search & Learn: Virtual Machine Automatic Boot

What steps must you take to configure a virtual guest to automatically start at boot time?

1. Launch Virtual Machine Manager.
2. Double-click on the guest virtual machine profile.
3. *View Details*
4. *Boot Option | start vm on host bootup*
5. Check or uncheck the **Start virtual machine on host boot up** check box and click **Apply**.
6. Add the following to the **/etc/sysconfig/libvirt-guests** file:

```
ON_BOOT=ignore
```



Practice Performance Checklist

Configuring Virtual Machines at Boot-time

- Configure the **serverX** (vserver) virtual machine to not start at boot time.
- Configure the **guest** virtual machine to start at boot time.
- Reboot the physical machine (desktopX).
- Confirm the **guest** virtual machine started automatically.
- Configure the **guest** virtual machine to not start at boot time.
- Reboot the physical machine (desktopX).
- Confirm the virtual machine did not start automatically.
- IMPORTANT:** After you successfully complete the lab, delete the **guest** virtual machine and the logical volume it uses for storage. Those resources will need to be available for the criterion test.

Test

Criterion Test

Case Study

Virtual Workstation for William Wonderboy

William Wonderboy just joined the company as a software developer. He needs a machine of his own to write code and do testing without disturbing the work of others. You have been assigned the task of building a virtual machine for him to use.

Create a virtual machine named **wonderboy** with an LVM storage device named **/dev/vol0/wonderboy**. Use the installation media found at the following URL:

- <http://instructor.example.com/pub/rhel6/dvd>

Mr. Wonderboy's virtual machine must have 768 MB RAM and 10 GB of disk storage.

Use a static IP address of 192.168.0.200+X/24, with a gateway and DNS server of 192.168.0.254. Set the hostname to **hostX.example.com**.

Choose an appropriate time zone. Use **redhat** as the root password.

The virtual disk should be partitioned as follows (you will have to re-initialize the disk):

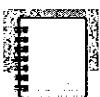
- 250 MB for **/boot**
- 1 GB of swap space
- 6 GB for **/**
- The rest of the space allocated to **/home**

Choose the **Software Development Workstation** software set.

Once the installation is complete, configure NTP to connect to instructor.example.com

Configure this machine to start automatically when the physical host reboots.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Introduction to KVM Virtualization

In this section you learned how to:

- Describe the basic function, components, and benefits of KVM virtualization

Virtual Guest Installation

In this section you learned how to:

- Install a virtual guest according to specification

Configuring Guests to Start at Boot Time

In this section you learned how to:

- Configure the guest to start when the virtualization host boots



UNIT SIXTEEN

AUTOMATED INSTALLATION OF RED HAT ENTERPRISE LINUX

Introduction

Topics covered in this unit:

- **system-config-kickstart**
- Serve Kickstart file
- Installation media
- Perform Kickstart installation
- Kickstart file details

Introductory Overview

Using *Kickstart*, a system administrator can create a single file containing the answers to all the questions typically asked during an installation. This file can then be accessed by the installer to automate installation of Red Hat Enterprise Linux.



Comparison

Kickstart in Red Hat Enterprise Linux is similar to Jumpstart for Oracle Solaris, or to an unattended installation for Microsoft Windows.

Basic Steps:

| | |
|----------------------------|---|
| 1. Create a Kickstart file | 2. Make the Kickstart file available to the installer |
| 3. Boot the installer | 4. Point the installer to the Kickstart file |

Table16.1.Kickstart Steps



References

Red Hat Enterprise Linux Installation Guide
• Chapter 32 - Kickstart Installations

Create a Kickstart File with system-config-kickstart

| | |
|---|---|
| 1. Create a Kickstart file <ul style="list-style-type: none"> • Using system-config-kickstart • Using a text editor | 2. Make the Kickstart file available to the installer |
| 3. Boot the installer | 4. Point the installer to the Kickstart file |

Table 16.2. Kickstart Steps

The tool **system-config-kickstart** presents a number of dialogs. Each dialog that corresponds to a category of questions normally asked by the interactive installer. Default values are not provided by **system-config-kickstart** for all items. In a Kickstart file, required values which are missing may cause the installer to interactively prompt for an answer or to abort the installation entirely.



References

Red Hat Enterprise Linux Installation Guide

- Chapter 33 - Kickstart Configurator

/net/instructor/kickstart



Practice Case Study

Creating a Kickstart File with **system-config-kickstart**

Perform this exercise on your desktopX machine as **student**.

In this exercise, you will pretend that your organization is rolling out Red Hat Enterprise Linux-based workstations for its engineers, and you have been tasked with creating a Kickstart file to facilitate this.

You will install **system-config-kickstart**, and use it to create a Kickstart file according to the parameters specified below:

- Choose the appropriate timezone.
- The **root** password should be **redhat**
- The Kickstart should perform a fresh installation from the web server **http://instructor.example.com/pub/rhel6/dvd**
- The system should have a 100 MB, **ext4** filesystem mounted at **/boot**
- The system should have a 512 MB swap partition
- All remaining disk space should be allocated to an **ext4** partition mounted at **/**
- The **eth0** device should start at boot-time and use DHCP for configuration
- Install the **Base** package set
- Have a post-installation script that adds:

ENGINEERING WORKSTATION

to the file **/etc/issue**

- Save the Kickstart file as **/home/student/engineer.cfg**

How would you address the case study described above? Take notes on your process in the space below and then implement it.

Make the Kickstart File Available to Installers

| | |
|----------------------------|---|
| 1. Create a Kickstart file | 2. <i>Make the Kickstart file available to the installer</i> <ul style="list-style-type: none"> • Network servers: FTP, HTTP, NFS • DHCP/TFTP server • USB disk or CD-ROM • Local hard disk |
| 3. Boot the installer | 4. Point the installer to the Kickstart file |

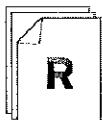
Table 16.3. Kickstart Steps

In order to begin an automated installation, the installer must be able to access the Kickstart file. There are several methods to make the Kickstart file available, and one of the most common is through a network server such as an FTP server, a web server or an NFS server. This method is fairly easy to deploy and makes it easy to manage changes.

Using a DHCP server with TFTP and PXE is more complex to configure and not all locations will allow this method. If you want to configure Kickstart using this method, follow the documentation listed in the references below.

Providing the Kickstart file on USB or CD-ROM can be a convenient way to access the file. Simply place the Kickstart file on the boot media used to start the installation. Be careful when changing the Kickstart, though, as you will need to change it on all of your media.

It can be useful to provide the Kickstart file on the local disk. This allows a quick way to rebuild a development server.



References

Red Hat Enterprise Linux Installation Guide

- Section 32.8 (Making the Kickstart File Available)



Practice Exercise

Make the Kickstart File Available to Installers via HTTP

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Login to desktopX as **student**.
2. Become **root** and deploy a web server.



Note

Recall that you need to Install, Start, Enable and Test

3. Copy **/home/student/engineer.cfg** to **/var/www/html/**
4. Access **http://desktopX/engineer.cfg** from a web browser to test availability.

Create Boot Media

| | |
|---|---|
| 1. Create a Kickstart file | 2. Make the Kickstart file available to the installer |
| <p>3. Boot the installer</p> <ul style="list-style-type: none"> • Installation disks • PXE • boot.iso | 4. Point the installer to the Kickstart file |

Table 16.4. Kickstart Steps

In older versions of Red Hat Enterprise Linux, the **boot.iso** file could be found in the **images/** directory on the installation media. In Red Hat Enterprise Linux 6 it is a separate download from Red Hat Network.

To create a CD, use **cdrecord boot.iso**.

To create a USB stick, use **dd if=boot.iso of=/dev/sdb1**, however, note that the name **/dev/sdb1** will vary depending on the dynamic assignment of device names.

PXE can be used to start the installation from the network. It is fairly complex to configure and may not be allowed in your location, but PXE can be used to start an installation over the network.



References

Red Hat Enterprise Linux Installation Guide

- Section 2.3 - Making Minimal Boot Media

Point the Installer to a Kickstart File

| | |
|----------------------------|---|
| 1. Create a Kickstart file | 2. Make the Kickstart file available to the installer |
| 3. Boot the installer | <p>4. <i>Point the installer to the Kickstart file</i></p> <ul style="list-style-type: none"> • <code>ks=http://server/dir/file</code> • <code>ks=ftp://server/dir/file</code> • <code>ks=nfs:server:/dir/file</code> • <code>ks=hd:device:/dir/file</code> • <code>ks=cdrom:/dir/file</code> |

Table16.5.Kickstart Steps

Once you have chosen your Kickstart method, let the installer know where the Kickstart file is located.

For a virtual machine installation, you can provide the Kickstart URL in a box under **URL Options**. For a physical machine, boot using installation media and press the **Tab** key, then enter one of the **ks=** entries above.

In the classroom environment, we will attempt to replicate a physical installation using virtual machines. Create a new virtual machine and choose PXE as the installation method. PXE will provide a boot screen just like booting from installation media. With **Install or upgrade an existing system** highlighted, press the **Tab** key. This will show the boot line as below:

```
> vmlinuz initrd=initrd.img
```

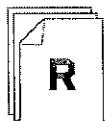
Add a space, then one of the Kickstart entries. The following provide some examples, including the entire boot line. These examples use a Kickstart file named **ks.cfg**, but it could be any name. Once you have entered your Kickstart location, press **Enter** to start the installation.

```
> vmlinuz initrd=initrd.img ks=http://desktopX/ks.cfg
```

```
> vmlinuz initrd=initrd.img ks=ftp://desktopX/pub/ks.cfg
```

```
> vmlinuz initrd=initrd.img ks=nfs:desktopX/var/ftp/pub/ks.cfg
```

```
> vmlinuz initrd=initrd.img ks=hd:sdb1:/kickstart-files/ks.cfg  
> vmlinuz initrd=initrd.img ks=cdrom:/kickstart-files/ks.cfg
```



References

Red Hat Enterprise Linux Installation Guide

- Section 32.8 - Making the Kickstart File Available

Red Hat Enterprise Linux Installation Guide

- Section 28.4 - Automating the Installation with Kickstart

Red Hat Enterprise Linux Installation Guide

- Section 28.2 - Kickstart sources



Practice Exercise

Initiating a Kickstart Installation

Before you begin...

Gracefully shutdown your **serverX (vserver)** virtual machine to reclaim system resources.

Gracefully shutdown your **wonderboy** virtual machine. Delete the virtual machine and delete the disk used for this virtual machine.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Use the Kickstart file created earlier (**engineer.cfg**) to deploy a new virtual machine.

1. On desktopX, create a new virtual machine using **virt-manager**. Choose Network Boot (PXE) as the installation method. Configure the virtual machine using the defaults if not specified below:
 - Name: engineer
 - Create a disk image on the computer's hard drive: 4 GB
2. Devise and enter an appropriate Kickstart invocation line for the **engineer.cfg** file, then start the installation.

Modify a Kickstart File

With each installation, the installer, **anaconda**, will create **/root/anaconda-ks.cfg** containing the settings used to generate this system.



Note

Partitioning information is commented.

Example Kickstart file:

```
# Kickstart file automatically generated by anaconda.

#version=RHEL6
install
url --url=ftp://instructor.example.com/pub/rhel6/dvd
lang en_US.UTF-8
keyboard us
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$UaJVgaTh$KrpFf3K04r9hCZ2hsaa
# Reboot after installation
reboot
firewall --disabled
authconfig --useshadow --enablemd5
selinux --enforcing
timezone --utc America/New_York
bootloader --location=mbr --driveorder=vda --append="crashkernel=auto rhgb quiet"
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
#clearpart --all --drives=vda

#part /boot --fstype=ext4 --size=100
#part pv.ZS1CDM-iUYu-Gfua-YX0W-MSzd-ftBY-7qTB1E --size=28000
#part swap --size=512
#volgroup vol0 --pesize=32768 pv.ZS1CDM-iUYu-Gfua-YX0W-MSzd-ftBY-7qTB1E
#logvol /home --fstype=ext4 --name=home --vgname=vol0 --size=500
#logvol / --fstype=ext4 --name=root --vgname=vol0 --size=8192
repo --name="Red Hat Enterprise Linux" --baseurl=ftp://instructor.example.com/pub/rhel6/
dvd/ --cost=100

%packages
@Base
@Console internet tools
@Core
@Desktop
@Desktop Platform
@Development Tools
@General Purpose Desktop
@Graphical Administration Tools
@Internet Browser
@Network file system client
@Printing client
@X Window System
lftp
```

```
mutt  
ntp  
%end  
  
%post  
# Turn on graphical login  
perl -pi -e 's,id:3:initdefault,id:5:initdefault,' /etc/inittab  
%end
```

Why might you want to manually edit a Kickstart file?

1. The GUI and/or **system-config-kickstart** is unavailable.
2. LVM instructions are needed.
3. Individual packages need to be included or omitted (not just groups).

If you manually edit a Kickstart file, use **ksvalidator** to validate your Kickstart file.

ksvalidator is part of the **system-config-kickstart** package, so you may have to install the package first. **ksvalidator** will ensure the keywords are properly used, but it will not validate URL paths, individual packages or groups nor any part of **%post** or **%pre**.

For instance, if you misspell **firewall --disabled** you may get the following output from **ksvalidator**:

```
[student@desktopX]$ ksvalidator /tmp/anaconda-ks.cfg  
The following problem occurred on line 12 of the Kickstart file:  
  
Unknown command: firewall  
  
[student@desktopX]$ ksvalidator /tmp/anaconda-ks.cfg  
The following problem occurred on line 12 of the Kickstart file:  
  
no such option: --dsabled
```



References

Red Hat Enterprise Linux Installation Guide

- Section 32.4 - Kickstart Options

Red Hat Enterprise Linux Installation Guide

- Section 32.5 - Package Selection

Red Hat Enterprise Linux Installation Guide

- Section 32.6 - Pre-installation Script

Red Hat Enterprise Linux Installation Guide

- Section 32.7 - Post-installation Script



Practice Case Study

Modify a Kickstart File without system-config-kickstart



Note

For time's sake you will not perform an installation using this Kickstart file.

As **root** on desktopX, create a copy of **/root/anaconda-ks.cfg** called **/home/student/projman.cfg**. Using only a text editor, modify that file so it meets the following criteria:

1. The installation must be fully automated, and exactly like the current desktopX installation (including partitioning), except...
 - The **Backup Server** package group will be installed
 - The **mtx** package, which is not installed with the **Backup Server** group by default, will be installed
 - None of the existing scripting from **%pre** and **%post** will be used
 - An **/etc/issue** file will be created in **%post**, which reads:

PROJECT MANAGEMENT

2. **ksvalidator** must be able to validate the file

When complete, copy the file to **/var/www/html/**

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Test

Criterion Test

Performance Checklist

Kickstart a Virtual Machine

Before you begin...

Shutdown the **engineer** virtual machine you created earlier and delete the virtual machine and its disk.

