Lesson #1 Introduction to Linux distributions, virtualization of a system, and installation of a Linux OS

Advanced Linux Administration

Lecturer - Karel Šrot, Aleš Zelinka

WELCOME :)

Course details

10 min

Who we are?

Red Hatters working on Red Hat Enterprise Linux.

10+ years tinkering with Linux professionally

- Robin Hack
- Karel Šrot
- Aleš Zelinka

Structure of this course

- Introduction and theory (~40 minutes)
  - Create technical background required to solve labs during workshop
- Workshop (~50 minutes)
  - Individual labs to exercise presented technology
- Course is constantly evolving and YOUR valuable FEEDBACK can help us keep improving it:
  - WIS-FIT Forum - mostly for students to discuss and openly share information
  - Email lecturers - azelinka@, ksrot@, rhack@redhat.com

Successful completion of the course

- Maximum number of points you can earn for this course is 100 points
- During the year you can earn 20 points in mid-semester assignment
- At the end of this semester there will be a practical exam worth 80 points
- To successfully pass the course and get a credit, you need to get at least 50 points in total.
  - The practical exam has a form of having to install a system according to some predefined requirements in some given time limit.
- Bonus tasks during workshop labs worth up-to 5 points, which you can work on after successful completion of non-bonus labs.

Learning Objective

The course covers advanced topics of administration of Linux operating systems as well as services typically running on such systems. The course puts emphasis on *practical training of administration skills* and *problem solving*.

Lessons

1. Introduction to Linux distributions, virtualization of a system, and installation of a Linux OS
2. Managing and troubleshooting system boot and services
3. Software package management on Linux systems.
4. Creation and basic management of storage for OS, application, and user data
5. Cryptography in GNU/Linux and secured communication.
6. Disk encryption and data backup.

---- mid-semester assignment worth 20 points ----

7. Increasing system security through SELinux.
8. Network management, security practices, and troubleshooting.
9. Creation and management of Linux Containers.
10. Configuration management and monitoring of systems.
11. Advanced systemd topics.
12. Designing solutions for complex Linux usage scenarios.
13. Practical exam (80 points)

Note: Order of lessons might change during a semester.

Workshop instructions

- Your active participation is important
- Each lab provides an estimate how much time should you spend on it if everything goes well.
- We encourage you to help each other or rise your hand to get help from lecturers.
- If you couldn’t finish all workshop labs during this class, you should complete all the labs on your own later because learned skills will be used in following lectures or during a final practical exam.
- Recommendation: Focus on a specific lab and leave any exploration or deep dive desires for a later self-study.

Introduction

40 min

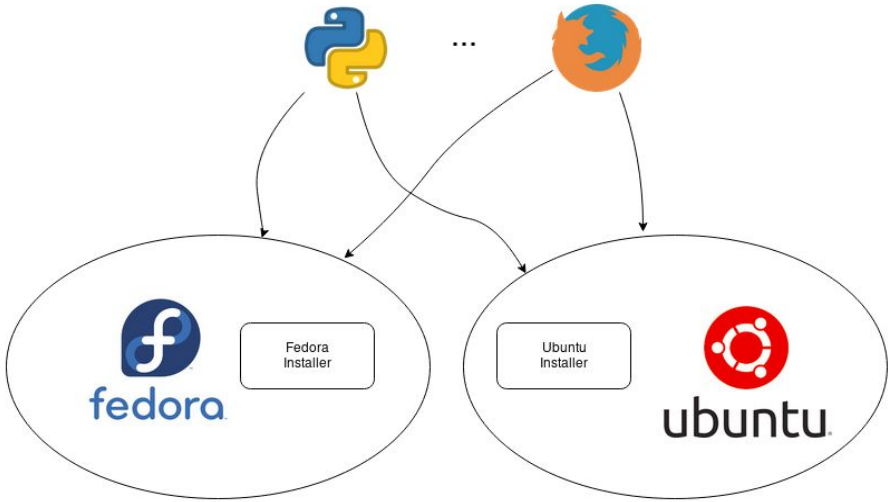
How to pick a linux distro for a particular use-case

Q: What is a distribution?

