BaLu Wiki Documentation

Release 0.23

Balle & Lilu

Aug 08, 2017

Contents

CHAPTER 1

Arch Linux

Install yaourt

- Edit /etc/pacman.conf

```
[archlinuxfr]
Server = http://repo.archlinux.fr/$arch
```

- Install yaourt

```
pacman -Sy yaourt
```

- Now you can install packages from aur.archlinux.org with

```
yaourt -S <pkg>
```

CHAPTER 2

Cgroups

Overview

- Cgroups group processes so that you can define ressource limits and get stats for it
- Processes are called tasks
- Every process can only be in one cgroup
- A cgroup can inherit the properties of another cgroup

```
mkdir /cgroup/
mount -t cgroup -o memory nodev /cgroup/
```

Installation

```
yum install libcgroup
```

Create a new cgroup

- Temporarily

```
mkdir /cgroup/<groupname>
```

- Or

```
cgcreate -a <user> -g memory,cpu:<groupname>
```

- Permanent by editing `/etc/cgconfig.conf`

```
group <name> {
  [<permissions>]
  <controller> {
      <param name> = <param value>;
   }
}
```

- e.g.

```
mount {
      cpuset  = /cgroup/cpuset;
      cpu     = /cgroup/cpu;
      cpuacct = /cgroup/cpuacct;
      memory  = /cgroup/memory;
      devices = /cgroup/devices;
      freezer = /cgroup/freezer;
      net_cls = /cgroup/net_cls;
      blkio   = /cgroup/blkio;
}

group students {
  blkio {
    blkio.throttle.read_bps_device = "1000";
    blkio.throttle.write_bps_device = "1000";
  }
}
```

- Dont forget to restart the cgconfig service in order to load the changes!

```
service cgconfig restart
```

Map user and processes to a cgroup

- Edit `/etc/cgrules.conf`

```
<user> <subsystems> <cgroup>
<user>:<command> <subsystems> <cgroup>
```

- names with a prepending @ are groups

Manually starting a process in a cgroup

```
cgexec -g <subsystems>:<cgroup> <command> <arguments>
```

Define limits

- For memory

```
memory.limit_in_bytes = 1000000
```

- Cpu time (default 1024 is 100% so 100 is ~10%)

```
cpu.shares = 100
```

- CPU pinning

```
cpuset.cpus = 0-5,14,15
```

- Storage time (100% is value of 1024)

```
blkio.weight = 512
```

CHAPTER 3

Debian

Install with sysvinit as init system

- On boot screen of install medium type tab and append

```
preseed/late_command="in-target apt-get install -y sysvinit-core"
```

How to build a deb package

- Unzip archive
- cd source
- dh_make -e me@mail.net -f ../tar-file
- choose single binary
- Edit debian/control file
- define dependency e.g.
- libssl0.9.8 (>= 0.9.8~)
- Enter description
- Edit debian/rules

```
override_dh_auto_configure:
    dh_auto_configure -- --prefix=/opt/myprogram
```

- Move debian/init.d.ex to debian/myprogram.init and maybe edit it
- dpkg-buildpackage -rfakeroot -b

Setting up a chroot environment

```
debootstrap unstable /data/debian-tree
```

CHAPTER 4

Debugging

Create a core dump file

```
ulimit -c unlimited
```

Read core dump file

```
gdb /path/to/application /path/to/corefile
```

CHAPTER 5

Fedora

Restart network

```
systemctl restart NetworkManager.service
```

Allow SELinux exception

- Click on selinux alert in bottom right corner
- Click on Troubleshoot
- Paste solution as root

Disable boot logo

```
plymouth-set-default-theme details -R
```

Gnome3 settings

- Install gnome-tweak-tool
- Install gconf-editor

Change login background

- Edit /usr/share/backgrounds/beefy-miracle/default/beefy-miracle.xml

Install flash plugin

```
rpm -ivh http://linuxdownload.adobe.com/adobe-release/adobe-release-i386-1.0-1.noarch.
↪rpm
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-adobe-linux
yum check-update
yum install flash-plugin nspluginwrapper alsa-plugins-pulseaudio libcurl
```

Upgrade system

- http://fedoraproject.org/wiki/Upgrading_Fedora_using_yum

CHAPTER 6

Filesystem

Show used inodes

```
df -i
```

Dont reserve 5% space for system

```
mke2fs -m 0
```

Find out filesystem of unmounted device

```
parted -l /dev/sda
```

SSD

- Add the following options to fstab

```
noatime,discard
```

CHAPTER 7

Gnome3

Configure startup applications

- You can eithe use

```
gnome-session-properties
```

- or

```
gnome-tweak-tool
```

Install a new theme

- Goto www.gnome-look.org
- Download a theme
- Unzip it to ~/.themes
- Install it with gnome-tweak-tool