- Boot serverX if it is not running. Copy the **/root/anaconda-ks.cfg** file from serverX to desktopX and call it **/home/student/test.cfg**. Shutdown serverX after you have copied the file to reclaim system resources for the rest of the lab.
- Modify **test.cfg** according to the following criteria:
 - Add **clearpart --all** and **zerombr**, and partition storage according to the following:
 - /boot (ext4) 200 MB
 - swap 512 MB
 - / (ext4) 3 GB
 - Add the **gimp** package
 - Create a **/root/install-date** file with the date and time.
- Copy **test.cfg** to **/var/www/html/** on desktopX. Make sure the file is readable by Apache. Start the **httpd** daemon if it is not already running.
- Create a logical volume in the volume group **vol0** named **test** large enough to serve as the disk for your virtual machine.
- Start a virtual machine installation using your **test.cfg** Kickstart file. Name the virtual machine **test**. Use PXE as the installation method. and allocate 768 MB of RAM and 1 CPU to the virtual machine. Use the logical volume you created in the previous step as the storage for your virtual machine.
- Reboot your virtual machine when it is finished installing and confirm that it installed correctly.



Personal Notes



Unit Summary

Create a Kickstart File with system-config-kickstart

In this section you learned how to:

- Create a Kickstart configuration from scratch with the **system-config-kickstart** utility
- Identify key configuration elements found inside a Kickstart configuration file

Make the Kickstart File Available to Installers

In this section you learned how to:

- Make a Kickstart configuration available via the network (NFS, FTP, HTTP) to the installer
- Make a Kickstart configuration available via local media to the installer

Create Boot Media

In this section you learned how to:

- Create boot media to launch the installer
- Boot over the network to launch the installer

Point the Installer to a Kickstart File

In this section you learned how to:

- Perform an installation using the now available Kickstart configuration file

Modify a Kickstart File

In this section you learned how to:

- Modify an existing Kickstart configuration with the **system-config-kickstart** utility or a text editor

Appendix A. Solutions

Network Configuration and Troubleshooting



Practice Quiz

Viewing and Modifying Network Configuration

- Fill in the below table with the commands, utilities and filenames

| Setting Category | View Current Configuration | Change Configuration |
|----------------------------|--|--|
| IP Address and Subnet Mask | <u>ip addr</u> | <u>NetworkManager or /etc/sysconfig/network-scripts/ifcfg-eth*</u> |
| Routing/Default Gateway | <u>ip route</u> | <u>NetworkManager, /etc/sysconfig/network-scripts/ifcfg-eth* or /etc/sysconfig/network</u> |
| System Hostname | <u>hostname</u> | <u>NetworkManager, hostname or /etc/sysconfig/network</u> |
| Name Resolution | <u>host can be used to resolve names. /etc/hosts and /etc/resolv.conf show the current settings.</u> | <u>NetworkManager, /etc/hosts or /etc/resolv.conf</u> |

Table A.1. Network Configuration from the Command-Line

2. What would a dynamic **ifcfg-eth0** contain?

DEVICE=eth0

ONBOOT=yes

BOOTPROTO=yes

3. What would a static **ifcfg-eth0** contain?

DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.0.X+100
PREFIX=24
GATEWAY=192.168.0.254
DNS1=192.168.0.254



Practice Exercise

Document Network Settings

Before we break our network configuration, let us document the current settings on our serverX system.

1. Log in to serverX as root.
2. a. What is its current IP Address?

192.168.0.X+100

Found by running **ip addr**

- b. What is its current CIDR subnet mask?

/24

Found by running **ip addr**

- c. What is its current default gateway?

192.168.0.254

Found by running **ip route**

- d. What is its current hostname?

serverX.example.com

Found by running **hostname**

- e. What are its current DNS servers?

192.168.0.254

Found by running **cat /etc/resolv.conf**



Test

Criterion Test

Exercise

Troubleshooting Network Configuration from the Command-line

All of the following should be performed on your virtual server, serverX. You will start by running a script that will "break" your network configuration. You will have ten minutes to resolve each of the three problems. Be sure to document what you have found, as we will review at the end.

1. Run the first script to misconfigure your networking:

lab-break-net 1

2. Symptom: A web browser is unable to access the web page at **http://instructor.remote.test**
3. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
4. Document what you have found

Use this space for notes

|

SOLUTION: The hostname *instructor.remote.test* is not being resolved to an IP address, as */etc/resolv.conf* points to the wrong DNS server. Correcting the problem by modifying */etc/sysconfig/network-scripts/ifcfg-eth0* to have **DNS1=192.168.0.254**

5. Run the second script to misconfigure your networking:

lab-break-net 2

6. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
7. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
8. Document what you have found

Use this space for notes

SOLUTION: The host `instructor.remote.test` is on a separate network and unreachable. `ip route` shows the default gateway pointing to the wrong router. Correcting the problem by modifying `/etc/sysconfig/network-scripts/ifcfg-eth0` to have `GATEWAY=192.168.0.254`

9. Run the third script to misconfigure your networking:

`lab-break-net 3`

10. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
11. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
12. Document what you have found

Use this space for notes

SOLUTION: No other host is reachable, as `ip addr` shows the IP address of the `eth0` interface is incorrect. Solve by modifying `/etc/sysconfig/network-scripts/ifcfg-eth0` to have `IPADDR=192.168.0.100+X`.

Administering Users and Groups



Practice Exercise

Create Users Using Command-Line Tools

Create a number of users on your serverX system, setting an initial password (recording in the blanks below).

For the users created, put them into groups as listed here.

| Group | groupname | user_list |
|-------------------|------------------|---------------------------------------|
| Professors | <i>profs</i> | <i>faraday, juliet, elvis</i> |
| Graduate Students | <i>grads</i> | <i>jack, kate, james, elvis</i> |
| Summer Interns | <i>interns</i> | <i>walt, ben, claire, hugo, elvis</i> |

Table A.2. User and Group Assignments

Notice that there is one additional user that you will need to create named **elvis**, who should be placed in all three groups.

1. Log into serverX as *root*.
2. Add the user *juliet*.

```
useradd juliet
```

3. Confirm that *juliet* has been added using the **id** command.

```
id juliet
```

4. Confirm that *juliet* has been added by examining the **/etc/passwd** file.

```
grep 'juliet' /etc/passwd
```

5. Use the **passwd** command to initialize *juliet*'s password and write down the password here:

```
passwd juliet
```

6. Continue adding the remaining users from the list below, remembering to set an initial password and writing them down next to each username:

- faraday
- jack
- kate
- james
- walt
- ben
- claire

- hugo
- elvis

You can run the pair of commands, **useradd** and **passwd**, with the argument of the individual username.

Beyond the scope of this class, but you could also "script" the operation, perhaps something like this:

```
for NAME in faraday jack kate james walt ben claire hugo elvis
do
    useradd ${NAME}
    echo "password" | passwd --stdin ${NAME}
done
```

7. Create the groups specified in the table above, and assign the appropriate group members.

```
groupadd profs
for USERNAMES in faraday elvis; do
    usermod -aG profs ${USERNAMES}
done

groupadd grads
for USERNAMES in jack kate james elvis; do
    usermod -aG grads ${USERNAMES}
done

groupadd interns
for USERNAMES in walt ben claire hugo elvis; do
    usermod -aG interns ${USERNAMES}
done
```



Practice Quiz

Account Maintenance

1. What command would lock **elvis**'s account?

usermod -L elvis

2. What command would then unlock it?

usermod -U elvis

3. What command would cause **elvis**'s account to expire on March 15th, 2012?

chage -E 2012-03-15 elvis

**Practice Quiz****LDAP Client Configuration**

1. What seven pieces of information are typically provided by *User account information* services?
`username:password:UID:GID:GECOS:/home/dir:shell`
2. What "other" type of information can be provided by a *network directory service*?
Authentication method
3. What are the three pieces of information a client machine needs to be configured to get user information from an LDAP directory service?
Server's fully-qualified hostname, Base DN, and CA certificate
4. What does the command `getent passwd ldapuser1` do? Why is this useful?
The command `getent passwd ldapuser1` will print out the account information for the `ldapuser1` user. It is useful to find information about user accounts, such as home directories, shell, names, etc.

**Practice Case Study****Automounting NFS Directories**

These steps should be performed on serverX.

Your company is now taking on several new clients:

1. The Organization of Secret Hidden Undertakings (or OSHU)
2. Race Along the Lake Investments, Inc. (or RALII)

You company has a new NFS server with shares for storing files related to these "special" clients named `instructor.example.com`, with currently two shares: `/var/nfs/oshu` and `/var/nfs/ralii`, with the expectation that more will be added as new clients are signed.

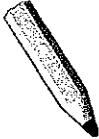
The workstations need to use `autofs` to automatically mount these shares to: `/special/oshu` and `/special/ralii`, respectively with read-only permissions.

Given the expectation that additional clients will be signed shortly, implement this using `autofs` wildcards and metacharacters.

1. Modify the configuration of the automounter on serverX so that if any directory is accessed in `/special`, the automounter will mount the NFS export `instructor.example.com:/var/nfs dirname` on it if that export exists. In this example, "dirname" stands in for any possible directory name:

```
# cat /etc/auto.master
/special /etc/auto.special
# cat /etc/auto.special
* -ro instructor.example.com:/var/nfs/&
```

2. Reload the service with **service autofs reload**.
3. Test, by in turn accessing the contents of **/special/oshu** and **/special/ralii**.



Test

Criterion Test

Exercise

Get Network User Information from an LDAP Directory Service

You will now configure serverX to get information about network users from an LDAP directory server available to all machines in the classroom.

Here is information that was provided to you about the LDAP server:

- Hostname: *instructor.example.com*
- Search Base DN: *dc=example,dc=com*
- CA Certificate: *http://instructor.example.com/pub/EXAMPLE-CA-CERT*¹

You will then configure serverX to automatically mount the home directories of your LDAP-based network users when they log in.

Here is information which was provided to you about the NFS storage server that contains the home directories:

- Hostname: *instructor.example.com*
 - Exported Directory: **/home/guests/**
1. Login to serverX as *root*. If you use **ssh**, include the **-X** option to forward graphical interfaces to your workstation.
 2. Use **system-config-authentication** to configure serverX to get network user information from the classroom LDAP server.
 1. For "User Account Database", choose *LDAP*.
 - a. For "LDAP Search Base DN:", enter *dc=example,dc=com*.
 - b. For LDAP Server:, enter *instructor.example.com*
 - c. Select the "Use TLS to encrypt connects" checkbox.
 - d. Choose "Download CA Certificate...", and enter *http://instructor.example.com/pub/example-ca.crt*.
 2. For "Authentication Method", choose *LDAP*.
 3. Choose *Apply*.

3. Use the **id** command to confirm that the user **ldapuserX** (replace X with your station number) is defined on the system.

```
[root@server1 ~]# id ldapuser1
uid=1701(ldapuser1) gid=1701(ldapuser1) groups=1701(ldapuser1)
```

4. Use the **getent** command to look up the user information for **ldapuser1**.

```
[root@server1 ~]# getent passwd ldapuser1
ldapuser1:*:1701:1701:LDAP Test User 1:/home/guests/ldapuser1:/bin/bash
```

5. Login to serverX as **ldapuserX** or any of the other network users, using the password "password". (Do not worry if you get an error on login about a non-existent home directory, that is expected.)

```
[student@desktop1 ~]# ssh ldapuser1@server1
ldapuser1@server1's password: password
Could not chdir to home directory /home/guests/ldapuser1: No such file or directory
-bash-4.1$
```

6. Modify the configuration of the automounter on serverX so that if any directory **ldapuserX** is accessed in /home/guests, the automounter will attempt to mount the equivalent NFS exported directory from **instructor.example.com:/home/guests/ldapuserX** (Hint: Use wildcard syntax.)

1. Add the following line to **/etc/auto.masters**.

```
/home/guests          /etc/auto.guests
```

2. Create the file **/etc/auto.guests**, with the following single line.

```
*      -rw,hard,intr    instructor.example.com:/home/guests/&
```

3. Restart the automounter. (Note: in early Red Hat Enterprise Linux 6 releases, the **autofs** service script has a bug, and the standard *restart* does not work.)

```
[root@server1 ~]# service autofs stop
Stopping automount:                                     [  OK  ]
[root@server1 ~]# service autofs start
Starting automount:                                     [  OK  ]
```

7. Log into serverX as **ldapuserX**, where "X" is your station number, with the password "password". The user's home directory should be automatically mounted.

```
[student@desktop1 ~]$ ssh ldapuser1@server1
The authenticity of host 'server1 (192.168.0.101)' can't be established.
RSA key fingerprint is 33:fa:a1:3c:98:30:ff:f6:d4:99:00:4e:7f:84:3e:c3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1,192.168.0.101' (RSA) to the list of known hosts.
ldapuser1@server1's password: password
Last login: Thu Dec 16 14:59:49 2010 from instructor.example.com
```

```
[ldapuser1@server1 ~]$ pwd  
/home/guests/ldapuser1  
[ldapuser1@server1 ~]$ df  
Filesystem      1K-blocks      Used Available Use% Mounted on  
...  
instructor.example.com:/home/guests/ldapuser1  
      1032192      36864     943104    4% /home/guests/ldapuser1
```

Command-line Process Management



Practice Performance Checklist

Launching Graphical Tools from Bash

- Log in to your serverX host graphically as **student**.
- Open a terminal window.
- Within the window switch to a **root** shell.

```
[student@serverX ~]$ su -  
Password: redhat  
[root@serverX ~]#
```

- Launch **nautilus** in the foreground from the command line.

```
[root@serverX ~]# nautilus
```

- Use the keyboard shortcut to get your shell prompt back without terminating the process.

```
[root@serverX ~]# nautilus  
Ctrl+z  
[1]+ Stopped nautilus
```

- Put **nautilus** in the background.

```
[root@serverX ~]# bg  
[1]+ nautilus &
```

- List your current shell jobs.

```
[root@serverX ~]# jobs  
[1]+ Running nautilus &
```

- Exit the root shell.

```
[root@serverX ~]# Ctrl+d  
[student@serverX ~]$
```



Practice Exercise

Managing Processes

Before you begin...

On serverX run the **lab-setup-processes** command.

```
[root@serverX ~]# lab-setup-processes
```

1. Change the priority of the process that is using the most CPU resources to 5.

```
[root@serverX ~]# top
...
PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM     TIME+ COMMAND
26669 root      22   2 3828   84   12 R 78.3  0.0   0:18.41 hippo
26670 root      22   2 4008  280  156 S 21.2  0.1   0:04.94 process101
...
```

In **top**, press the **r** key to renice a process, then enter the PID (**26669** in the example above). Press **5** to change the process to a nice value of **5**.

2. Terminate the process that is using the most memory resources.

In **top**, press the **M** key to sort by memory consumption.

```
PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM     TIME+ COMMAND
26668 root      22   2 55032  50m  160 S  0.0 10.2   0:00.18 elephant
```

Press the **k** key to kill a process. Enter the PID (**26668** in the example above). Enter the signal, or simply accept the default of **15**.