How to pick a linux distro for a particular use-case

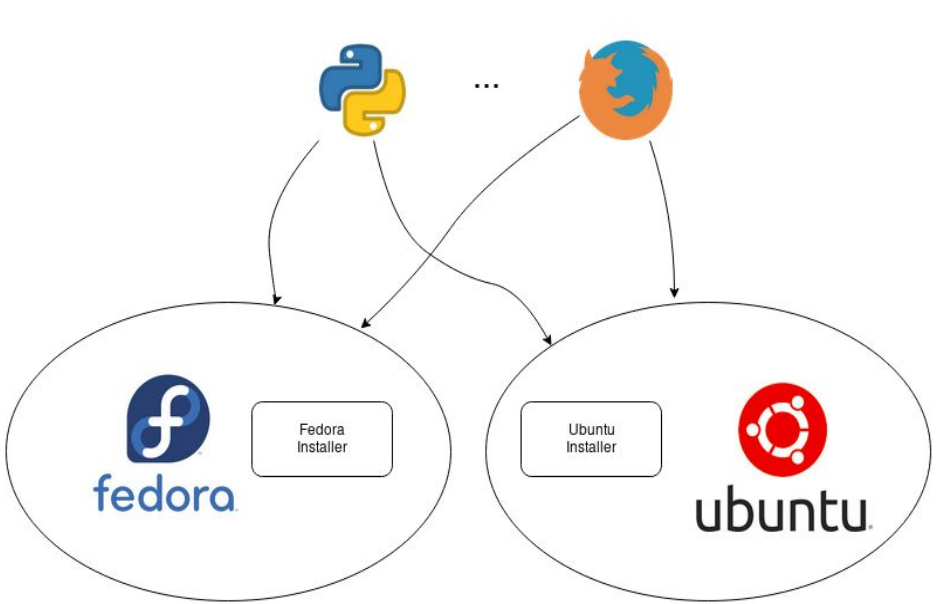
Q: What is a distribution?

A: A collection of (mostly) OpenSource/Free software which, put together, works as an operating system.



How distributions differ?

- System Installer
  - Anaconda
  - Debian Installer
  - ...



How distributions differ?

Software Packaging and Management - how software is packaged and distributed

- `$ dnf search firefox`
- `$ aptitude search firefox`

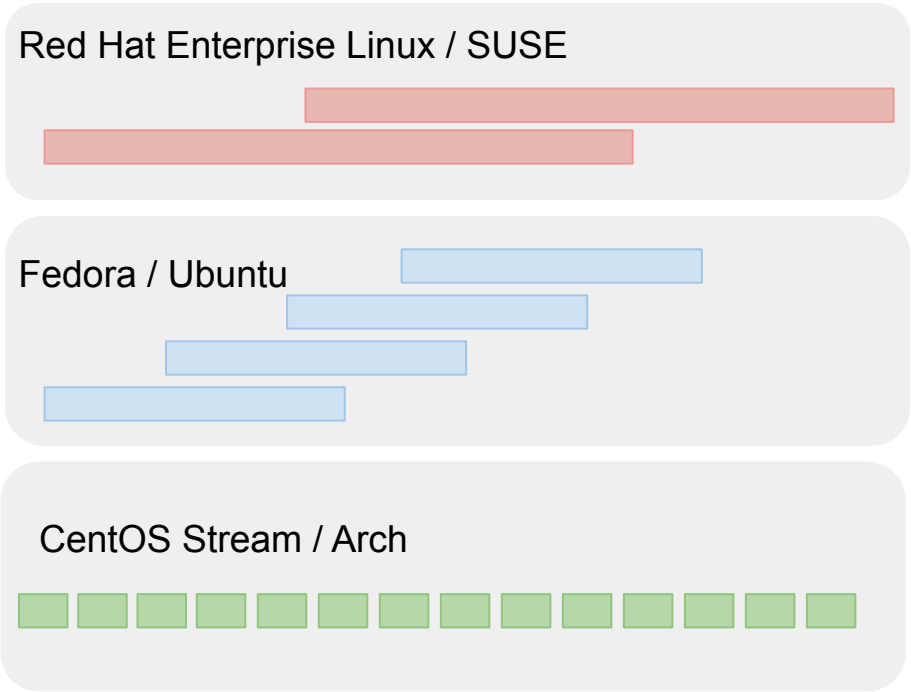
Lesson #3 “Software Package management” goes into more details.

How distributions differ?

Support length - for how long do you get support?

Support level - what do you get?