Cool themes

- http://gnome-look.org/content/show.php/Ambiance+Dark+Blue?content=169553
- http://gnome-look.org/content/show.php/E17gtk?content=163472
- http://gnome-look.org/content/show.php/Cenodark-red?content=165815
- http://gnome-look.org/content/show.php/Just-Dark?content=168025q

Automatically place windows

- Enable Auto move windows extension
- Add new placement rule

Non blank screensaver

- killall gnome-screensaver
- yum install xscreensaver
- xscreensaver-demo

Changing keyboard shortcuts

- Applications -> System Tools -> System Settings -> Keyboard

Default shortcuts

- Alt+F1 - Switch between overview and desktop view
- Alt+F2 - Launch command
- Windows key - Goto activity view (enter first keys of what to execute / search)

Reload Gnome config

- Alt+F2 r

Install extensions

- Goto extensions.gnome.org
- Click on an extensions
- Use the on/off button to install/deinstall extension

Pidgin integration

- Install https://extensions.gnome.org/extension/170/pidgin-peristent-notification/
- Or install https://extensions.gnome.org/extension/258/notifications-alert-on-user-menu/

System monitor in top panel

- Install https://extensions.gnome.org/extension/120/system-monitor/

Declaring the primary monitor

- Goto settings -> monitor
- Use drag and drop to move the monitors around, the first is the primary

Configure desktop to be on all monitors

- gnome-tweak-tool -> desktops

Remove window decorations

- Assuming you use Adwaita as theme
- Edit /usr/share/themes/Adwaita/metacity-1/metacity-theme-1.xml
- Search for frame_geometry name=”normal”
- Add has_title=”false”
- Reload Gnome config

Alternative menu

- To access the alternative menu hold the Alt key
- This will for example allow you to hibernate instead of shutdown in the user menu

Switching back to old GNOME look and feel

- Either set “Use Fallback Mode” in System Settings -> Details -> Graphics
- Or install Mate http://mate-desktop.org/

CHAPTER 8

Grub2

Overview

- Setting can be found in `/etc/default/grub`
- Menu entries are located in `/etc/grub.d/` as shell scripts (ordered by their number)
- Do not directly edit /boot/grub/grub.cfg that the main config but its created from the above
- To generate a basic config run

```
grub-mkconfig -o /boot/grub/grub.cfg
```

- To install grub into MBR

```
grub-install /dev/sda
```

Global Kernel parameter

- Edit `/etc/default/grub` parameter `GRUB_CMDLINE_LINUX`

Create a new Linux entry

- Create a new script in `/etc/grub.d` e.g. 11_mylinux or edit `40_custom`

```
#!/bin/sh
set -e

menuentry "My Linux" {
  set root=(hd0,1)
  linux /vmlinuz (add other options here as required)
```

```
  initrd /initrd.img (if the other kernel uses/needs one)
}
```

- Dont forget to make it executable!

Boot encrypted root with ramdisk

```
menuentry "My Linux" {
  set root=(hd0,1)
  linux /vmlinuz root=/dev/mapper/root cryptdevice=/dev/sda4:root
  initrd /initrd.img
}
```

Boot encrypted root without ramdisk (untested)

```
menuentry "My Linux" {
  insmod gzio
  insmod part_gpt
  insmod cryptodisk
  insmod luks
  insmod gcry_twofish
  insmod gcry_sha256

  cryptomount -u <uuid>

  insmod lvm
  insmod ext2

  set root='lvm/vg-root'
  linux /vmlinuz root=/dev/mapper/vg-root
}
```

Boot Windows

- Create a new script in `/etc/grub.d` e.g. 99_windows

```
#!/bin/sh

menuentry "Windows XP" {
  set root="(hd0,3)"
  chainloader +1
}
```

Boot only signed kernel and ramdisk

- Generate gpg keypair
- Sign kernel

```
gpg --detach-sign vmlinuz
```

- Edit Grub config

```
trust boot.key
set check_signatures=enforce
```

Booting a rescue cd image

```
menuentry "SYSRESCUECD" {
  set iso=/systemrescuecd-x86-3.8.1.iso
  loopback loop ${iso}
  linux  (loop)/isolinux/rescue64 nomodeset vga=791 docache setkmap=fr isoloop=${iso}
  initrd (loop)/isolinux/initram.igz
}
```

Convert grub1 menu.lst to grub2 config

```
grub-menulst2cfg /boot/grub/menu.lst /boot/grub/grub.cfg
```

CHAPTER 9

Inotify

- React on file change events
- Install incron
- incrontab -e

```
<directory> <file change mask> <command or action>  options
/var/www/html IN_CREATE /root/scripts/backup.sh
/sales IN_DELETE /root/scripts/sync.sh
/var/named/chroot/var/master IN_CREATE,IN_ATTRIB,IN_MODIFY /sbin/rndc reload
/tmp IN_ALL_EVENTS logger "file $@ changed"
```