3. Execute **lab-grade-processes** on serverX to confirm you identified and managed the correct processes.

```
[root@serverX ~]# lab-grade-processes
* Checking for terminated process... PASS
* Checking for reniced process... PASS
```

4. Run the **lab-cleanup-processes** command to clean up.

```
[root@serverX ~]# lab-cleanup-processes
```



Practice Quiz

cronie Scheduling

1. When will the following jobs run?

a. 00 07 25 12 * /usr/local/bin/open_presents

7am Christmas Day

b. */5 * * * * /usr/local/bin/take_stats

every 5 minutes

c. 07 03 * * * /sbin/service xend restart

every day at 3:07am

d. 30 16 * * 5 /usr/local/bin/mail_checks

every Friday at 4:30pm

2. Devise a cron entry which would run the script **/usr/local/bin/vacuum_db** once a month on the first day of the month.

05 02 1 * * /usr/local/bin/vacuum_db

(of course, the actual hour and minute are arbitrary.)

3. What if the machine in the previous question was down for maintenance on February first? What would be a better way to insure the database doesn't operate 2 (or more) months between vacuuming?

Write the job as a simple shell script, and drop it into **/etc/cron.monthly**:

```
#!/bin/bash  
/usr/local/bin/vacuum_db
```



Test

Criterion Test

The following are to be performed on your serverX machine.

Exercise

Managing Processes

1. On your serverX, execute the script **manage_processes_start**.
2. The server station should now be very sluggish.
3. Determine the process which is using excessive amounts of memory, and terminate the process.
 1. In a terminal, start the program **top**.
 2. Within **top**, enter "O" to bring up the sort menu.
 3. Enter **n** to sort by memory use.

4. Note the PID of the top memory consumer.
 5. Enter **k** to kill a process.
 6. Enter the PID of the process to terminate.
 7. Hit **<RETURN>** to choose the default signal number 15.
4. Determine the processes which are using excessive amounts of CPU, and renice them to a niceness of 15.
 1. In a terminal, start the program **top**.
 2. Note the PIDs of the top CPU consumer.
 3. Enter **r** to renice a process.
 4. Enter the PID of the process to renice.
 5. Enter the new niceness value of 15.
 6. Repeat the last 3 steps for each process which is using an excessive amount of memory.
 5. Create a system cron job which once every half hour readjusts all processes owned by the user **elvis** to a niceness of 10

Using a text editor, create the file **/etc/cron.d/nice_elvis** with the following content:

```
# Adjust the priority of processes owned by user elvis every half hour
# to a nice value of 15.
*/30 * * * * /usr/bin/renice -n 15 -u elvis
```

Get Help from Red Hat



Practice Resequencing Exercise

Working with Red Hat Global Support Services

Below are the steps taken when interacting with Red Hat Global Support Services. Mark the order the steps should be taken:

- 4 Gather relevant diagnostic info (log information, core dumps, etc.)
- 7 Have support ticket be transferred to a technician in your region
- 1 Define the problem
- 6 Contact Red Hat via phone or web
- 5 Determine the severity level
- 3 Gather background information
- 2 Search documentation and kbase articles

Manage System Resources

Default Log Files

Fill in the name of the log file as you review the contents of **/etc/rsyslog.conf** on serverX.

1. All authentication-related messages go to /var/log/secure
2. Anything e-mail related goes to /var/log/maillog
3. Messages related to cron go to /var/log/cron
4. All other messages sent at **info** priority or higher are saved in /var/log/messages



Practice Quiz

Review rsyslog

Answers the following questions:

1. Which two fields are used to match log events?
Facility and Priority
2. What is the effect of a wildcard in the first field?
It matches every facility
3. What is the effect of a wildcard in the second field?
It matches every priority
4. Is it possible for the same log event to be recorded in more than one log?
Yes, it will go to any log with matching criteria



Practice Performance Checklist

Analyze a Log Summary Report

Determine the amount of free space for the root filesystem from the latest logwatch report on serverX.

- Open the email of root

```
[root@serverX ~]# mail
```

- Locate and read the most recent logwatch report. If there is no logwatch report, run the **logwatch** command to generate one manually.

```
[root@serverX ~]# mail
Heirloom Mail version 12.4 7/29/08. Type ? for help.
"/var/spool/mail/root": 1 message 1 new
>N 1 logwatch@serverX.exa Tue Jan 1 00:25 96/2948 "Logwatch for serverX."
```

& Enter

(use Space to scroll to the end)

----- Disk Space Begin -----

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------------------|------|------|-------|------|------------|
| /dev/mapper/vgsrv-root | 3.3G | 2.3G | 876M | 73% | / |
| /dev/vda1 | 248M | 30M | 206M | 13% | /boot |
| /dev/mapper/vgsrv-home | 248M | 11M | 225M | 5% | /home |

----- Disk Space End -----

Logwatch End

- Record the amount of free space for the / filesystem:

In the example above, 876M.



Practice Case Study

Redirect Log Summary e-mails

Change the configuration of **logwatch** on serverX to send log summary reports to user **student** rather than user **root**.

Bonus Question: How would you send the **logwatch** reports to both **student** and **root**?

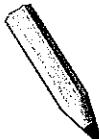
1. Modify the **/etc/logwatch/conf/logwatch.conf** file to include **MailTo=student**.
2. To send it to both **student** and **root**, change the line to:

MailTo = student root

3. Run the **logwatch** command:

[root@serverX]# logwatch

4. Check email of **student** and **root** for new reports.



Test

Criterion Test

Case Study

Log Debugging Messages

Your company wants to include a debug log on your server for analysis. Redirect all debugging level messages (and higher priority) to a file named **/var/log/debug.log**.

1. Configure **rsyslog** to create a **debug.log** in **/var/log/** directory collecting debugging messages from all services.

Add this line to **/etc/rsyslog.conf**:

```
*.debug          /var/log/debug.log
```

2. Activate the changes made to **rsyslog**.

```
[root@serverX ~]# service rsyslog restart
Shutting down system logger:                                [  OK  ]
Starting system logger:                                    [  OK  ]
```

3. Use the **logger** command to send a message to **rsyslogd** with **debug** priority and verify the message was logged to the new log file.

```
[root@serverX ~]# logger -p debug Testing debug
[root@serverX ~]# tail /var/log/debug.log
Jan  5 11:03:04 server1 root: Testing debug
```

Note that the message will *not* show up in **/var/log/messages** if you configured and tested it correctly because **/var/log/messages** only gets messages of **info** priority and higher (not **debug**).

Installing and Managing Software



Practice Quiz

Red Hat Network Registration

1. The command-line tool that begins the registration with the Red Hat Network is rhn_register.
2. The first registration choice determines whether a system registers with Hosted RHN or RHN Satellite.
3. Optionally additional web proxy server and authentication information may need to be provided.
4. An RHN user name or RHN account and its matching password must be provided for successful Red Hat Network registration.
5. The last questions to be answered during the registration process are system name and whether to upload hardware and software or package profile information.



Practice Exercise

Searching For and Installing Packages

Login as **root** on serverX and perform the following tasks:

1. Attempt to run the command **gnuplot**. You should find that it is not installed.
2. Search for plotting packages.

yum search plot

3. Find out more information about the **gnuplot** package.

yum info gnuplot

4. Install the **gnuplot** package.

yum install gnuplot

5. Attempt to remove the **gnuplot** package, but say no.

yum remove gnuplot

How many packages would be removed? 1

6. Attempt to remove the **gnuplot-common** package, but say no.

yum remove gnuplot-common

How many packages would be removed? 2



Practice Exercise

Handling Third-Party Software

In this exercise you will gather information about a third-party package, extract files from it, and install it as a whole on your desktopX system.

1. Download `wonderwidgets-1.0-4.x86_64.rpm` from <http://instructor/pub/materials>.

2. What files does it contain?

```
rpm -q -p wonderwidgets-1.0-4.x86_64.rpm -l
```

3. What scripts does it contain?

```
rpm -q -p wonderwidgets-1.0-4.x86_64.rpm --scripts
```

4. How much disk space will it use when installed?

```
rpm -q -p wonderwidgets-1.0-4.x86_64.rpm -i
```

5. Use `yum localinstall` to install the package.

```
yum localinstall wonderwidgets-1.0-4.x86_64.rpm
```



Practice Exercise

Using yum Repositories

You will configure your server to use a separate `yum` repository to obtain updates, and update your machine.

1. Create the file `/etc/yum.repos.d/errata.repo`, to enable the “Updates” repository found on the instructor machine. It should access content found at the following URL: `ftp://instructor.example.com/pub/rhel6/Errata`

Create the file `/etc/yum.repos.d/errata.repo` with the following content:

```
[updates]
name=Red Hat Updates
baseurl=ftp://instructor.example.com/pub/rhel6/Errata
enabled=1
gpgcheck=1
```

2. Update all relevant software provided by the repository, using `yum update`.

```
yum update
```



Test

Criterion Test

Case Study

Update and Install Software

Before you begin...

Reset your serverX lab system with the **lab-setup-server** command.

You have a new server, serverX, to administrate that has very specific software requirements. It must have the latest version of the following packages installed:

kernel (existing package w/ an update)

xsane-gimp (new package)

bzip2 (updated package)

For security reasons it should not have the **xinetd** package installed.

Do not install all updates. Only install updates for the packages listed above if they are available.

Updated packages can be found at the following URL: <ftp://instructor.example.com/pub/rhel6/Errata>

When you finish, run the **lab-grade-packages-2** evaluation script to make sure that you have done everything correctly.

1. Create the file **/etc/yum.repos.d/updates.repo** with the following content:

```
[updates]
name=Red Hat Updates
baseurl=ftp://instructor.example.com/pub/rhel6/Errata
enabled=1
gpgcheck=1
```

2. Install the specified packages and updates, making sure to specify the individual packages, so that a full update is not performed. Note that, when packages are explicitly named on the command line, *install* and *update* are essentially synonyms.

(Do not be overly concerned if your package count is not identical to that listed below.)

```
[root@serverX ~]# yum install kernel xsane-gimp bzip2
```

```
...
```

```
Transaction Summary
```

```
=====
Install      15 Package(s)
Upgrade      2 Package(s)
```

```
Total download size: 39 M
```

```
Is this ok [y/N]: y
```

```
...  
Importing GPG key 0xFD431D51 "Red Hat, Inc. (release key 2) <security@redhat.com>"  
from /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release  
Is this ok [y/N]: y  
Importing GPG key 0x2FA658E0 "Red Hat, Inc. (auxiliary key) <security@redhat.com>"  
from /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release  
Is this ok [y/N]: y  
...  
Complete!
```

3. Make sure the **xinetd** package is removed.

```
[root@serverX ~]# yum erase xinetd  
...  
No Match for argument: xinetd  
Package(s) xinetd available, but not installed.  
No Packages marked for removal
```

In this case, *xinetd* was not installed.

Administer Remote Systems

- Fill in the blanks as your instructor demonstrates the use of **ssh** and covers these key points.
1. SSH is more secure than telnet because all communication between hosts is encrypted.
 2. **ssh -X user@host.fqdn** initiates a remote connection to host.fqdn as **user**.
 3. The first time an SSH connection is made to a system, the public key of the remote system is stored locally so its identity can be verified each time a future connection is started.
 4. The exit command is used to finish an SSH session and return to the local shell.



Practice Quiz

Remote Shell Access Quiz

Connect to serverX from desktopX using a remote shell. Answer the following questions running commands from that remote shell:

1. The Disk Utility command is **palimpsest**. /dev/vda is the name of the hard drive on serverX.
2. Red Hat Enterprise Linux 6 (Santiago) is the name of the OS release according to **/etc/redhat-release**.
3. Run **nautilus** or use the command-line in the remote shell on serverX to perform the following:
 - Create a file named **a1.txt** in **/root**
 - Create a directory named **b2** in **/home/student** which is owned by the **student** user and the **student** group.



Practice Performance Checklist

Remote File Transfers

- Use **rsync** to backup **student**'s home directory on desktopX to the **/tmp** directory on serverX.

```
[student@desktopX ~]$ rsync /home/student serverX:/tmp  
...
```

- Create a new file named **z.txt** in **student**'s home directory.

```
[student@desktopX ~]$ echo test > ~/z.txt
```

- Use the same **rsync** command to backup **student**'s home directory on desktopX to the **/tmp** directory on serverX.

```
[student@desktopX ~]$ rsync /home/student serverX:/tmp  
...
```

- Remove the **Desktop** directory from the backup on serverX. Run the same **rsync** command.

```
[student@desktopX ~]$ ssh serverX 'rm -rf /tmp/student/Desktop'  
...  
[student@desktopX ~]$ rsync /home/student serverX:/tmp  
...
```



Practice Performance Checklist

Securely Transferring Backups

- Create an SSH key-pair as **student** on desktopX.

```
[student@desktopX ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/student/.ssh/id_rsa):  
Created directory '/home/student/.ssh'.  
Enter passphrase (empty for no passphrase): <RETURN>  
Enter same passphrase again: <RETURN>  
Your identification has been saved in /home/student/.ssh/id_rsa.  
Your public key has been saved in /home/student/.ssh/id_rsa.pub.  
...
```

- Send the SSH public key to the **student** account on serverX.

```
[student@desktopX ~]$ ssh-copy-id serverX  
The authenticity of host 'serverX (192.168.0.101)' can't be established.  
RSA key fingerprint is 33:fa:a1:3c:98:30:ff:f6:d4:99:00:4e:7f:84:3e:c3.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'serverX,192.168.0.101' (RSA) to the list of known  
hosts.  
student@serverX's password: student  
Now try logging into the machine, with "ssh 'serverX'", and check in:  
  
.ssh/authorized_keys
```

to make sure we haven't added extra keys that you weren't expecting.

- Run the **rsync** command used before to backup **student**'s home directory on desktopX to the **/tmp** directory on serverX.

```
[student@desktopX ~]$ rsync -av /home/student serverX:/tmp  
sending incremental file list  
student/  
...  
student/.ssh/id_rsa.pub
```

```
student/.ssh/known_hosts

sent 3384 bytes received 150 bytes 7068.00 bytes/sec
total size is 2806 speedup is 0.79
```

Securing SSH Search & Learn

1. Use **yum** or **rpm** to determine which package provides the SSH service.

```
[root@serverX ~]# rpm -q -f /usr/sbin/sshd
openssh-server...
[root@serverX ~]# yum search 'ssh server'
openssh-server...
```

2. Use **rpm -q -l** or **rpm -q -c** to determine the primary configuration file for the service.

```
[root@serverX ~]# rpm -q openssh-server -c
/etc/pam.d/sshd
/etc/ssh/sshd_config
```

3. Reviewing the man page for the configuration file, which directive disables root login?

```
[root@serverX ~]# man sshd_config
/root login
```

Answer: **PermitRootLogin**

4. Which directive in that configuration file disables password login?

```
[root@serverX ~]# man sshd_config
/password
```

Answer: **PasswordAuthentication**



Practice Performance Checklist Securing SSH

- If not done earlier, generate SSH keys on desktopX and copy the public key to the **student** account on serverX and verify that the keys are working:

```
[student@desktopX ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa):
```

```
Created directory '/home/student/.ssh'.
Enter passphrase (empty for no passphrase): <RETURN>
Enter same passphrase again: <RETURN>
Your identification has been saved in /home/student/.ssh/id_rsa.
Your public key has been saved in /home/student/.ssh/id_rsa.pub.
...
[student@desktopX ~]$ ssh-copy-id serverX
The authenticity of host 'serverX (192.168.0.101)' can't be established.
RSA key fingerprint is 33:fa:a1:3c:98:30:ff:f6:d4:99:00:4e:7f:84:3e:c3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'serverX,192.168.0.101' (RSA) to the list of known
hosts.
student@serverX's password: student
Now try logging into the machine, with "ssh 'serverX'", and check in:

    .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

- Configure SSH on serverX to prevent root logins.

Modify the **/etc/ssh/sshd_config** file so that the line looks as follows (be sure to remove the comment):

```
PermitRootLogin no
```

- Restart the SSH service.