- Security fixes
- Bug fixes
- New features / software
- New HW support



Distributions: Red Hat family

- Fedora
  - end-user oriented; test bed for new technologies, 6 month release
- RHEL
  - for servers: stable, mature, +-10 year support, guaranteed security updates, paid (free for some use-cases)
- CentOS Stream
  - Free rolling release of what will become RHEL

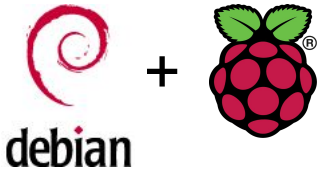
Distributions: Debian family

- Debian stable
  - 5 years, stable, security updates, often for servers
- Debian testing
  - what will become stable once it matures
  - end-user oriented; comparable with Fedora
- Ubuntu / Mint
  - end-user oriented, 9 months updates
- Ubuntu LTS
  - 10 years support (free for some use-cases), targeted at enterprise

Distributions: other full featured distros

- SUSE
- Arch / Manjaro
- Slackware
  - the oldest distribution that is still maintained
- Linux From Scratch
  - as much a distro as virus is a living organism

Distributions: Specialized distros



- device-specialized
  - Raspberry Pi OS ~ Debian clone for Raspberry Pi minicomputer
- Purpose-specialized
  - Kali ~ security and forensics
  - SystemRescueCD ~ rescue and maintenance

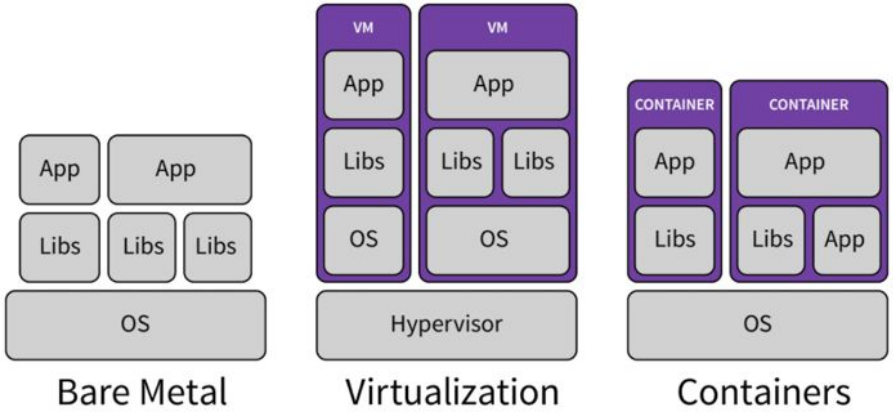
Q: Who has linux installed on laptop?

Q: Which distro?

Introduction to virtualization

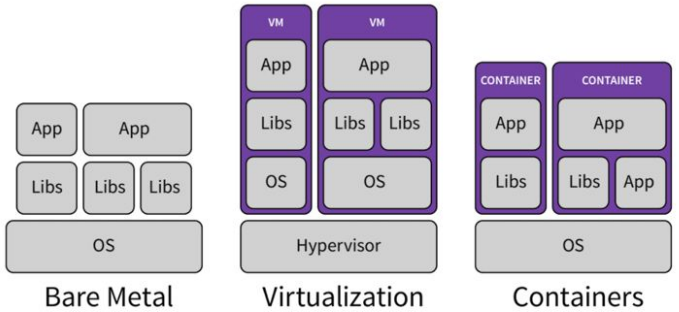
Three paradigms of Operating system deployment

- Physical computer / Bare metal
- Virtual machine
- Container



Physical computer / Bare metal

- HW selected to fit the Apps
- As an owner of a physical server, you control everything: the HW, OS (kernel and userspace) and apps
- Apps not isolated from each other
  - Share kernel and libs

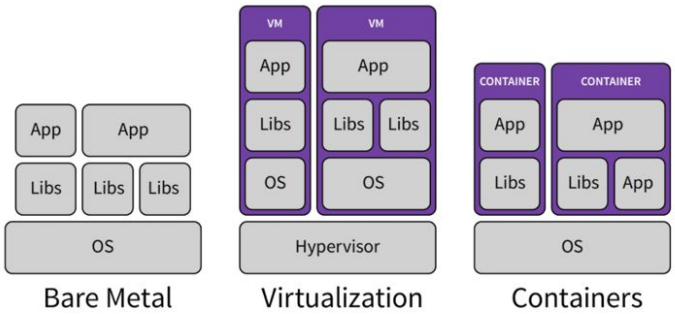


Virtualization

Hypervisor layer pretends it is HW to multiple OSes

- HW can be configured by software
  - Demo: virt-manager HW details
- Virtualized HW can be less or more (overcommit) than the physical one
  - Demo: CPU & memory overcommit

Q: How does overcommit work?