Recursive inotify

- https://github.com/splitbrain/Watcher
- http://www.splitbrain.org/blog/2011-01/07-watcher_a_recursive_incron_alternative

```
[DEFAULT]
logfile=/var/log/watcher.log
pidfile=/var/run/watcher.pid

[data]
watch=/data
events=create,delete,modify
recursive=true
autoadd=true
command=rsync -a --delete $filename /media/backup
```

CHAPTER 10

IPMI

Overview

- Install freeipmi (http://www.gnu.org/software/freeipmi/documentation.html) or ipmitool
- Read http://www.thomas-krenn.com/de/wiki/IPMI_Grundlagen
- Check that your system has IPMI

```
dmidecode | grep IPMI
cat /proc/ipmi/0/stats
ls /dev/ipmi0
```

- If you dont have ipmi0

```
modprobe ipmi_devintf
```

Configuration

```
bmc-config --checkout > ipmi.conf
```

- Edit ipmi.conf

```
cat ipmi.conf | bmc-config --commit
```

Get sensor data

- Local

```
ipmi-sensors
ipmitool sdr list
```

- Remote

```
ipmi-sensors -h $host -u $user -P
ipmitool sdr list -H $host -U $user -P
```

- More detailed and better parsable data

```
ipmimonitoring -h $host -u $user -P
```

Get chassis status

```
ipmi-chassis -h $host -u $user -P -s
```

Power machine on / off

```
ipmipower --on -h $host -u $user -P
ipmitool power reset -H $host -U $user
```

Activate chassis LED

```
ipmi-chassis -h $host -u $user -P -i 1
```

Read system event logs

- General information

```
ipmi-sel -h $host -u $user -P -i
ipmitool sel elist
```

- Real logs

```
ipmi-sel -h $host -u $user -P
```

Configure network for remote console

```
ipmitool lan set 2 ipaddr $IP
ipmitool lan set 2 netmask 255.255.255.0
ipmitool lan set 2 defgw ipaddr $GW
ipmitool lan print 2
```

Configure user

```
ipmitool user set name <userid> balle
ipmitool user set password <userid> ""
```

- admin privs

```
ipmitool channel setaccess 1 <userid> link=on ipmi=on callin=on privilege=4
ipmitool channel setaccess 2 <userid> link=on ipmi=on callin=on privilege=4
```

- user privs

```
ipmitool channel setaccess 1 <userid> link=on ipmi=on callin=on privilege=2
ipmitool channel setaccess 2 <userid> link=on ipmi=on callin=on privilege=2
```

- dont forget to enable the user

```
ipmitool user enable <userid>
```

Get serial console

Restart ipmi controller

```
ipmitool bmc reset cold
```

Check that ipmi controller is ok

```
ipmitool bmc selftest
```

CHAPTER 11

Kernel

USB shuts itself down

- Append `usbcore.autosuspend=-1` to kernel parameters

What is load?

- With a quad core system a load average of 5 means that all 4 cpus are busy for 100% and processes to fill another cpu are waiting
- http://blog.scoutapp.com/articles/2009/07/31/understanding-load-averages

```
cat /proc/loadavg
```

- field 1-3 = load average of jobs in the run queue (state R) or waiting for disk I/O (state D) averaged over 1, 5, and 15 minutes
- field 4 = number of currently runnable kernel scheduling entities (processes, threads) / number of kernel scheduling entities that currently exist on the system
- field 5 PID of the process that was most recently created on the system

Find out which driver is in use

- network card

```
ls -al /sys/class/net/eth5/device/driver/module
```

- generic

```
lspci | grep VGA
02:00.0 VGA compatible controller: Matrox Electronics Systems Ltd. MGA G200e [Pilot]␣
↪ServerEngines (SEP1) (rev 02)

find /sys | grep driver.*02:00
```

- or more easy

```
lspci -vv
```

Use old network device names

- Start kernel with the parameters `net.ifnames=0 biosdevname=0`
- Or disable automatic renaming in udev

```
ln -s /dev/null /etc/udev/rules.d/80-net-setup-link.rules
```

- Or rename devices with udev

```
cat > /etc/udev/rules.d/99-rename-to-eth0.rules << EOF
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="$(cat /sys/class/net/
↪ens33/address)", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
EOF
```