```
[root@serverX ~]# service sshd restart
```

- Confirm that **root** cannot log in with SSH, but **student** is permitted to log in.
- Configure SSH on serverX to prevent password authentication.

Modify the **/etc/ssh/sshd_config** file so that the line looks as follows:

```
PasswordAuthentication no
```

Key tar Options

1. c = create
2. x = extract
3. t = test
4. v = verbose
5. f = file name

6. z = gzip

7. j = bzip2



Practice Exercise

Archive and Compress Files

1. Login to serverX as **root**.

```
[root@desktopX ~]# ssh root@serverX
```

2. Create an archive of **/etc** using **gzip** compression. Save the file as **/tmp/etc.tar.gz**.

```
[root@serverX ~]# tar cvzf /tmp/etc.tar.gz /etc  
...  
[root@serverX ~]# exit
```

3. Copy the **/tmp/etc.tar.gz** file to **/backups** on your desktopX machine.

```
[root@desktopX ~]# mkdir /backups  
[root@desktopX ~]# rsync serverX:/tmp/etc.tar.gz /backups
```

4. Extract the compressed archive to **/backups** on desktopX.

```
[root@desktopX ~]# cd /backups  
[root@desktopX backups]# tar xjvf etc.tar.gz  
...
```



Test

Criterion Test

Exercise

SSH Keys and File Archives

1. Reset serverX by running **lab-setup-remote**.
2. Install the SSH public key generated previously on desktopX to the **student** account on serverX.

```
[student@desktopX ~]$ ssh-copy-id serverX  
student@serverX's password: student  
Now try logging into the machine, with "ssh 'serverX'", and check in:
```

```
.ssh/authorized_keys
```

to make sure we haven't added extra keys that you weren't expecting.

3. Archive **student**'s home directory on desktopX into **/tmp/student.tar.bz2**.

```
[student@desktopX home]$ cd /home  
[student@desktopX home]$ tar cf /tmp/student.tar.bz2 student/
```

4. Copy the **/tmp/student.tar.bz2** file on desktopX to **/tmp** on serverX.

```
[student@desktopX ~]$ scp /tmp/student.tar.bz2 serverX:/tmp  
student.tar.bz2                                100% 2659      2.6KB/s   00:00
```

5. Remove **student**'s home directory on serverX.

```
[student@desktopX ~]$ ssh root@serverX  
root@serverX's password: redhat  
[root@serverX ~]# rm -fr /home/student/
```

6. Restore **student**'s home directory from the **/tmp/student.tar.bz2** archive.

```
[root@serverX ~]# cd /home/  
[root@serverX home]# tar xf /tmp/student.tar.bz2  
[root@serverX home]# ls -al student/  
total 18  
drwx----- 5 student student 1024 Dec 15 13:37 .  
drwx----- 8 root     root    1024 Dec 15 13:59 ..  
-rw-----  1 student student  80 Dec 15 13:37 .bash_history  
-rw-r--r--  1 student student  18 Jun 22 11:49 .bash_logout  
-rw-r--r--  1 student student 176 Jun 22 11:49 .bash_profile  
-rw-r--r--  1 student student 124 Jun 22 11:49 .bashrc  
drwxr-xr-x  2 student student 1024 Jul 14 11:55 .gnome2  
drwxr-xr-x  4 student student 1024 Dec 13 03:24 .mozilla  
drwx----- 2 student student 1024 Dec 15 12:48 .ssh
```

7. Install the SSH public key from the backup to desktopX and verify you can use the SSH keys to get from serverX to desktopX without typing a password.

```
[root@serverX home]# su - student  
[student@serverX ~]$ ssh-copy-id desktopX  
The authenticity of host 'desktopX (192.168.0.1)' can't be established.  
RSA key fingerprint is 3b:2d:7a:6f:f6:1f:26:37:e9:86:a4:aa:51:4a:9d:0d.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'desktopX,192.168.0.1' (RSA) to the list of known hosts.  
student@desktopX's password: student  
Now try logging into the machine, with "ssh 'desktopX'", and check in:
```

```
.ssh/authorized_keys
```

to make sure we haven't added extra keys that you weren't expecting.

```
[student@serverX ~]$ ssh desktopX hostname  
desktopX.example.com
```

8. Verify you have accomplished the tasks by running **lab-grade-remote**.

Deploy and Secure Services



Practice Performance Checklist

Manage the System Clock

- Configure desktopX to synchronize with `instructor.example.com` using NTP.
 1. Launch the Date & Time management tool.
 2. Click the Date and Time tab.
 3. Enable NTP, point to `instructor.example.com`.
 4. Synchronize clock immediately.
- Set the timezone to the appropriate setting for your locale.
 1. Click the Time Zone tab.
 2. Set the timezone for your locale.
- Make the hardware clock store UTC time.
 1. On the Time Zone tab, select UTC.

Installing New Services

What command is commonly used to perform each of these steps of deploying a new service on a Red Hat Enterprise Linux system?

- **Install** the software: `yum`
- **Start** the service: `service`
- **Enable** the service at bootup: `chkconfig`
- **Test** the services: will be covered on the next pages



Practice Exercise

Enabling a VNC Server

1. Install the `tigervnc-server` package on serverX.

```
[root@serverX ~]# yum install tigervnc-server
```

2. Configure VNC display 1 for student.

Add the following to `/etc/sysconfig/vncservers`:

```
VNCSERVERS="1:student"
```

3. Set **redhat** as the VNC password for student.

```
[student@serverX ~] vncpasswd  
Password: redhat  
Verify: redhat
```

4. Start and enable the VNC service.

```
[root@serverX ~]# service tigervnc start  
[root@serverX ~]# chkconfig tigervnc on
```

5. You will verify the connection in the next section.



Practice Exercise

Connect to VNC Securely

1. Configure the VNC server on serverX to allow local connections only (unless you already did this in the previous exercise).

Edit **/etc/sysconfig/vncservers** and add, if needed, the following:

```
VNCERVERARGS[1]="-localhost"
```

2. Connect to the VNC server on serverX securely from desktopX using an SSH tunnel

```
[student@desktopX ~] vncviewer -via serverX localhost:1
```

3. Verify everything is completed as specified.



Practice Performance Checklist

Deploy an FTP server

Deploy an FTP server on serverX. Verify it is working and enabled.

- Install the **vsftpd** package.

Note that the package is already installed, but the following confirms that.

```
[root@serverX ~]# yum install vsftpd
```

- Start the **vsftpd** service.

```
[root@serverX ~]# service vsftpd start
```

- Enable the **vsftpd** service.

```
[root@serverX ~]# chkconfig vsftpd on
```

- Publish a copy of **/etc/hosts** to the anonymous FTP document root.

```
[root@serverX ~]# cp /etc/hosts /var/ftp/
```

- Test the FTP server on desktopX with an ftp client (**lftp** or **Nautilus**) to connect to the server `ftp://serverX.example.com`. Download the **hosts** file to student's home directory.



Practice Performance Checklist

Restrict FTP Access

Because FTP is an insecure protocol, it is a security risk to allow normal users to connect and authenticate. Configure your FTP server to permit anonymous connections only.

- Use **lftp** to connect to serverX and authenticate as **student** to confirm it allows non-anonymous users.
- Which file is the main vsftpd configuration file?

/etc/vsftpd/vsftpd.conf

- Which configuration file directive controls non-anonymous access to the system?

local_enable

- Configure vsftpd to deny access by local, non-anonymous users.

Edit the **/etc/vsftpd/vsftpd.conf** configuration file and modify the following line as:

```
local_enable=no
```

Restart the service.

```
[root@serverX ~]# service vsftpd restart
```

- Retest your server and confirm student no longer has authenticated access to your FTP server.



Practice Performance Checklist

Deploy a Web Server

The instructor will split up the class into groups. Once you are in your group, do the following:

Given that the name of the web server package is **httpd**, deploy a web server on serverX. It should provide HTTP file services. It should be active when your server is rebooted.

- Install the **httpd** package.

```
[root@serverX ~]# yum install httpd
```

- Start the **httpd** service.

```
[root@serverX ~]# service httpd start
```

- Enable the **httpd** service.

```
[root@serverX ~]# chkconfig httpd on
```

- Create a symbolic link in your web server document root to the **/pub** directory in your FTP server and call it **pub**.

```
[root@serverX ~]# cd /var/www/html  
[root@serverX html]# ln -s ../../ftp/pub pub
```

- Create an **index.html** file in the document root of your web server with the following contents:

```
<h1>Classroom Web Services</h1>  
<p>  
<a href="pub">Click here</a> to view public files.
```

- Reboot and verify this content is available through your web browser before you notify the public to ensure your customers can access it as well.

- Test the web server using the **Firefox** browser.

Pointing the **Firefox** browser to *http://serverX.example.com*



Practice Performance Checklist

Allow HTTP and FTP through the Firewall

For this activity, the instructor will break you up into small groups. One student in each group will determine the steps needed to perform the activity while the others watch, make suggestions, and take notes. Once the activity has been successfully completed by the group, the students taking notes should return to their own computers and repeat the activity there.

Virtual Training students: Complete the activity on your own, and ask the instructor if you have any questions.

- If not done already, deploy a default configuration FTP server on serverX.

```
[root@serverX ~]# yum install vsftpd  
...  
[root@serverX ~]# service vsftpd start  
...  
[root@serverX ~]# chkconfig vsftpd on
```

- If not done already, deploy a default configuration HTTP server on serverX.

```
[root@serverX ~]# yum install httpd  
...  
[root@serverX ~]# service httpd start  
...  
[root@serverX ~]# chkconfig httpd on
```

- Enable the firewall on serverX

```
[root@serverX ~]# system-config-firewall-tui
```

Toggle on **Enabled**

Select **Customize**

- Allow connections to the FTP service through the firewall on serverX

Toggle on **FTP** under Customize

- Allow connections to the HTTP service through the firewall on serverX

Toggle on **WWW (HTTP)** and **Secure WWW (HTTPS)** under Customize

- Allow connections to the SSH service through the firewall on serverX

Toggle on **SSH** under Customize

Select **Forward**, then **Close**

Select **Ok**, then **Yes** to save

- Test that each service is accessible from desktopX



Test

Criterion Test

Performance Checklist

Deploy File Sharing Services

Before you begin...

Reset your serverX lab system with the **lab-setup-server** command.

Nickel and Copper Cutlery want to publish an on-line catalog to their customers. Deploy FTP and HTTP services and confirm they are working and enabled at boot.

- Create a file called **index.html** with exactly two lines that contain the following content:

```
NICKEL AND COPPER CUTLERY
On-line catalog coming soon!
```

```
[root@serverX ~]# yum install -y httpd vsftpd
...
Package httpd-2.2.15-5.el6.x86_64 already installed and latest version
Package vsftpd-2.2.2-6.el6.x86_64 already installed and latest version
Nothing to do
[root@serverX ~]# echo 'NICKEL AND COPPER CUTLERY' > /var/ftp/pub/index.html
[root@serverX ~]# echo 'On-line catalog coming soon!' >> /var/ftp/pub/index.html
```

- Configure serverX to provide both FTP and web services. Disable non-anonymous FTP access.

```
[root@serverX ~]# sed -i.orig 's/^local_enable=/#local_enable=' /etc/vsftpd/
vsftpd.conf
[root@serverX ~]# diff /etc/vsftpd/vsftpd.conf.orig /etc/vsftpd/vsftpd.conf
15c15
< local_enable=YES
---
> #local_enable=YES
[root@serverX ~]# service httpd start
Starting httpd: [ OK ]
[root@serverX ~]# chkconfig httpd on
[root@serverX ~]# service vsftpd start
Starting vsftpd for vsftpd: [ OK ]
[root@serverX ~]# chkconfig vsftpd on
```

- Configure your serverX machine to serve identical file content to both anonymous FTP and HTTP users. The following URLs should both display the file you created above:

- <ftp://serverX/pub/index.html>
- <http://serverX/index.html>

```
[root@serverX ~]# rmdir /var/www/html/
[root@serverX ~]# ln -s ..//ftp/pub /var/www/html
[root@serverX ~]# ls /var/www/html
index.html
```

- Grading: Reboot your serverX machine. Use a web browser to confirm your services are functioning correctly.

```
[root@serverX ~]# curl http://serverX/index.html
```

```
NICKEL AND COPPER CUTLERY
On-line catalog coming soon!
[root@serverX ~]# curl ftp://serverX/pub/index.html
NICKEL AND COPPER CUTLERY
On-line catalog coming soon!
```

SELinux Management



Practice Quiz

Basic SELinux Concepts

1. To which of the following does SELinux apply security context (check all that apply)?

(select one or more of the following...)

- a. Ports
- b. Processes
- c. Files
- d. Directories
- e. Remote file systems

2. SELinux can be used to:

(select one or more of the following...)

- a. Protect a service from running on other ports.
- b. Protect user data from applications like the web server
- c. Block remote systems from accessing local ports

This describes a firewall.
d. Keep the system updated

This describes something like Red Hat Network.
e. Access a web server

This describes a web browser like Firefox.

3. Which of the following are standard SELinux context types?

(select one or more of the following...)

- a. selinux_type

This is non-existent.
b. object_r

This is an SELinux role.
c. httpd_sys_content_t
d. tmp_t
e. user_u

This is an SELinux context user.



Practice Quiz

SELinux Modes

1. SELinux permissive mode allows logging, but not protection.

2. SELinux enforcing mode protects the system.
3. Which of the following are valid SELinux modes?

(select one or more of the following...)

- a. enforcing
- b. testing
- c. permissive
- d. disabled
- e. logging



Practice Exercise

Changing Enforcing and Permissive Modes

1. On serverX, change the default SELinux mode to permissive and reboot.

Modify the **/etc/sysconfig/selinux** file so the line appears as follows:

```
SELINUX=permissive
```

Reboot your virtual server:

```
[root@serverX ~]# init 6
```

2. After reboot, verify the system is in permissive mode.

```
[root@serverX ~]# getenforce  
Permissive
```

3. Change the default SELinux mode to enforcing.

Modify the **/etc/sysconfig/selinux** file so the line appears as follows:

```
SELINUX=enforcing
```

4. Change the current SELinux mode to enforcing.

```
[root@serverX ~]# setenforce 1  
[root@serverX ~]# getenforce  
Enforcing
```

**Practice Exercise**

Correcting SELinux File Contexts

You have been asked to adjust your remote machine's DNS configuration to exactly match the configuration from your desktop machine. You decide the easiest way is to copy the file **/etc/resolv.conf** from the local machine to the remote machine.

1. Transfer the **/etc/resolv.conf** file from your desktop machine to *root*'s home directory on serverX.