Virtualization

Hypervisor layer pretends it is HW to multiple OSes

- HW can be configured by software
  - Demo: virt-manager HW details
- Virtualized HW can be less or more (overcommit) that the physical one
  - Demo: CPU & memory overcommit

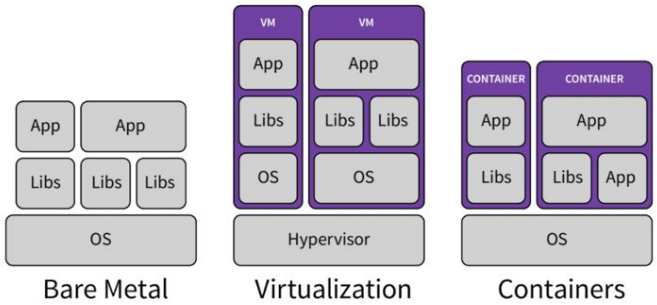
Q: How does overcommit work?

A: The same way you can have more processes than CPU cores and allocate more memory that you physically have (swap)

Virtualization

Primarily used for

- Sharing (saving) HW resources
- Isolation
  - apps from one VM don’t see Apps from other VMs
- HW independence

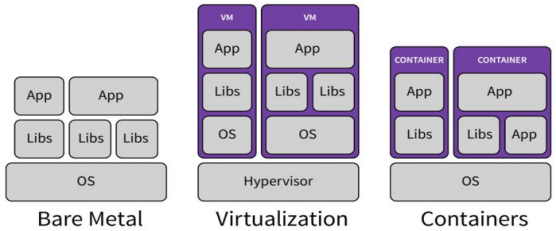


Containers

Containers are multiple userlands on top of common kernel

Similar properties as VMs (isolation, sharing,...) but because we don’t boot full OSes, containers are:

- Very lightweight
- Very fast to start

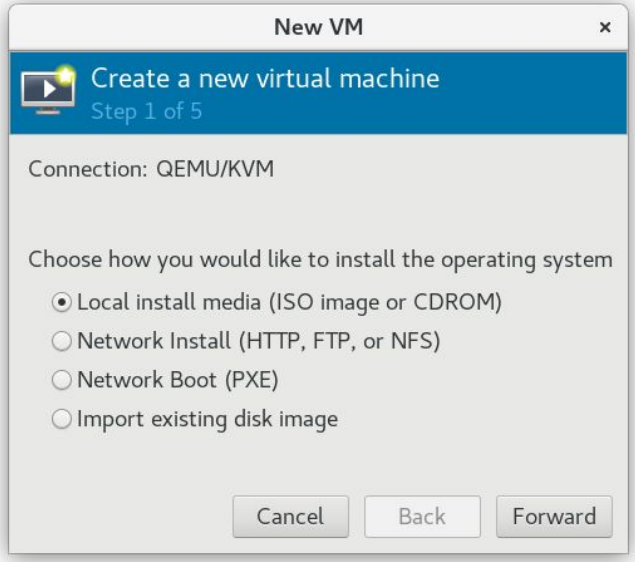


Demo: same kernel, different userland

```
$ uname -rv ; ps a
```

OS installation methods

- **Local**
  - Download full installation media, boot from it
- Network
  - Download just the installer media, boot from it, download the rest during the installation
- PXE
  - Download installation media/installer and make it available in your local network (dhcp, tftp), boot from network



OS installation methods

Automated/unattended installations

- Fedora/RHEL/Centos: kickstarts
- Debian FAI / preseed files for debconf

Kickstart example

```
#version=DEVEL
ignoredisk --only-use=vda
autopart --type=lvm
# Partition clearing information
clearpart --none --initlabel
# Use graphical install
graphical
# Use network installation
url --url="https://mirror.karneval.cz/pub/linux/fedora/linux/releases/30/
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
# System language
lang en_US.UTF-8

# Network information
network  --hostname=localhost.localdomain
# Root password
rootpw --iscrypted $6$pSYwjwacnj31VxzV$JZxBzx8Dh/n9pEKriJxfXY4Gmevd4QNTge
# Run the Setup Agent on first boot
firstboot --enable
# System services
services --enabled="chronyd"
```

Kickstart: how to create?