Availale parameters for kernel module

```
modinfo <module_name>
```

Show current kernel boot parameters

```
cat /proc/cmdline
```

Hotplug CPUs

- Enable

```
echo 1 > /sys/devices/system/cpu/cpu<No>/online
```

- Disable

```
echo 1 > /sys/devices/system/cpu/cpu<No>/online
```

Check if virtualization is enabled

```
grep vmx /proc/cpuinfo
```

Check if TPM is available

```
grep smx /proc/cpuinfo
```

Hide kernel symbols

- Create /etc/sysctl.d/50-kptr-restrict.conf

```
kernel.kptr_restrict = 1
```

CHAPTER 12

LVM

Register a physical device

```
pvcreate <dev>
```

Create a volumne group

```
vgcreate <vgname> <dev>
```

Add a new device to a volumne group

```
vgextend <vgname> <dev>
```

Display all volumne groups

```
vgdisplay
```

Which device is in which volumne group

```
pvs
```

Just show free space of volumne groups

```
vgs
```

Create a logical device

```
lvcreate -L 10G -n <name> <vgname>
```

Resize a logical device

```
lvresize -L +/-1G /dev/vgname/lvname
```

- Dont forget to do a resize of the filesystem
- Shrinking usually requires umounting

Create a snapshot from lvname

```
lvcreate -L 1G -n snap --snapshot /dev/vgname/lvname
```

Troubleshooting

- Rescan

```
pvscan
lvscan
```

- Check that volumes / volume groups are active (Attr a)

```
lvs / pvs
```

- Reactivate all

```
vgchange -ay
```

CHAPTER 13

Memory Management

What is the Linux Page Cache

- http://www.linux-tutorial.info/modules.php?name=MContent&pageid=310

Page cache and swap config

- http://www.westnet.com/~gsmith/content/linux-pdflush.htm

Large page support

- http://linuxgazette.net/155/krishnakumar.html

choosing an i/o scheduler

- http://www.redhat.com/magazine/008jun05/features/schedulers/

Misc

- openbook: understanding the linux virtual memory manager

http://www.kernel.org/doc/gorman/html/understand/

- die anzeige bei cached steht für den disk cache

Page cache

- To free pagecache:

```
echo 1 > /proc/sys/vm/drop_caches
```

- To free dentries and inodes:

```
echo 2 > /proc/sys/vm/drop_caches
```

- To free pagecache, dentries and inodes:

```
echo 3 > /proc/sys/vm/drop_caches
```

Configs

- /proc/sys/vm/dirty_writeback_centisecs 500 - 5 sekunden wann der disk cache aufgeräumt wird
- /proc/sys/vm/dirty_expire_centiseconds 3000 - 30 sekunden wann die page auf dirty gesetzt wird
- /proc/sys/vm/dirty_ratio - maximal % für disk cache
- dirty_background_ratio - % von ram ab wann pdflush aufräumt
- vfs_cache_pressure - wenn 0 gibt nach möglichkeit keinen vfs cache frei wenn 100 oder mehr gib lieber vfs cache frei

CHAPTER 14

OpenPandora

Emulator handbook

- http://dl.openhandhelds.org/pandora//uploads/Home/Pandora%20-%20Emulators/yoshis_pandora_emulator_fact_sheets_v07.pdf

Firmware update

- Install and run http://repo.openpandora.org/?page=detail&app=szupdater1.openpandora.org

Firmware update (old way)

- http://www.openpandora.org/index.php?option=com_content&view=article&id=199&Itemid=40&lang=en
- hold right shoulder button
- boot from sd-card

Start SSH server

```
/etc/init.d/dropbear start
```

Deactivate boot splash

- Edit /boot/autoboot.txt and append `psplash=false` at the end of the `setenv bootargs` parameter
- Or hit `Alt + D-pad right` to disable it temporarily

Turn wifi off

- pandora-ctl stop wifi

Handling PND Files

- Run in console with pnd_run
- Mount PND File
- /usr/pandora/scripts/pnd_run.sh -m -p <pnd_file>

Installing software

- opkg install <ipk-file>
- Or use Milky (pacman for pandora)

Setting up cross-compile environment

- Download http://git.openpandora.org/cgi-bin/gitweb.cgi?p=pandora-misc.git;a=blob_plain;f=sdk_installer/openpandora_toolchain.sh;hb=HEAD
- dos2unix openpandora_toolchain.sh
- chmod a+x openpandora_toolchain.sh
- Execute openpandora_toolchain.sh

Cross compile some source

- Go to the source tree and Execute

```
pandora-dev/sdk_utils/pandora_configure.sh --prefix=/data/muh/
make
make install
pandora-dev/sdk_utils/pnd_make.sh -p zsh.pnd -d /data/muh
```

- pnd_make.sh must run as root
- Maybe you have to add `-I$PNDSDK/usr/include` to CFLAGS in Makefile or pandora_configure.sh

Building PND file

```
pandora-dev/sdk_utils/pnd_make.sh -p some-new-app.pnd -d /dir-to-compress
```

- Maybe you want to edit PXML.xml to specify exec script

```
<exec command="scripts/install.sh"/>
```

- To rebuild a mounted pnd on pandora use

```
mksquashfs . /tmp/new.pnd ; cat PXML.xml icon.png >> /tmp/new.pnd
```