```
scp /etc/resolv.conf root@serverX:
```

2. Shell into serverX as **root**. All of the following steps should occur on your server.

3. Observe the SELinux context of the initial **/etc/resolv.conf**.

```
ls -Z /etc/resolv.conf
```

Original **/etc/resolv.conf** context: system_u:object_r:net_conf_t:s0

4. Move **resolv.conf** from *root*'s home directory to **/etc/resolv.conf**.

```
mv /root/resolv.conf /etc
```

5. Observe the SELinux context of the newly copied **/etc/resolv.conf**.

```
ls -Z /etc/resolv.conf
```

New **/etc/resolv.conf** context: unconfined_u:object_r:admin_home_t:s0

6. Restore the SELinux context of newly positioned **/etc/resolv.conf**.

```
restorecon /etc/resolv.conf
```

7. Observe the SELinux context of the restored **/etc/resolv.conf**.

```
ls -Z /etc/resolv.conf
```

Restored **/etc/resolv.conf** context: system_u:object_r:net_conf_t:s0

**Practice Quiz**

Monitoring SELinux Violations

1. What file contains log entries providing unique identifiers for SELinux violations? /var/log/audit/audit.log
2. Given the UUID of an SELinux violation, what command generates a text report of the problem? sealert -l UUID

 Test

Criterion Test

Exercise

Managing SELinux

Before you begin...

Before you begin, run the `lab-setup-selinux` command on desktopX

1. Login to serverX as **student**. Open a terminal and switch to the **root** user.
2. Copy the `web_content.tgz` archive from `instructor:/var/ftp/pub/materials` to `/tmp`.

```
[root@serverX ~]# cp /net/instructor/var/ftp/pub/materials/web_content.tgz /tmp
```

3. Extract the archive into `/tmp`.

```
[root@serverX ~]# cd /tmp  
[root@serverX tmp]# tar -xvf web_content.tgz
```

4. Move the extracted directory to `/var/www/html`.

```
[root@serverX tmp]# mv web_content /var/www/html/  
[root@serverX tmp]# cd
```

5. Start the web service.

```
[root@serverX ~]# service httpd start
```

6. Try to observe the new directory with your web browser by visiting the URL `http://serverX/web_content`.

```
[root@serverX ~]# elinks -dump http://serverX/web_content
```

7. Search your system for the UUIDs of any SELinux violations your attempt to browse the newly installed content might have generated.

```
[root@serverX ~]# cat /var/log/messages | grep 'sealert -l'
```

8. Generate text reports for the violations.

```
[root@serverX ~]# sealert -l UUID > ~/httpd_selinux.log
```

Where *UUID* is the UUID given in `/var/log/messages`

9. Follow the report's advice to restore the SELinux contexts of the newly installed content.

Find the **Fix Command** section in `~/httpd_selinux.log`

```
[root@serverX ~]# restorecon -Rv /var/www/html/web_content/
```

10. Confirm that you can view the material from your web browser by visiting the URL *http://serverX/web_content*.

```
[root@serverX ~]# elinks -dump http://serverX/web_content
```

Managing Simple Partitions and File Systems



Practice Quiz

Add a New File System

1. Identify a disk that has some free space `fdisk -cu`
2. Create a new partition on that disk `fdisk -cu /dev/device`
3. Update the kernel partition table `reboot`
4. Create a file system on the partition `mkfs -t ext4 /dev/device`
5. Determine the UUID of the file system `blkid /dev/device`
6. Create a mount point `mkdir /directory`
7. Add an entry to the file system table file Add an entry to /etc/fstab like the following:
`UUID=cb79b7d0-dc14-4402-8465-6857346c9a53 /directory ext4 defaults 1 2`
8. Mount the file system `mount -a`



Practice Resequencing Exercise

Create Encrypted File System

For each of the file or directory names below, write down the number of its definition from the list at the bottom.

- | | |
|----------|--|
| <u>1</u> | Create a new partition |
| <u>4</u> | Create an ext4 file system |
| <u>2</u> | Format the new partition for encryption |
| <u>6</u> | Mount the file system on the unlocked device |
| <u>8</u> | Create an entry in /etc/fstab |
| <u>5</u> | Create a directory to use as a mount point |
| <u>3</u> | Unlock the encrypted partition |
| <u>7</u> | Create an entry in /etc/crypttab |
| <u>9</u> | Make LUKS aware of the password file |
-
1. **fdisk**
 2. **cryptsetup luksFormat /dev/vdaN**
 3. **cryptsetup luksOpen /dev/vdaN secret**
 4. **mkfs -t ext4 /dev/mapper/secret**
 5. **mkdir /secret**
 6. **mount /dev/mapper/secret /secret**
 7. **secret /dev/vdaN /password/file**

8. **/dev/mapper/secret /secret ext4 defaults 1 2**
9. **cryptsetup luksAddKey /dev/vdaN /password/file**



Practice Exercise

Create and Use a New Swap Partition

Create and use a new 256 MB swap partition on your virtual server, serverX.

1. Start **fdisk** and create a new partition



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand

2. Change the partition type to **swap**.

Type **t** to change the partition type to "0x82 Linux Swap"

3. Prepare the new partition for use as swap

mkswap /dev/vdaN

(where *N* is the partition number)

4. Determine the UUID

blkid /dev/vdaN

5. Add the new partition to **/etc/fstab**

UUID=uuid swap swap defaults 0 0

6. Determine current amount of swap

swapon -s

7. Activate the new swap

swapon -a

8. Verify newly activated swap

swapon -s

**Test**

Criterion Test

Case Study

Managing Simple Partitions and File Systems

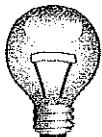
Before you begin...

Make sure to run the **lab-setup-storage** from your desktopX system, which will prepare your serverX system for the lab.

When you are ready, run the **lab-grade-storage** script on serverX to check your work.

Your department wants to use some unallocated storage on the servers. Create additions to your system according to the following list:

- Create a new partition and **ext4** file system that is 400 MB in size. The file system should persistently mount under **/data**.
- Persistently add a swap partition that is 200 MB in size.
- Create an encrypted device with an **ext4** file system that is 256 MB in size and uses the password **testing123**. The system should prompt for the password at boot and mount the file system to **/test**.

**Important**

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

1. Use **fdisk** to add 3 partitions, for the standard file system, the swap partition, and the encrypted file system, respectively.
 - a. Start an interactive session with **fdisk**, and print the existing table.

```
[root@serverX ~]# fdisk -cu /dev/vda
Command (m for help): p
Disk /dev/vda: 21.5 GB, 21474836480 bytes
...
      Device Boot      Start        End      Blocks   Id  System
  /dev/vda1    *        2048     526335      262144   83  Linux
  /dev/vda2       526336    9914367     4694016   8e  Linux LVM
```

- b. In order to avoid later problems caused by a limited number of primary partitions, create an extended partition which spans the remainder of the disk.

```
Command (m for help): n
Command action
```

```

      e   extended
      p   primary partition (1-4)
e
Partition number (1-4): 3
First sector (9914368-41943039, default 9914368): <RETURN>
Using default value 9914368
Last sector, +sectors or +size{K,M,G} (9914368-41943039, default
41943039): <RETURN>
Using default value 41943039
Command (m for help): p

Disk /dev/vda: 21.5 GB, 21474836480 bytes
...

Device Boot      Start        End      Blocks   Id  System
/dev/vda1 *       2048      526335     262144   83  Linux
/dev/vda2         526336     9914367    4694016   8e  Linux LVM
/dev/vda3       9914368     41943039    16014336    5  Extended

```

- c. Add a 400 MB partition for the **ext4** file system.

```

Command (m for help): n
Command action
  l   logical (5 or over)
  p   primary partition (1-4)
l
First sector (9916416-41943039, default 9916416): <RETURN>
Using default value 9916416
Last sector, +sectors or +size{K,M,G} (9916416-41943039, default 41943039): +400M

```

- d. Add a 200 MB partition, and adjust its label appropriately for a swap partition.

```

Command (m for help): n
Command action
  l   logical (5 or over)
  p   primary partition (1-4)
l
First sector (10737664-41943039, default 10737664): <RETURN>
Using default value 10737664
Last sector, +sectors or +size{K,M,G} (10737664-41943039, default 41943039): +200M
Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): 82
Changed system type of partition 6 to 82 (Linux swap / Solaris)

```

- e. Add a 256 MB partition for the encrypted **ext4** partition.

```

Command (m for help): n
Command action
  l   logical (5 or over)
  p   primary partition (1-4)
l
First sector (11149312-41943039, default 11149312): <RETURN>
Using default value 11149312
Last sector, +sectors or +size{K,M,G} (11149312-41943039, default 41943039): +256M

```

- f. Admire your work, and exit committing changes.

```
Command (m for help): p
Disk /dev/vda: 21.5 GB, 21474836480 bytes
...
Device Boot Start End Blocks Id System
/dev/vda1 * 2048 526335 262144 83 Linux
/dev/vda2 526336 9914367 4694016 8e Linux LVM
/dev/vda3 9914368 41943039 16014336 5 Extended
/dev/vda5 9916416 10735615 409600 83 Linux
/dev/vda6 10737664 11147263 204800 82 Linux swap / Solaris
/dev/vda7 11149312 11673599 262144 83 Linux
```

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

- g. Noting the warning, confirm that the kernel is not yet aware of the newly created disk partitions.

```
[root@serverX ~]# cat /proc/partitions
major minor #blocks name
8 0 20971520 vda
8 1 262144 vda1
8 2 4694016 vda2
253 0 557056 dm-0
253 1 3473408 dm-1
253 2 262144 dm-2
```

- h. Reboot serverX.
i. Once serverX has rebooted, confirm that the kernel is now aware of the new partitions.

```
[root@serverX ~]# cat /proc/partitions
major minor #blocks name
8 0 20971520 vda
8 1 262144 vda1
8 2 4694016 vda2
8 3 1 vda3
8 5 409600 vda5
8 6 204800 vda6
8 7 262144 vda7
253 0 557056 dm-0
253 1 3473408 dm-1
253 2 262144 dm-2
```

2. Format the **ext4** partition, decide upon a mount point (we will use **/data**), and configure the system to automatically mount the partition on bootup.

- a. Lay down the **ext4** file system on the 400MB partition, and determine the resulting UUID.

```
[root@serverX ~]# mkfs.ext4 /dev/vda5
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
...
[root@serverX ~]# blkid /dev/vda5
/dev/vda5: UUID="b06c6e49-c056-4af0-a6e1-ee79602f5bf8" TYPE="ext4"
```

- b. Add an entry (line) to **/etc/fstab**, which associates the file system (identified by UUID) with the intended mount point.

```
UUID=b06c6e49-c056-4af0-a6e1-ee79602f5bf8 /data ext4 defaults 1 2
```

- c. Create the mount point (directory), and use the **mount** command to mount all defined mount points (thereby confirming the correctness of your newly created **/etc/fstab** entry).

```
[root@serverX ~]# mkdir /data
[root@serverX ~]# mount -a
```

- d. Use the **df** command to confirm the newly created file system is mounted appropriately.

```
[root@serverX ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vgsrv-root
                  3.3G  2.1G  1.1G  67% /
tmpfs           246M   88K  246M   1% /dev/shm
/dev/vda1        248M   30M  206M  13% /boot
/dev/mapper/vgsrv-home
                  248M   11M  226M   5% /home
/dev/vda5        388M   11M  358M   3% /data
```

3. Initialize the swap partition, and arrange for it to be activated on bootup.

- a. Initialize the swap partition.

```
[root@serverX ~]# mkswap /dev/vda6
Setting up swapspace version 1, size = 204796 KiB
no label, UUID=c3f56fc4-cd69-48fe-bd87-fef41a1db3ae
```

- b. Add an entry (line) to **/etc/fstab**, which identifies the swap partition by UUID.

```
UUID=c3f56fc4-cd69-48fe-bd87-fef41a1db3ae swap swap defaults 0 0
```

- c. Activate the partition, noting active swap partitions before and after.

```
[root@serverX ~]# cat /proc/swaps
```

```
Filename                                Type      Size   Used  Priority
/dev/dm-0                               partition 557048  0     -1
[root@serverX ~]# swapon -a
swapon: /dev/mapper/vgsrv-swap: swapon failed: Device or resource busy
[root@serverX ~]# cat /proc/swaps
Filename                                Type      Size   Used  Priority
/dev/dm-0                               partition 557048  0     -1
/dev/vda6                               partition 204792  0     -2
```

(Note the "busy" complaint is referring to the already active swap partition.)

4. Create and configure the encrypted partition.

- Scrub the partition with random data. (In lab, this step can take a significant amount of time, and can be safely skipped.)

```
[root@serverX ~]# cat /dev/urandom > /dev/vda7
          ( ... significant delay ... )
cat: write error: No space left on device
```

- Initialize the LUKS encryption layer.

```
[root@serverX ~]# cryptsetup luksFormat /dev/vda7
WARNING!
=====
This will overwrite data on /dev/vda7 irreversibly.

Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: testing123
Verify passphrase: testing123
```

- Open the LUKS device, choosing an arbitrary name for accessing the plaintext layer (we will use `test_plaintext`)

```
[root@serverX ~]# cryptsetup luksOpen /dev/vda7 test_plaintext
Enter passphrase for /dev/vda7: testing123
```

- Configure the `/etc/crypttab` file to automatically open the device on bootup, prompting the user for a password.

```
[root@serverX ~]# echo "test_plaintext /dev/vda7" >> /etc/crypttab
```

- Create the `ext4` file system.

```
[root@serverX ~]# mkfs.ext4 /dev/mapper/test_plaintext
mke2fs 1.41.12 (17-May-2010)
...
[root@serverX ~]# blkid /dev/mapper/test_plaintext
/dev/mapper/test_plaintext: UUID="5bf2d39e-cb79-4f4a-a276-2a306ad506c1"
TYPE="ext4"
```

- Establish a mount point (we will use `/test`) and add a line similar to the following to `/etc/fstab`.

```
UUID=5bf2d39e-cb79-4f4a-a276-2a306ad506c1 /test ext4 defaults 1 2
```

- g. Create the mount point, mount all partitions, and confirm that the encrypted file system was mounted.