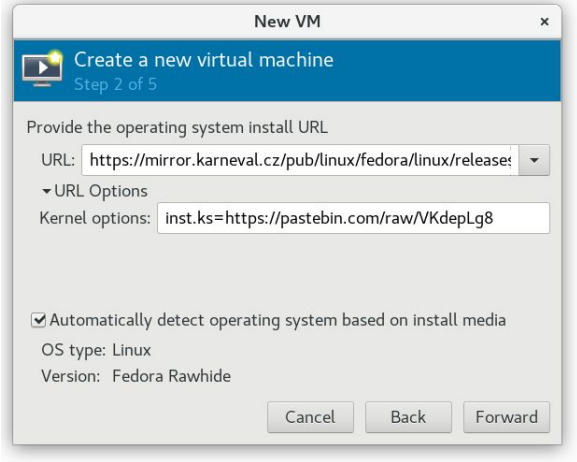
- Install once interactively, you'll find a kickstart for your system generated at /root/anaconda-ks.cfg
  - Demo
- Tweak the kickstart using Kickstart Syntax Reference
- Check your kickstart
  - `$ ksvalidator`

Read more in Fedora Kickstart Guide

Kickstart: how to use?

Make kickstart available from network (as that's usually the only available source during fresh installs)

- https://rentry.co or Pastebin.com for testing purposes (use URL pointing to a RAW data)
- Internal web server for kickstarts with private/sensitive data



Start network installation and provide additional kernel configuration with your kiskart URL

- `inst.ks=http://url.of.your.kickstart`

Lesson #2 will go into more detail for what kernel options are

Anaconda advanced configuration - virtual consoles

Press Ctrl+Alt+[F1..F6] to access virtual consoles

- F6 - the Anaconda GUI (default)
- F1 - tmux showing various logs
- F2,F3,F5 - interactive shells
  - Can do advanced setup e.g. fdisk, mdadm, LVM,...
- F4 - NetworkManager log

You can press the keys virtually via the “Send Key” menu in virt-manager.

Workshop

40 min

Lab 1 - Create Fedora38 Server Virtual Machine [20 min] (1/2)

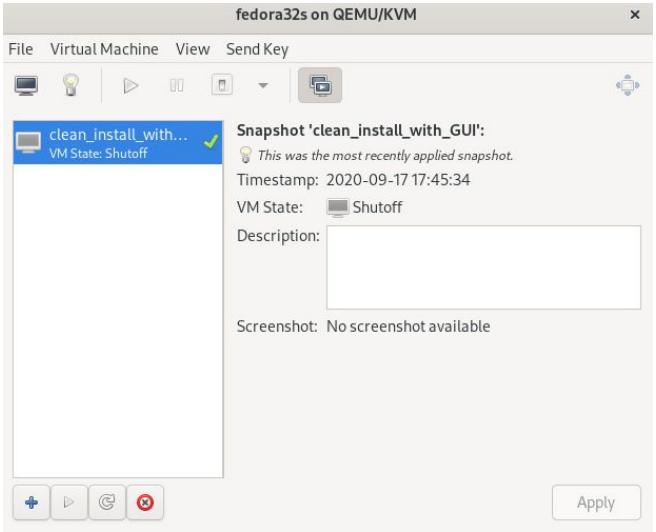
- Use VM management app (virt-manager preferred) to create new VM
  - Use your downloaded Fedora38 Server ISO as the source
  - Memory: 2GB, CPU: 1, Storage: 30GB
  - Keep the rest as default, start the installation
- During Installation
  - Pick English as the default language
  - Confirm default disk partitioning
  - Set root password
  - Finish installation
- Recommended SW for virtualization
  - Linux: virt-manager (e.g. dnf search virt-manager)
  - Windows: vmWare Player
  - MacOSX: Virtualbox

Lab 1 - Create Fedora38 Server Virtual Machine [20 min] (2/2)

Known issues with virtualization solutions

- Disabled virtualization in BIOS (lenovo)
- Can’t run 64bit OS in Virtualbox on Windows [1] [2]

Create a snapshot of the freshly installed VM.



Lab 2 - Verify VM is functional [5 min]

Verify that your new VM is fully functional by installing a package as root. In terminal do:

- Login as root
- Install “midnight commander” filemanager
  - `$ dnf install mc`
- Optional: Install graphical desktop
  - $ dnf group install GNOME
  - $ systemctl set-default graphical.target

Lab 3 - Install Fedora38 using kickstart [15 min]

Install Fedora 38 Server using kickstart method.

- Create Fedora38 kickstart file or reuse existing one
- Modify the kickstart
  - Use `‘url’` and `‘reboot’` directives
    - Use any url from this mirrorlist
  - install `‘mc’` package (see %packages directive)
  - Remove `‘cdrom’` directive (conflicts with url)
- Upload kickstart to pastebin.com (or similar service)
- In virt-manager select network installation
  - For URL use any from the mirrorlist again
  - Use URL options / kernel options `inst.ks=http://your.pastebin.com/raw/xyz`