Setting up complete development environment

- http://blogs.distant-earth.com/wp/?p=90

CHAPTER 15

Redhat

Building a RPM package

- Install rpmbuild
- Setup environment

```
mkdir ~/rpmbuild
cd ~/rpmbuild
mkdir BUILD  BUILDROOT  RPMS  SOURCES  SPECS  SRPMS
```

- Put the source archive in the SOURCES directory (e.g. myprogram-0.1.tgz)
- Make a spec file in SPECS (e.g. myprogram.spec)

```
Name:           myprogram
Version:        0.1
Release:        1
Summary:        A short description

Group:          System Environment/Libraries
License:        GPL
URL:            https://some.url.net
Packager:       Your name <you@somewhere.net>

Requires:       some_other_package>=1.2.3, another_package

Source:         %name-%version.tgz

%description
Some longer description of this software package

%prep
%setup
```

```
%build
./configure --prefix=%{buildroot}
make

%install
rm -rf %{buildroot}
make install
```

- Build the package

```
rpmbuild -bb SPECS/myprogram.spec
```

- rpmbuild will complain that there are files that are not packaged. Put those into the %files section of the spec file

```
%files
/path/to/file1
/path/to/file2
```

- Rebuild the package
- The RPM should now be available in the RPM subdir
- Only install dont recompile

```
rpmbuild -bi --short-circuit SPECS/myprogram.spec
```

- Dont want to package all installable files?

```
%define _unpackaged_files_terminate_build 0
```

- For more goto http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html-single/RPM_Guide/index.html

List contents of uninstalled rpm

```
rpm -qlp <rpm_file>
```

Unpack rpm without installing

```
mkdir bla
cd bla
rpm2cpio ../ethjudge-backend-libs-1.4-4.el7.noarch.rpm | cpio -idvm
```

Check checksum of files in a rpm

- Installed rpm

```
rpm -Vv <rpm_file>
```

- Uninstalled rpm

```
rpm -V -p <rpm_file>
```

Exclude a package from update

- Edit /etc/yum.conf

```
exclude=some_pkg*
```

Get all versions of a package and their repos

```
yum list <pkgname> --showduplicates
```

Downgrade a package

```
yum downgrade <pkgname>
```

Remove package with all dependencies

- Edit `/etc/yum.conf`

```
clean_requirements_on_remove=1
```

Examine installation history

```
yum history list [package]
yum history info <id>
```

Revert an update

```
yum history list
yum undo <id>
```

Service Configuration

- List all available services and their status

```
chkconfig --list
```

- Turn service on boot on or off

```
chkconfig <service> [on|off]
```

- Start or stop a service

```
service <service> [start|stop]
```

Firewall Config

- Prefered tool is `system-config-firewall`
- or lokkit

```
lokkit -p 80:tcp
lokkit -s http
```

- Script can be found under `/etc/sysconfig/iptables` but will be overwritten by the commands above

Bridged interface

- /etc/sysconfig/network-scripts/ifcfg-br0

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
DELAY=0
```

- /etc/sysconfig/network-scripts/ifcfg-eth0

```
BRIDGE=br0
```

Kickstart

- The kickstart file used to setup the system can be found in /root/anaconda-ks.cfg

Gnome-Keyring

- To reset Gnome-Keyring passwords run

```
rm ~/.gnome2/keyrings/*
```

Setting up a chroot environment

```
mkdir -p /data/redhat/var/cache/yum/x64_64/\$releaseserver
cp /etc/yum.repos.d/redhat.repo /data/redhat/var/cache/yum/x64_64/\$releaseserver
yum --disablerepo=* --enablerepo=redhat --disableplugin=* --installroot=/data/redhat␣
↪install bash
```

CHAPTER 16

Systemd

List all units and their status

- All running

```
systemctl
```

- List only services

```
systemctl list-units --type=service
```

- All available

```
systemctl list-unit-files
```

List all failed services

```
systemctl --failed
```

Start / Stop service

- All services can be found in */usr/lib/systemd/system*

```
systemctl [start|stop] sshd.service
```

Activate service on boot

```
systemctl enable sshd.service
```

Show status of a service

```
systemctl status sshd.service
```

List all available targets (runlevels)

```
systemctl list-units --type=target
```

Change default target (runlevel)

```
systemctl set-default multi-user.target
```

- or

```
rm /etc/systemd/system/default.target
ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/default.target
```

Persistent logs

- Normally journald logs to /run/log/journal this is a tmpfs and logs are deleted on reboot
- To avoid this edit `/etc/systemd/journald.conf`

```
[Journal]
Storage=persistent
```

- If that's not working try

```
mkdir /var/log/journal
chgrp systemd-journal /var/log/journal
chmod 775 /var/log/journal
```

Filtering logs

- Use the force of TAB completion!