```
[root@serverX ~]# mkdir /test [root@serverX ~]#mount -a  
[root@serverX ~]# df -h  
Filesystem           Size  Used Avail Use% Mounted on  
...  
/dev/mapper/test_plaintext  
      246M  6.1M  228M   3% /test  
...
```

Controlling Access to Files

Collaborative Directory Permissions

In the quiz below, use both POSIX ACLs and standard permissions, as appropriate, to solve these problems.

- Given a normal directory, where the owning *user* has **rwx** permissions, the owning *group* has **rwx** permissions, and *other* has **---** permissions, what command would grant a second group **r-x** permissions without changing the permissions of the existing owning group or *other*?

setfacl -m g:group:r-x /directory

- What command would automatically grant that second group read-write access to any newly created regular files in that directory?

setfacl -m d:g:group:rw /directory

- Bonus question.* What command would automatically set the owning *group* as the owning group of any newly created files in that directory?

chmod g+s /directory, or **chmod 2770 /directory** assuming the original directory permissions as listed in the first question.

You need to ensure the directory has the set-GID permission. It turns out that there is not a way to do this with **setfacl**.



Test

Criterion Test

Case Study

Using ACLs to Grant and Limit Access

This lab uses users and groups created earlier on serverX. If you do not already have the users and groups defined, run **lab-add-users** on serverX.

Graduate students need a collaborative directory titled **/opt/research**, where they can store generated research results. Only members of the groups **profs** and **grads** should be able to create new files in the directory, and new files should have the following properties:

- The directory should be owned by user **root**.
- New files should be group owned by the group **grads**.
- Professors (members of the group **profs**) should automatically have read/write access to new files.
- Summer interns (members of the group **interns**) should automatically have read-only access to new files.

- Other users (not a member of groups **profs**, **grads**, or **interns**) should not be able to access the directory and its contents at all.

See the Solutions appendix when you are done to check your solution and to see some possible approaches.

- One possible solution is:

```
[root@serverX ~]# mkdir /opt/research
[root@serverX ~]# chgrp grads /opt/research/
[root@serverX ~]# chmod 2770 /opt/research/
[root@serverX ~]# setfacl -m g:profs:rwx /opt/research/
[root@serverX ~]# setfacl -m g:interns:rx /opt/research/
[root@serverX ~]# setfacl -m d:g:profs:rw /opt/research/
[root@serverX ~]# setfacl -m d:g:interns:r /opt/research/
```

The first three lines create **/opt/research** and make sure that it is owned by user **root**, group **grads** and that the owning user and group have and read, write, and access files, and that **grads** will own files created in the directory (set-GID is on). The next two lines grant appropriate permissions for group **profs** and group **interns** on the directory using ACLs. The next two lines grant appropriate permissions for group **profs** and group **interns** on new files created in the directory through default ACLs.



Note

A more sophisticated and somewhat better solution is to replace the last two lines of the solution above as follows:

```
[root@serverX ~]# setfacl -m d:g:profs:rwx /opt/research/
[root@serverX ~]# setfacl -m d:g:interns:rx /opt/research/
[root@serverX ~]# setfacl -m d:o::-- /opt/research/
```

By adding execute permission to the default ACLs, members of groups **profs** and **interns** will automatically be able to access newly created subdirectories of **/opt/research**. (Regular files do not get the effective execute permission automatically because the default *ACL mask* on a new regular file is **rw-**.

The last line of the improved solution above removes all permissions from *other* for newly created files. It is not strictly necessary because the permissions on the **/opt/research** directory already blocks all access for *other* to the directory and its contents, but it is shown here as a useful example.

Use **ls** and **getfacl** to confirm your solution.

```
[root@serverX ~]# getfacl /opt/research/
getfacl: Removing leading '/' from absolute path names
# file: opt/research/
# owner: root
# group: grads
# flags: -s-
user::rwx
```

```
group::rwx
group:interns:r-x
group:profss:rwx
mask::rwx
other::---
default:user::rwx
default:group::rwx
default:group:interns:r-x
default:group:profss:rwx
default:mask::rwx
default:other::---
[root@serverX ~]# date > /opt/research/foo
[root@serverX ~]# ls -l /opt/research/foo
-rw-rw---+ 1 root grads 29 Dec 23 12:31 /opt/research/foo
[root@serverX ~]# getfacl /opt/research/foo
getfacl: Removing leading '/' from absolute path names
# file: opt/research/foo
# owner: root
# group: grads
user::rw-
group::rwx          #effective:rwx
group:interns:r-x   #effective:r--
group:profss:rwx    #effective:rwx
mask::rw-
other::---
```

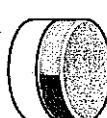
Managing Flexible Storage with Logical Volume Manager



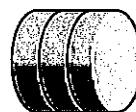
Practice Quiz LVM Components

1. Fill in the following graphic with the names of the components.

Unused Space



4. _____



3. _____



2. _____



1. _____

1. Physical storage

2. Physical volumes

3. Volume group

4. Logical volume

2. What are the smallest pieces (chunks or blocks) of the physical volume?

Physical extents

3. What is the smallest size you could make a logical volume?

The size of a single physical extent

4. What references the physical extents of a logical volume?

Logical extents



Practice Exercise

Implement LVM and Create a Logical Volume

Before you begin...

Make sure to run the **lab-setup-lvm** from your desktopX system, which will prepare your serverX system for the practice exercise.

All of these steps will be performed on serverX.

1. Create a new partition of 512 MB and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

```
[root@serverX ~]# fdisk -cu /dev/vda
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (9914368-12582911, default 9914368): Enter
Using default value 9914368
Last sector, +sectors or +size{K,M,G} (9914368-12582911, default 12582911): +512M

Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 8e
```

```
Changed system type of partition 3 to 8e (Linux LVM)
```

```
Command (m for help): w
The partition table has been altered!
... Output Omitted ...
[root@serverX ~]# reboot
[root@serverX ~]# pvcreate /dev/vda3
Physical volume "/dev/vda3" successfully created
```

2. Create a Volume Group named **shazam** using the Physical Volume created in the previous step.

```
[root@serverX ~]# vgcreate shazam /dev/vda3
Volume group "shazam" successfully created
```

3. Create and format with **ext4**, a new Logical Volume of 256 MB called **/dev/shazam/storage**.

```
[root@serverX ~]# lvcreate -n storage -L 256M shazam
Logical volume "storage" created
[root@serverX ~]# mkfs -t ext4 /dev/shazam/storage
mke2fs 1.41.12 (17-May-2010)
... Output Omitted ...
```

4. Modify your system such that **/dev/shazam/storage** is mounted at boot time as **/storage**.

```
[root@serverX ~]# mkdir /storage
```

Add the following line to the bottom of **/etc/fstab** on serverX:

```
/dev/shazam/storage    /storage    ext4    defaults 1 2
```

Growing a Logical Volume Basic Steps

1. Verify available space in the volume group
2. Extend the logical volume
3. Extend the file system



Practice Exercise

Extend a Logical Volume

All of these steps will be performed on serverX.

1. Determine the amount of free space in Volume Group **shazam**.

```
[root@serverX ~]# vgdisplay shazam
```

```
--- Volume group ---
VG Name          shazam
System ID
Format          lvm2
... Output Omitted...
VG Size         508.00 MiB
PE Size          4.00 MiB
Total PE        127
Alloc PE / Size 64 / 256.00 MiB
Free  PE / Size 63 / 252.00 MiB
VG UUID         accvy0-3bhi-9jK8-eg7u-44AL-ARRg-r7Uo30
```

2. Extend the logical volume **/dev/shazam/storage** with *half* the available extents in the volume group using command-line tools.

```
[root@serverX ~]# lvextend -l +32 /dev/shazam/storage
Extending logical volume storage to 384.00 MiB
Logical volume storage successfully resized
```

3. Extend the file system mounted on **/storage** using command-line tools.

```
[root@serverX ~]# resize2fs /dev/shazam/storage
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/shazam/storage is mounted on /storage; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 2
Performing an on-line resize of /dev/shazam/storage to 393216 (1k) blocks.
The filesystem on /dev/shazam/storage is now 393216 blocks long.
```

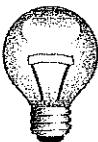


Practice Exercise

Extend a Volume Group

All of these steps will be performed on serverX.

1. Create a new 512 MB partition and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

```
[root@serverX ~]# fdisk -cu /dev/vda

Command (m for help): n
Command action
      e   extended
      p   primary partition (1-4)
e
Selected partition 4
First sector (10962944-12582911, default 10962944): Enter
Using default value 10962944
```

```
Last sector, +sectors or +size{K,M,G} (10962944-12582911, default 12582911): Enter
Using default value 12582911

Command (m for help): n
First sector (10964992-12582911, default 10964992): Enter
Using default value 10964992
Last sector, +sectors or +size{K,M,G} (10964992-12582911, default 12582911): +512M

Command (m for help): t
Partition number (1-5): 5
Hex code (type L to list codes): 8e
Changed system type of partition 5 to 8e (Linux LVM)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@serverX ~]# reboot
[root@serverX ~]# pvcreate /dev/vda5
Physical volume "/dev/vda5" successfully created
```

2. Extend the Volume Group **shazam** by adding the Physical Volume created in the previous step.

Use **vgextend** to extend the volume group.

```
[root@serverX ~]# vgextend shazam /dev/vda5
Volume group "shazam" successfully extended
```

Determining Snapshot Size

1. Expected rate of change
2. Required snapshot time



Practice Exercise

Creating an LVM Snapshot

Compare the contents of our existing logical volume, **/dev/shazam/storage**, to a new snapshot volume, **/dev/shazam/storagesnap**, while making changes to the original volume.

All of these steps will be performed on serverX.

1. Copy the file **/usr/share/dict/linux.words** to **/storage** so you have some data to compare.

```
[root@serverX ~]# cp /usr/share/dict/linux.words /storage
```

2. Create a new 20 MB snapshot logical volume of **/dev/shazam/storage** called **storagesnap**.

```
[root@serverX ~]# lvcreate -n storagesnap -L20M -s /dev/shazam/storage
Logical volume "storagesnap" created
```

3. Manually mount **/dev/shazam/storagesnap** read only at **/storagesnap**

```
[root@serverX ~]# mkdir /storagesnap
[root@serverX ~]# mount -o ro /dev/shazam/storagesnap /storagesnap
```

4. List the contents of **/storagesnap** and note that they are the same as **/storage**.

```
[root@serverX ~]# ls /storagesnap /storage
/storage:
linux.words  lost+found

/storagesnap:
linux.words  lost+found
```

5. Delete the file **/storage/linux.words** and note that it still exists in **/storagesnap**.

```
[root@serverX ~]# rm /storage/linux.words
rm: remove regular file `/storage/linux.words'? y
[root@serverX ~]# ls /storagesnap /storage
/storage:
lost+found

/storagesnap:
linux.words  lost+found
```

6. Clean up: unmount **/storagesnap**, remove the directory, and delete the **storagesnap** logical volume.

```
[root@serverX ~]# umount /storagesnap
[root@serverX ~]# rmdir /storagesnap
[root@serverX ~]# lvremove /dev/shazam/storagesnap
Do you really want to remove active logical volume storagesnap? [y/n]: y
Logical volume "storagesnap" successfully removed
```



Test

Criterion Test

Case Study

LVM Case Study

Before you begin...

Make sure to run the **lab-setup-lvm** from your desktopX system, which will prepare your serverX system for the lab.

```
[root@serverX ~]# lab-setup-lvm
```

When you are ready, run the **lab-grade-lvm** script on serverX to check your work.

Allison needs to store data for her business. Her customer database is currently 256 MB in size. The data in the database changes about 10 MB per hour on a typical day. The backup software takes 10 minutes to complete a full run.

Create a new Volume Group called **allison** with enough space for both a 512 MB volume and a snapshot of that volume for the backup software.

Once the volume group is created, create within it a 512 MB logical volume for Allison's customer database called **custdb**. Also create a snapshot volume of Allison's customer database called **custdbsnap** for her backup software.

1. Create a new 1 GB partition using **fdisk**, and reboot the machine for the changes to take effect.

```
[root@serverX ~]# fdisk -cu /dev/vda
Command (m for help): p
Disk /dev/vda: 21.5 GB, 21474836480 bytes
...
      Device Boot      Start        End      Blocks   Id  System
/dev/vda1    *        2048     526335     262144   83  Linux
/dev/vda2       526336    9914367    4694016   8e  Linux LVM

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (9914368-41943039, default 9914368):
Using default value 9914368
Last sector, +sectors or +size{K,M,G} (9914368-41943039, default 41943039): +1G

Command (m for help): w
...
```

The kernel still uses the old table. The new table will be used at the next reboot or after you run partprobe(8) or kpartx(8). Syncing disks.

Reboot serverX.

2. Prepare the newly create partition for use with LVM.

```
[root@serverX ~]# pvcreate /dev/vda3
Physical volume "/dev/vda3" successfully created
```

3. Create a new volume group called **allison** using the new partition.

```
[root@serverX ~]# vgcreate allison /dev/vda3
```

```
Volume group "allison" successfully created
```

4. Create a 512 MB logical volume for Allison's customer database.

```
[root@serverX ~]# lvcreate -n custdb -L512M allison
Logical volume "custdb" created
```

5. Create a 10 MB snapshot volume of Allison's customer database

```
[root@serverX ~]# lvcreate -n custdbsnap -L10M -s /dev/allison/custdb
Rounding up size to full physical extent 12.00 MiB
Logical volume "custdbsnap" created
```

6. When you are ready, run the **lab-grade-lvm** script on serverX to check your work.

Control the Boot Process

Write a definition for each of these key terms:

1. bootloader
a program that loads an operating system kernel into memory and executes it.
2. GRUB
GRand Unified Bootloader, the bootloader used by Red Hat Enterprise Linux



Practice Performance Checklist

Booting an Alternate Kernel

Perform all of the following steps on serverX.

- Configure **yum** to point to the **Errata** repository on the **instructor** machine with the following command:

```
[root@serverX ~]# wget http://instructor/pub/gls/errata.repo -O /etc/yum.repos.d/errata.repo
```

- Install the **kernel** update that is available. This will take over 3 minutes to install.

```
[root@serverX ~]# yum update -y kernel
```

- Boot into the new kernel.

Gracefully reboot the system with the **reboot** command. GRUB will be configured to boot using the new kernel by default.

- Reboot and choose the old kernel.

Again, reboot the system with the **reboot** command. When the GRUB countdown appears, hit the **Esc** key to display the GRUB menu. Use the arrow keys to select the lower-numbered kernel then hit **Enter** to begin the boot process.

Runlevel Definitions

1. Write a definition for this key term:

runlevel

The state of a system that defines which services are available.

2. What are each of these runlevels typically used for?

runlevel 5 - Graphical desktop

runlevel 3 - Multi-user non-GUI

runlevel 1 - Single-user mode, no network (similar to "Safe Mode" in Windows)



Practice Performance Checklist

Changing the root Password

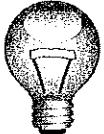
This timed drill is designed to give you practice changing the root password on a system with an unknown root password.

Perform all of the following steps on serverX.

- Begin by running the **lab-setup-bootbreak-4** script. This will change the root password to something unknown and mark the current time.

```
[root@serverX ~]# lab-setup-bootbreak-4
```

- Get into the system and reset the root password to **redhat**.



Important

At the release of Red Hat Enterprise Linux 6, there was an SELinux bug which blocked the **passwd** command in single-user mode (#644820). If you have the original *selinux-policy* package installed, you must run the **setenforce 0** command in runlevel 1 before the **passwd** command for it to work. After changing the password you should run **setenforce 1** again to put SELinux back in enforcing mode.

Interrupt the GRUB countdown (Esc key). Use "e" to edit current configuration. Select **kernel** line to correct with arrow keys. Type "e" again to edit the current line, appending a "space" and "**single**". Type "b" to boot with the current changes.

```
# setenforce 0
# passwd
Changing password for user root.
New password: redhat
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password: redhat
passwd: all authentication tokens updated successfully.
# setenforce 1
```

- Once you have reset the password, change the system into runlevel 5 and run the **lab-grade-bootbreak-4** script.

```
# init 5
```

```
[root@serverX ~]# lab-grade-bootbreak-4
```

- View the feedback from the script to ensure you completed the task correctly. The grading script will display a time, write it down.
- Repeat the process again at least five times.
- Circle your best time.



Practice Performance Checklist

Getting Past a GRUB Misconfiguration

Perform all of the following steps on serverX.

- Run the **lab-setup-bootbreak-5** script to introduce an issue with the boot process.
- Fix the issue so the system can boot and you can log in.

The problem to be solved is a misspelled GRUB directive or file name. The boot process must be interrupted so hit the **Esc** when the GRUB countdown appears. Type the **e** command to display the GRUB configuration that is being used and identify the error. Use the editing commands displayed below the GRUB menu to correct the typographical error. Type the **b** command to accept the changes and begin the boot process.



Practice Performance Checklist

Making Persistent GRUB Changes

Perform all of the following steps on serverX.

- Reboot and confirm the issue from the previous problem is not persistently fixed. You will need to apply the fix as before to boot the system.
Repeat the steps you performed in the previous problem to get the system booted enough to display a shell prompt.
- Edit the configuration file to fix the issue permanently.

Edit the GRUB configuration file with your favorite text editor to correct the typographical error permanently:

```
[root@serverX ~]# vim /boot/grub/grub.conf
```

- Revert to the older kernel. Ensure that when you reboot, the older kernel is the default kernel.

Modify the GRUB configuration file so the **default** directive selects the older kernel. Remember that GRUB boot stanzas begin numbering from 0.

Kernel Arguments Search & Learn

1. Install the **kernel-doc** package.
2. Reference the material in **kernel-parameters.txt** found in the **/usr/share/doc/kernel-doc*/Documentation/** directory.
3. Each team must research and summarize the following kernel parameters:

Team 1:

- console

Console redirection: **console=ttyS0**

Team 2:

- enforcing

SELinux control: **enforcing=0|1**

enforcing=1 sets Enforcing mode. **enforcing=0** sets Permissive mode.

- selinux

SELinux control: **selinux=0|1**

selinux=0 disables SELinux entirely. **selinux=1** enables SELinux.

Team 3:

- init

Init process: **init=/bin/bash**

Team 4:

- root

Mount root file system:

root=/path/to/root/volume

- ro

ro - read-only root file system

- rw

rw - read-write root file system



Practice Performance Checklist

Passing Kernel Arguments

Earlier we had to turn off SELinux enforcing mode to change the **root** password in runlevel 1. There is a kernel parameter that allows us to do that without using commands from the shell. Perform the following steps on serverX.

- Before you reboot your serverX machine, check its default SELinux status by executing the **getenforce** command. Confirm the system normally boots into Enforcing mode.

```
[root@serverX ~]# getenforce  
Enforcing
```

- Reboot your serverX machine and pass **enforcing=0** to the kernel when the system boots.

Stop the GRUB countdown with the **Esc** key then used the **a** command to add the extra **enforcing=0** argument to the kernel. Type **Enter** to accept the changes and begin the boot process.
- Once serverX finishes booting, check its SELinux status. Confirm the system booted into Permissive mode.

```
[root@serverX ~]# getenforce  
Permissive
```



Practice Performance Checklist

Changing the Default Runlevel

You are configuring a new system that you will be accessing remotely. The system is currently booting into runlevel 5 by default, but this machine will be housed in a data center where you will only log into it remotely. Perform the following steps on serverX.

- Change serverX to boot to runlevel 3 by default.

Change the line in **/etc/inittab** to the following:

```
id:3:initdefault:
```

- Reboot serverX.

```
[root@serverX ~]# reboot
```

- You have successfully completed this lab if serverX boots into textual mode without human interaction.



Test

Criterion Test

Exercise

Bad Brian Blowup Recovery

Before you begin...

Run **lab-setup-bootbreak** on desktopX to reset serverX back to its original state.

Brian was a summer intern who acted as a system administrator for one of your critical servers, serverX. Your company's strained relationship with him finally blew up and resulted in his immediate firing. Sadly, when Bad Brian went out the door he took the root password for serverX with him.

You have been assigned the responsibility of getting control of serverX back:

1. Run the **lab-setup-bootbreak-6** script on serverX to prepare it for this lab exercise. This will assign your system with an unknown root password and reboot the system.
2. Set the root password to **redhat**.

Because the system won't initialize it will be necessary to boot into runlevel 1 and fix the misconfigured ram disk entry and root's password.

1. Intercept grub during the initialization process by hitting any key when the grub menu displays.
2. Highlight the title line of choice and hit **e** to edit.
3. Highlight the ramdisk line and hit **e** to edit.
4. Modify the ram disk line by removing **-BROKEN** from the entry, hit **<ENTER>**.
5. Highlight the kernel line then hit **e** to edit.
6. Add a space and the number one ("1") at the end of the kernel line, hit **<ENTER>**, then hit **b** to boot the system using your temporary modifications to grub.
7. When the shell prompt displays run **passwd** and set root's password to **redhat**.

(Due to a bug in early releases of Red Hat Enterprise Linux 6, the single user mode shell did not have the correct SELinux context to execute the **passwd** command. Although a fix was soon released, the solutions below temporarily suspend and restore SELinux as a reminder of this issue.)

```
Telling INIT to go to single user mode.  
[root@localhost /]# setenforce 0  
[root@localhost /]# passwd
```

```
Changing password for user root.  
New password: redhat  
BAD PASSWORD: it is based on a dictionary word  
BAD PASSWORD: is too simple  
Retype new password: redhat  
passwd: all authentication tokens updated successfully.  
[root@localhost ~]# setenforce 1
```

8. Type **exit** to boot the system into multiuser mode, bringing up the network so additional fixes can be made.
3. Install the kernel update, but configure the system so the old kernel will continue to be used by default.

```
[root@server1 ~]# yum update kernel  
...  
Dependencies Resolved  
=====  
| Package          | Arch | Version | Repository | Size |  
=====  
| Installing:      |       |          |            |       |  
|   kernel         | x86_64 | 2.6.32-71.7.1.el6 | Updates | 22 M |  
| Updating for dependencies:  
|   kernel-firmware | noarch | 2.6.32-71.7.1.el6 | Updates | 1.1 M |  
  
Transaction Summary  
=====  
Install    1 Package(s)  
Upgrade    1 Package(s)  
  
Total download size: 23 M  
Is this ok [y/N]: y  
...  
Complete!
```

Edit **/boot/grub/grub.conf** changing *default=0* to match the index of the old kernel stanza, such as *default=1*. (Recall that the first stanza is considered stanza 0).

NOTE: You should also inspect the ram disk line to ensure that it is correct.

4. Pass the **selinux=1** argument to the kernel at boot time.
Edit **/boot/grub/grub.conf**, adding **selinux=1** to the end of the kernel line past all other arguments.
5. Set runlevel 3 as the default.
Edit **/etc/inittab**, changing 5 to 3 in the second field of the only active line.
6. Once your system is booted, run the **lab-grade-bootbreak-6** script on serverX to determine how well you did.

Tuning and Maintaining the Kernel



Practice Exercise

Loading Modules and Setting Default Parameters

You have been asked to load the `nf_conntrack_ftp` kernel module, and configure it appropriately for an FTP server listening on TCP port 21 and on 8021.

The following commands are to be run on your serverX.

1. Use the locate command to convince yourself that the `nf_conntrack_ftp` module is supported by your kernel.

```
locate track_ftp
```

2. Load the FTP connection track module.

```
modprobe nf_conntrack_ftp
```

3. Convince yourself it's loaded.

```
lsmod
```

4. Unload the FTP connection track module.

```
modprobe -r nf_conntrack_ftp
```

5. Convince yourself it's unloaded.

```
lsmod
```

6. In addition to the standard port 21, you are planning to run an FTP server on the non-standard port 8021. Examine options which might let you specify non-standard ports.

```
modinfo nf_conntrack_ftp
```

7. Configure a file `/etc/modprobe.d/local.conf` which implements this option upon loading

```
options nf_conntrack_ftp ports=21,8021
```

1. What (familiar) command performs a kernel update? yum update
2. New kernels are installed, not updated. Because every file owned by the kernel package is versioned, or resides in a versioned directory, RPM is willing to have concurrent versions installed.
3. By default, when "updating" a kernel, `yum` will keep a total of 3 versions installed, automatically removing any older version.
4. In order to use your new kernel, you must reboot your machine.

5. While the machine will automatically reboot to your upgraded kernel, you may still choose an older kernel from the GRUB bootloader's menu.
6. If removing a kernel manually, you must specify not only the package name (kernel), but also the version number.

Manage Virtual Machines



Practice Performance Checklist

Virtual Guest Installation

In this lab you will install a new virtual machine with Red Hat Enterprise Linux using **virt-manager** and the graphical installer. Once you have successfully completed the lab you will need to remove both the virtual machine and its logical volume to reclaim system resources needed for other labs.

Perform the following steps on desktopX:

- Gracefully shutdown your serverX virtual machine (**vserver**) to reclaim system CPU and RAM resources.

Launch **virt-manager** by selecting Applications → System Tools → Virtual Machine Manager. Right-click on the icon for the **vserver** virtual machine then select **Shut Down** → **Shut Down**.

- Create a logical volume 10 GB in size from the **vol0** volume group and name it **guest**.

```
[root@desktopX ~]# lvcreate -n guest -L 10G vol0
```

- Create a Red Hat Enterprise Linux 6 virtual machine with the following characteristics:

- Name = guest
- Install media = network install from <http://instructor.example.com/pub/rhel6/dvd>
- Memory (RAM) = 768 MB
- CPUs = 1
- Storage device = the logical volume created in the previous step

Within **virt-manager**, right-click on the **localhost (QEMU)** item and select **New**. When the “New VM” dialog box appears, type **guest** for the name and choose the **Network Install (HTTP, FTP, or NFS)** radio button for installation method. Click the **Forward** button when you are ready to proceed.

Type <http://instructor.example.com/pub/rhel6/dvd> in the URL field. Click the **Forward** button when you are ready to proceed. If a warning dialog box appears cautioning about the permissions of **/home/student/.virtinst/boot**, then click **Yes** and move on.

In the next dialog box, select **768 MB** for **Memory (RAM)** and leave the **CPUs** set to 1. Click the **Forward** button when you are ready to proceed.

For storage, select the **Select managed or other existing storage** radio button, then specify the **/dev/vol0/guest** pathname. Click the **Forward** button when you are ready to continue.

After reviewing the final dialog box, click **Finish** to complete the creation of the virtual machine and begin your interaction with the Red Hat installer, Anaconda.

When the text-based menus appear, select the appropriate language and keyboard choices for your locale. Each time choose **OK** to proceed to the next menu. Once the network settings have been specified, the graphical installer will appear. Select **View → Resize to VM** from the **virt-manager** menus.

- When the installation begins, choose your keyboard and language. Build your guest system according to the following specifications:
 - When asked about the Virtio Block Device, choose **Re-initialize all**.
 - Choose the appropriate time zone
 - Assign **redhat** as the root password
 - Choose the Desktop software set
 - Use the defaults for everything else

Click the **Next** button to move beyond the introductory screen.

On the storage screen, make sure the **Basic Storage Devices** radio button is selected and click **Next**. If a **Warning** dialog box appears suggesting the storage needs to be reinitialized, click the **Re-initialize all** button to wipe the virtual machine's drive.

When the network configuration screen appears, leave the default hostname chosen. The network will be configured because a network installation is being performed. Click then **Next** button to continue.

Choose an appropriate timezone and make sure the **System clock uses UTC** checkbox is checked. Click **Next** to continue.

Specify the root password of **redhat** twice then click the **Next** button. When the **Weak Password** dialog box appears, ignore the warning and click the **Use Anyway** button to continue.

Since the problem exercise said to use the default partitioning scheme, click the **Next** button to advance past the disk partitioning screen. Click the **Write changes to disk** button when the warning dialog box appears. You will see the disk get partitioned and formatted at this point.

The software selection screen will appear next. Select the radio button for the **Desktop** software set instead of the default **Basic Server**. Click the **Next** button to continue. After the software dependency checks complete the installation will begin.



Practice Group Exercise

Search & Learn: Virtual Machine Automatic Boot

What steps must you take to configure a virtual guest to automatically start at boot time?

1. Launch Virtual Machine Manager.
2. Double-click on the guest virtual machine profile.
3. Choose View → Details
4. Select Boot Options
5. Check or uncheck the Start virtual machine on host boot up check box and click Apply.
6. Add the following to the /etc/sysconfig/libvirt-guests file:

```
ON_BOOT=ignore
```

Practice Performance Checklist

Configuring Virtual Machines at Boot-time

- Configure the **serverX** (vserver) virtual machine to not start at boot time.
- Configure the **guest** virtual machine to start at boot time.
Launch the Virtual Machine Manager and double-click on the **guest** virtual machine. Choose View → Details and select Boot Options. Ensure the Start virtual machine on host boot up check box is checked and click Apply if necessary. Add **ON_BOOT=ignore** to /etc/sysconfig/libvirt-guests.

Alternately, from the command-line:

```
[root@desktopX ~]# virsh autostart guest  
[root@desktopX ~]# echo 'ON_BOOT=ignore' >> /etc/sysconfig/libvirt-guests
```

- Reboot the physical machine (desktopX).

```
[root@desktopX ~]# reboot
```

- Confirm the **guest** virtual machine started automatically.

Open the Virtual Machine Manager and verify that **guest** shows that it is running.

Alternately, from the command-line:

```
[root@desktopX ~]# virsh list  
 Id Name State  
-----  
 1 guest running
```

- Configure the **guest** virtual machine to not start at boot time.

Uncheck the autostart check box following the steps above, or from the command-line:

```
[root@desktopX ~]# virsh autostart --disable guest
```

- Reboot the physical machine (desktopX).