```
journalctl <TAB>
journalctl _COMM=<TAB>
```

- Errors since last boot

```
journalctl -b -p err
```

- List all reboots

```
journalctl --list-boots
```

- Since today

```
journalctl --since today
```

- One hour ago

```
journalctl --since -1h
```

- Or timerange

```
journalctl --since=2012-10-15 --until="2011-10-16 23:59:59"
```

- For a specific file

```
journalctl /some/file
```

- Tailed

```
journalctl -f
```

- For a single pid

```
journalctl _PID=123
```

- For a single unit (service)

```
journalctl -u <servicename>
```

- For kernel messages

```
journalctl -k
```

- For network stuff

```
journalctl _COMM=network
```

- For a SELinux context

```
journalctl _SELINUX_CONTEXT=<security context>
```

- For a single user

```
journalctl _UID=<userid>
```

- Full output for last 10 messages

```
journalctl -l -o verbose -n 10
```

- Where to find the log files?

```
cd /var/log/journal
```

- How to configure max hd space for logs? Edit /etc/systemd/journald.conf

```
SystemMaxUse=100M
```

- Log rotation (/etc/systemd/journald.conf)

```
MaxRetentionSec=1day
MaxFileSec=1month
```

- How to log to syslog (edit /etc/systemd/journald.conf)

```
ForwardToSyslog=yes
```

- Export log as JSON

```
-o json
```

Remote logging

- Install systemd-journal-gateway
- On server edit `/etc/systemd/journal-remote.conf` and start service systemd-journal-remote
- On log client edit `/etc/systemd/journal-upload.conf`, to URL to `http://<ip_of_logserver>:19531` and start service systemd-journal-upload

Journald Web Gateway

- Install systemd-journal-gateway
- Start service systemd-journal-gateways
- Connect your browser to http://<ip>:19531
- To get an endless stream http://<ip>:19531/entries?follow
- To pull remote journal log an save it to a text file

```
nohup curl --silent -o some-host.log 'http://<ip>:19531/entries?follow' &
```

- Or to pull it in the original journal format

```
nohup curl --silent -H'Accept: application/vnd.fdo.journal' -o some-host.log 'http://
↪<ip>:19531/entries?follow' &
```

Rescue Mode / Debugging

- On Grub prompt try to set one of the following kernel parameter

```
systemd.unit=rescue.target      # (single user mode)
systemd.unit=emergency.target   # (only shell)
```

- Ask before starting a servce

  systemd.confirm_spawn=1

- Give me more log output

```
systemd.log_target=kmsg systemd.log_level=debug
```

- Get console output of legacy sysv init scripts

```
systemd.sysv_console=1
```

- Which units want which target?

```
systemctl show -p "Wants" multi-user.target
```

- To analyze which services was slow

```
systemd-analyze blame
```

What services do get started?

```
systemctl list-dependencies multi-user.target
```

Change runlevel

```
systemctl isolate <newtarget e.g. rescue.target or mutli-user.target>
```

Changing the default runlevel

```
ln -sf /usr/lib/systemd/system/multi-user.target /etc/systemd/system/default.target
```

An example service

```
[Unit]
Description=Just a simple test
After=syslog.target

[Service]
ExecStart=/bin/some-daemon
Type=forking
CPUShares=1500
MemoryLimit=1G
BlockIOWeight=500
```

```
[Install]
WantedBy=multi-user.target
```

- Afterwards exec

```
systemctl daemon-reload
systemctl start test.service
systemctl status test.service
```

Power management

```
systemctl suspend
systemctl hibernate
```

Use systemd as inetd

- http://0pointer.de/blog/projects/inetd.html

Chrooting

- Set up chroot environment with yum or debootstrap or whatever
- Old school with chroot()

```
[Service]
RootDirectory=/srv/chroot/foobar
```

- New age with kernel namespaces

```
systemd-nspawn -D <chroot_dir> <command>
```

- For more see http://0pointer.de/blog/projects/changing-roots

More security options

- Disable networking

```
PrivateNetwork=yes
```

- Isolate tmp dir

```
PrivateTmp=yes
```

- Read-only or inaccessible directories

```
InaccessibleDirectories=/home
ReadOnlyDirectories=/var
```

- Use capabilities (see man capabilities)

```
CapabilityBoundingSet=CAP_CHOWN CAP_KILL
```

- Use process limits

```
LimitNPROC=1
LimitFSIZE=0
```

- Limit device usage

  DeviceAllow=/dev/null rw

- Run as a specific user / group

```
User=nobody
Group=nobody
```

Only start a service if a specific device is found