```
[root@desktopX ~]# reboot
```

- Confirm the virtual machine did not start automatically.

Follow the steps above to ensure that the **guest** virtual machine is not started.

- IMPORTANT:** After you successfully complete the lab, delete the **guest** virtual machine and the logical volume it uses for storage. Those resources will need to be available for the criterion test.

From the GUI:

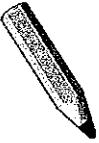
Open the Virtual Machine Manager. If the **guest** virtual machine is running, right-click on the **guest** virtual machine and select Shutdown → Force Off. Right-click on the **guest** virtual machine and select Delete, then click Delete. Remove the logical volume by running **lvremove -f /dev/vol0/guest**.

Warning: it may take a minute or more for the logical volume to be freed. If the **lvremove** command does not work, wait for a minute and try again.

From the command-line:

```
[root@desktopX ~]# virsh destroy guest
[root@desktopX ~]# virsh undefine guest
[root@desktopX ~]# lvremove -f /dev/vol0/guest
```

Warning: it may take a minute or more for the logical volume to be freed. If the **lvremove** command does not work, wait for a minute and try again.



Test

Criterion Test

Case Study

Virtual Workstation for William Wonderboy

William Wonderboy just joined the company as a software developer. He needs a machine of his own to write code and do testing without disturbing the work of others. You have been assigned the task of building a virtual machine for him to use.

Create a virtual machine named **wonderboy** with an LVM storage device named **/dev/vol0/wonderboy**. Use the installation media found at the following URI:

- <http://instructor.example.com/pub/rhel6/dvd>

Mr. Wonderboy's virtual machine must have 768 MB RAM and 10 GB of disk storage.

Use a static IP address of 192.168.0.200+X/24, with a gateway and DNS server of 192.168.0.254. Set the hostname to **hostX.example.com**.

Choose an appropriate time zone. Use **redhat** as the root password.

The virtual disk should be partitioned as follows (you will have to re-initialize the disk):

- 250 MB for **/boot**
- 1 GB of swap space
- 6 GB for **/**
- The rest of the space allocated to **/home**

Choose the **Software Development Workstation** software set.

Once the installation is complete, configure NTP to connect to `instructor.example.com`

Configure this machine to start automatically when the physical host reboots.

1. On desktopX, create a 10 GB logical volume named *wonderboy*.

```
[root@desktopX ~]# lvcreate -L 10G -n wonderboy vol0
```

2. On desktopX, open Virtual Machine Manager.

```
[root@desktopX ~]# virt-manager
```

3. Create a new virtual machine.

- a. Right-click on **localhost(QEMU)**, choose **New**.
- b. In the **Name** field type *wonderboy*.

In **Choose how you would install the operating system**, select **Network Install (HTTP, FTP, or NFS)**.

- c. **Provide the operating system install URL**, use `http://instructor.example.com/pub/rhel6/dvd`.

Hit the **Enter** key and notice that the values next to **OS type** and **Version** will auto-populate.



Note

If you receive a message about the search permissions choose **No** to move forward.

- d. **Modify Memory (RAM)** to read **768**. Leave the remaining values as defaults.

- e. Choose **Select managed or other existing storage**. Browse to or type in **/dev/vol0/wonderboy**. Leave the remaining values as defaults.
 - f. Verify values and click **Finish**.
4. Use Anaconda to install the guest.
 - a. Select Language.
 - b. Select Keyboard Type.
 - c. De-select IPv6 support.
 - d. Select **Basic Storage Devices**.
 - e. Select **Re-initialize all** or, if this is an installation that is over a previous installation, you will be prompted for **Fresh** or **Upgrade Installation**. Choose **Fresh Installation** if that is the case.
 - f. Set **Hostname** to *hostX.example.com*
 - g. Choose **Configure Network**.
 - i. Click on **Wired** tab (if necessary).
 - ii. Highlight **System eth0** and click **Edit**.
 - iii. Click on **IPv4 Settings**.
 - iv. Change **Method** to **Manual**.
 - v. Add, IP address: 192.168.0.X+200.

NOTE: Netmask should automatically fill to **24**.
 - vi. Click the area under the **Gateway** column, then, add 192.168.0.254 for your gateway.
 - vii. Add, **DNS servers**: 192.168.0.254
 - viii. Click **Apply** then close Network Connections.
 - h. Choose appropriate Timezone and check UTC.
 - i. Enter *redhat* as root's password.
 - j. Choose **Create Custom Layout**, and create the following partitions.
 - i. Delete any existing partitions (if necessary).
 - ii. Create: **Standard partition**, mount point /boot, ext4, size 250MB.
 - iii. Create: **Standard partition**, File System Type: swap, size 1024MB.
 - iv. Create: **Standard partition**, mount point /, ext4, size 6144MB.

- v. Create: Standard partition, mount point /home, ext4, select Fill to maximum allowable size.

Select Next to continue.

If prompted with Format Warnings choose Format.

Choose Write changes to disk.

- k. For Boot loader configuration, use defaults.
 - l. For Package Selection, select the Software Development Workstation software set. Leave the remaining values as defaults.
 - m. Monitor the installation.
 - n. When prompted, reboot the system.
5. Because we installed a graphical desktop we will see the first-boot Welcome screen.
- a. Click Forward for License information, then, Forward again.
 - b. For Set Up Software Updates, select No, I prefer to register at a later time.
 - c. When prompted for connecting to Red Hat Network, click on No thanks, I'll connect later.
 - d. Create a User
 - i. Username: *wonderboy*.
 - ii. Full Name: *William Wonderboy*.
 - iii. Password: *redhat*.
 - iv. Confirm password: *redhat*.
 - e. Date and Time
 - i. Click Synchronize date and time over network.
 - ii. Remove all current entries by highlighting an entry then clicking delete.
 - iii. Click on Add, then enter *instructor.example.com*, hit Enter. Then, click Forward.The system will locate NTP server (*instructor.example.com*) and continue.
 - f. You may see Insufficient memory to configure kdump! message click OK.
 - g. Click Finish.
6. Configure your machine to automatically start when the host machine reboots.
- a. In the *virt-manager* window for your guest, choose the View → Details menu item.

- b. Navigate to the *Boot Options* panel.
- c. Select the **Start virtual machine on host boot up** check box and click **Apply**.

Automated Installation of Red Hat Enterprise Linux



Practice Case Study

Creating a Kickstart File with **system-config-kickstart**

Perform this exercise on your desktopX machine as **student**.

In this exercise, you will pretend that your organization is rolling out Red Hat Enterprise Linux-based workstations for its engineers, and you have been tasked with creating a Kickstart file to facilitate this.

You will install **system-config-kickstart**, and use it to create a Kickstart file according to the parameters specified below:

- Choose the appropriate timezone.
- The **root** password should be **redhat**
- The Kickstart should perform a fresh installation from the web server **http://instructor.example.com/pub/rhel6/dvd**
- The system should have a 100 MB, **ext4** filesystem mounted at **/boot**
- The system should have a 512 MB swap partition
- All remaining disk space should be allocated to an **ext4** partition mounted at **/**
- The **eth0** device should start at boot-time and use DHCP for configuration
- Install the **Base** package set
- Have a post-installation script that adds:

ENGINEERING WORKSTATION

to the file **/etc/issue**

- Save the Kickstart file as **/home/student/engineer.cfg**
- The resultant Kickstart file should look something like the following:

```
#platform=x86, AMD64, or Intel EM64T
#version=DEVEL
# Firewall configuration
firewall --disabled
# Install OS instead of upgrade
install
# Use network installation
url --url="http://instructor.example.com/pub/rhel6/dvd"
# Root password
rootpw --iscrypted $1$Fjoetb6b$AmMP6QWUV/Eep5XY1nH140
# Network information
```

```

network --bootproto=dhcp --device=eth0 --onboot=on
# System authorization information
auth --useshadow --passalgo=md5
# Use graphical install
graphical
firstboot --disable
# System keyboard
keyboard us
# System language
lang en_US
# SELinux configuration
selinux --enforcing
# Installation logging level
logging --level=info

# System timezone
timezone America/New_York
# System bootloader configuration
bootloader --location=mbr
# Partition clearing information
clearpart --all --initlabel
# Disk partitioning information
part /boot --fstype="ext4" --size=100
part swap --fstype="swap" --size=1
part / --fstype="ext4" --grow --size=1

%post
echo ENGINEERING WORKSTATION >> /etc/issue
%end

%packages
@base
%end

```

Practice Exercise

Make the Kickstart File Available to Installers via HTTP

1. Login to desktopX as **student**.
2. Become **root** and deploy a web server.



Note

Recall that you need to Install, Start, Enable and Test

```

[root@desktopX]# yum install -y httpd
[root@desktopX]# service httpd start
[root@desktopX]# chkconfig httpd on
[root@desktopX]# elinks -dump http://desktopX

```

3. Copy **/home/student/engineer.cfg** to **/var/www/html/**

```
[root@desktopX ~]# cp /home/student/engineer.cfg /var/www/html/
```

4. Access <http://desktopX/engineer.cfg> from a web browser to test availability.

```
[root@desktopX ~]# elinks -source http://desktopX/engineer.cfg
```



Practice Exercise

Initiating a Kickstart Installation

Before you begin...

Gracefully shutdown your **serverX (vserver)** virtual machine to reclaim system resources.

Gracefully shutdown your **wonderboy** virtual machine. Delete the virtual machine and delete the disk used for this virtual machine.

Use the Kickstart file created earlier (**engineer.cfg**) to deploy a new virtual machine.

1. On desktopX, create a new virtual machine using **virt-manager**. Choose **Network Boot (PXE)** as the installation method. Configure the virtual machine using the defaults if not specified below:
 - Name: engineer
 - Create a disk image on the computer's hard drive: 4 GB
2. Devise and enter an appropriate Kickstart invocation line for the **engineer.cfg** file, then start the installation.

Once the installation screen is presented from PXE, press the **Tab** key and enter the Kickstart information as below:

```
> vmlinuz initrd=initrd.img ks=http://desktopX/engineer.cfg
```

If the Kickstart file has errors, the installation may abort. If that is the case, edit the **/var/www/html/engineer.cfg** file on desktopX and fix the issue. Destroy the **engineer** VM and its disk and restart the installation.



Practice Case Study

Modify a Kickstart File without system-config-kickstart



Note

For time's sake you will not perform an installation using this Kickstart file.

As **root** on desktopX, create a copy of **/root/anaconda-ks.cfg** called **/home/student/projman.cfg**. Using only a text editor, modify that file so it meets the following criteria:

1. The installation must be fully automated, and exactly like the current desktopX installation (including partitioning), except...
 - The **Backup Server** package group will be installed
 - The **mtx** package, which is not installed with the **Backup Server** group by default, will be installed
 - None of the existing scripting from **%pre** and **%post** will be used
 - An **/etc/issue** file will be created in **%post**, which reads:

PROJECT MANAGEMENT

2. **ksvalidator** must be able to validate the file

When complete, copy the file to **/var/www/html/**

1. [root@desktopX]# cp /root/anaconda-ks.cfg /home/student/projman.cfg
 [root@desktopX]# chown student:student /home/student/projman.cfg
 [root@desktopX]# chmod 644 /home/student/projman.cfg
2. As student, edit the **/home/student/projman.cfg** file and include the requirements above. Make sure you uncomment the **clearpart** and other partitioning lines. It should read as follows:

```
# Kickstart file automatically generated by anaconda.

#version=RHEL6
install
url --url=ftp://instructor.example.com/pub/rhel6/dvd
lang en_US.UTF-8
keyboard us
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$YZ07ZHEP$M0u01Ut96QBgertJRauEZ/
# Reboot after installation
reboot
firewall --disabled
authconfig --useshadow --enablemd5
selinux --enforcing
timezone --utc America/New_York
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb quiet"
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --all --drives=sda

part /boot --fstype=ext4 --size=100
part pv.LtUgsR-2QYc-f90V-xG2S-PcGm-sfYq-eHJk2i --size=28000
part swap --size=512

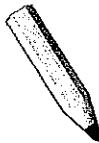
volgroup vol0 --pesize=32768 pv.LtUgsR-2QYc-f90V-xG2S-PcGm-sfYq-eHJk2i
logvol /home --fstype=ext4 --name=home --vgname=vol0 --size=500
```

```
logvol / --fstype=ext4 --name=root --vgname=vol0 --size=8192
repo --name="Red Hat Enterprise Linux" --baseurl=ftp://instructor.example.com/pub/
rhel6/dvd/ --cost=100

%packages
@Backup Server
@Base
@Console internet tools
@Core
@Desktop
@Desktop Platform
@Development Tools
@General Purpose Desktop
@Graphical Administration Tools
@Internet Browser
@Network file system client
@Printing client
@X Window System
ftp
lftp
libvirt
libvirt-client
logwatch
mtx
mutt
NSS-PAM-ldapd
ntp
policycoreutils-python
qemu-kvm
tigervnc
virt-manager
virt-viewer
%end

%post
echo PROJECT MANAGEMENT >> /etc/issue
%end
```

3. [root@desktopX]# **ksvalidator /home/student/projman.cfg**
4. [root@desktopX]# **cp /home/student/projman.cfg /var/www/html**



Test

Criterion Test

Performance Checklist

Kickstart a Virtual Machine

Before you begin...

Shutdown the **engineer** virtual machine you created earlier and delete the virtual machine and its disk.

In the Virtual Machine Manager GUI, right-click on **engineer** and choose Shutdown → Force Off. Then right-click on **engineer** again and choose Delete. Check the Delete associated storage files and click Delete.

The same can be accomplished from the command-line:

```
[root@desktopX]# virsh destroy engineer
[root@desktopX]# virsh undefine engineer
[root@desktopX]# rm -f /var/lib/libvirt/images/engineer.img
```

- Boot serverX if it is not running. Copy the **/root/anaconda-ks.cfg** file from serverX to desktopX and call it **/home/student/test.cfg**. Shutdown serverX after you have copied the file to reclaim system resources for the rest of the lab.

```
[root@serverX]# scp /root/anaconda-ks.cfg desktopX:/home/student/test.cfg
[root@serverX]# poweroff
```

- Modify **test.cfg** according to the following criteria:
 - Add **clearpart --all** and **zerombr**, and partition storage according to the following:
 - **/boot** (ext4) 200 MB
 - **swap** 512 MB
 - **/** (ext4) 3 GB
 - Add the **gimp** package
 - Create a **/root/install-date** file with the date and time.

Add the following to the Kickstart file

```
clearpart --all
zerombr
part /boot --fstype=ext4 --size=200
part swap --size=512
part / --fstype=ext4 --size=3072
...
[after %packages]
gimp
...
.

%post
date > /root/install-date
...
```

- Copy **test.cfg** to **/var/www/html/** on desktopX. Make sure the file is readable by Apache. Start the **httpd** daemon if it is not already running.

```
[root@desktopX ~]# cp /home/student/test.cfg /var/www/html/
[root@desktopX ~]# chmod 644 /var/www/html/test.cfg
[root@desktopX ~]# service httpd restart
```

- Create a logical volume in the volume group **vol0** named **test** large enough to serve as the disk for your virtual machine.

```
[root@desktopX ~]# lvcreate -n test -L 4G vol0
```

- Start a virtual machine installation using your **test.cfg** Kickstart file. Name the virtual machine **test**. Use PXE as the installation method, and allocate 768 MB of RAM and 1 CPU to the virtual machine. Use the logical volume you created in the previous step as the storage for your virtual machine.

In Virtual Machine Manager, click on New. Enter **test** as the name, and select the **Network Boot (PXE)** radio button. Click Forward.

Leave the **OS Type** and **Version** as Generic and click Forward.

Change **Memory (RAM)** to **768** MB and click Forward.

Select the **Select managed or other existing storage** radio button and enter (or browse to) **/dev/vol0/test**. Click Forward.

Verify the information is correct and click Finish to begin the installation.

When the installation menu appears, press the **Tab** key and add **ks=http://desktopX/test.cfg**, then press **Enter**. The installation should proceed and reboot when complete.

- Reboot your virtual machine when it is finished installing and confirm that it installed correctly.