```
BindToDevice=dev-sda5.device
```

I want more gettys / text consoles

```
ln -sf /usr/lib/systemd/system/getty@.service /etc/systemd/system/getty.target.wants/
↪getty@tty9.service
```

Python Coding

- http://www.freedesktop.org/software/systemd/python-systemd/
- https://pypi.python.org/pypi/pyjournalctl/0.7.0

Custom kernel

- CONFIG_DEVTMPFS
- CONFIG_CGROUPS (it is OK to disable all controllers)
- CONFIG_INOTIFY_USER
- CONFIG_SIGNALFD
- CONFIG_TIMERFD
- CONFIG_EPOLL
- CONFIG_NET
- CONFIG_SYSFS
- CONFIG_PROC_FS

- CONFIG_FHANDLE (libudev, mount and bind mount handling)
- CONFIG_SYSFS_DEPRECATED=n
- CONFIG_UEVENT_HELPER_PATH=”“
- CONFIG_FW_LOADER_USER_HELPER=n
- CONFIG_DMIID
- CONFIG_BLK_DEV_BSG
- CONFIG_NET_NS
- CONFIG_IPV6
- CONFIG_AUTOFS4_FS
- CONFIG_TMPFS_POSIX_ACL
- CONFIG_TMPFS_XATTR
- CONFIG_SECCOMP
- CONFIG_CGROUP_SCHED
- CONFIG_FAIR_GROUP_SCHED
- CONFIG_SCHEDSTATS
- CONFIG_SCHED_DEBUG

CHAPTER 17

Xorg

Dual View / Beamer

- Via config

```
Section "ServerLayout"
      Identifier     "X.org Configured"
      Screen      0  "Screen0" 0 0
      Screen      1  "Screen1" RightOf "Screen0"
      Option         "Xinerama" "true"
EndSection
```

- Or command

```
xrandr --output HDMI2 --right-of eDP1 --mode 1920x1200
```

Nvidia Xorg config

- Install nvidia driver and tools
- Use nvidia-xconfig
- Use nvidia driver not nv

Add new resolution

```
$ cvt 1024 600
# 1024x600 59.85 Hz (CVT) hsync: 37.35 kHz; pclk: 49.00 MHz
Modeline "1024x600_60.00"   49.00  1024 1072 1168 1312  600 603 613 624 -hsync +vsync
```

```
$ xrandr --newmode "1024x600"   49.00  1024 1072 1168 1312  600 603 613 624 -hsync␣
↪+vsync
$ xrandr --addmode default "1024x600"
```

- To activate the new mode execute

```
xrandr --output eDP1 --mode 1280x720
```

Keyboard config for German layout

- Paste into /etc/X11/xorg.conf.d/10-keyboard.conf

```
Section "InputClass"
      Identifier "keyboard"
      MatchIsKeyboard "yes"
      Option "XkbLayout" "de"
      Option "XkbVariant" "nodeadkeys"
EndSection
```

Change Keyboardlayout to dvorak

- setxkbmap dvorak
- setxkbmap de -variant dvorak

Learn dvorak

- http://learn.dvorak.nl

CHAPTER 18

Tools

Augeas

Overview

- Edit configuration files on the command line

Printing

- All known config files an values

```
augtool print
```

- One value of a config file

```
augtool print /files/etc/ssh/sshd_config/MaxSessions
```

Add / change a value

```
augtool -s set /files/etc/ssh/sshd_config/MaxSessions
```

- To change the third title in grub.conf

```
augtool -s set /files/etc/grub.conf/title[3] "Arch Linux"
```

Remove an entry

```
augtool -s rm /files/etc/ssh/sshd_config/MaxSessions
```

Collectd

General

- Delete all ~ and other backup files in /etc/collectd.d or you get tons of "Value is too old" error messages
- Graph aggregation http://collectd.org/wiki/index.php/Plugin:Aggregation

Check config file

```
collectd -ft
```

Debugging

- Add this to your config file

```
LoadPlugin logfile
<Plugin logfile>
  LogLevel debug
    File STDOUT
</Plugin>
```

- Start collectd in foreground

```
collectd -f -C /etc/collectd.conf
```

Example SNMP config

- For single values use `Table false`

```
<Plugin snmp>
  <Data "a_value">
        Type "gauge"
        Table true
        Values ".1.3.6.1.4.1.34097.9.80.1.1.6.1"
   </Data>
   <Host "some-host.domain.tld">
         Address "1.2.3.4"
         Version 1
         Community "public"
         Collect "a_value"
         Interval 5
   </Host>
</Plugin>
```

Graphite output plugin

```
LoadPlugin "write_graphite"
<Plugin "write_graphite">
 <Carbon>
```

```
   Host "127.0.0.1"
   Port "2003"
   Protocol "tcp"
   Prefix "collectd."
   EscapeCharacter "_"
   SeparateInstances true
   StoreRates false
   AlwaysAppendDS false
 </Carbon>
</Plugin>
```

Mongodb output

```
LoadPlugin write_mongodb
<Plugin "write_mongodb">
  <Node "default">
      Host "localhost"
      Port     "27017"
      Timeout    2000
      StoreRates true
  </Node>
</Plugin>
```

RRD output

- For using rrdcached (prefered method)

```
LoadPlugin rrdcached
<Plugin "rrdcached">
  DaemonAddress "unix:/var/run/rrdcached/rrdcached.sock"
  DataDir "/var/lib/collectd/rrd"
  CreateFiles true
</Plugin>
```

- For direct rrd

```
LoadPlugin rrdtool
<Plugin rrdtool>
   DataDir "/var/lib/collectd/rrd"
   CacheTimeout 120
   CacheFlush   900
   # default 3600, 86400, 604800, 2678400, 31622400
   # RRATimespan <seconds>
</Plugin>
```

Example tail file

```
LoadPlugin tail
<Plugin "tail">
  <File "/var/log/httpd/error_log">
    Instance "httpd_error"
    <Match>
```

```
      Regex "python"
      DSType "CounterInc"
      Type "counter"
      Instance "total"
    </Match>
  </File>
</Plugin>
```

Example exec plugin

- Source of script (e.g. /usr/bin/count_lines_in_file)

```
#!/bin/bash
HOSTNAME="${COLLECTD_HOSTNAME:-localhost}"
INTERVAL="${COLLECTD_INTERVAL:-60}"
FILE=$1

 while sleep "$INTERVAL"; do
   VALUE=`cat $FILE | wc -l`
   echo "PUTVAL \"$HOSTNAME/"`basename $FILE`"_count/counter\" interval=$INTERVAL N:
↪$VALUE"
 done
```

- Config for plugin

```
LoadPlugin exec
<Plugin exec>
  Exec "nobody" "/usr/bin/count_lines_in_file" "/var/log/httpd/error_log"
</Plugin>
```

Chrome

Extensions

- Ace Jump - https://chrome.google.com/webstore/detail/ace-jump/dffnomheiae
- Adblock Plus - https://chrome.google.com/webstore/detail/adblock-plus/cfhdojbkjhnklbpkdaibdccddilifddb
- Ghostery - https://chrome.google.com/webstore/detail/ghostery/mlomiejdfkolichcflejclcbmpeaniij
- ScriptBlock - https://chrome.google.com/webstore/detail/scriptblock/hcdjknjpbnhdoabbngpmfekaecnpajba
- WebRTC Block - https://chrome.google.com/webstore/detail/webrtc-block/nphkkbaidamjmhfanlpblblcadhfbkdm/related
- Spreed - https://chrome.google.com/webstore/detail/spreed-speed-read-the-web/ipikiaemdopojhpe
- Tab Ahead - https://chrome.google.com/webstore/detail/tab-ahead/naoaiblmpegfelhkapanmmaaghmi
- Last recent tab - https://chrome.google.com/webstore/detail/toggle-switch-recent-last/odhjcgnlbagjllfbilicalpigimhdcll

Shortcuts

- ctrl + l - locationbar
- ctrl + k - searchbar
- ctrl + u - source code
- ctrl + d - bookmark page
- ctrl + f - find in page
- ctrl + w - close tab
- ctrl + o - open file
- ctrl + n - new window
- ctrl + r - reload page
- ctrl + shift + r - reload page without cache
- ctrl + up + shift + t - restore recently closed tab(s)
- alt + 1-8 - switch to tab 1 - 8
- alt + <left> - back in history
- alt + <right> - forward in history
- alt + t - search tab (with tab ahead extension)
- alt + q - switch to last tab (with last recent tab extension)
- alt + j - jump to link or formular input (with ace jump extension)

Special pages

- chrome:about
- about:extensions
- about:plugins
- about:memory
- about:cache
- about:sandbox

Edit with Emacs

- Install chrome extension https://chrome.google.com/webstore/detail/edit-with-emacs/ljobjlafonikaiipfkggjbhkghgicgoh
- Install edit-server package in Emacs and eval

```
(require 'edit-server)
(edit-server-start)
```

- For more see http://www.emacswiki.org/emacs/Edit_with_Emacs

Emacs

Shortcuts

- https://github.com/balle/emacs/blob/master/emacs.org

SSH

- SSH over multiple hops http://www.gnu.org/s/tramp/#Multi_002dhops

Remote editing

- Use tramp with C-xf

```
ssh:user@host:file
```

- This script will ssh back to your machine and open the file you specified on the remote host
- Borrowed from https://gist.github.com/850795

```
#!/bin/sh
#
# Use this script as your EDITOR to allow editing remote files with emacsclient.
# Works by connecting to the Emacs machine with SSH and using a suitable tramp prefix.

# How to reach this machine from the one that's running Emacs
ME=user@remote-host

# How to reach the machine that's running Emacs from this machine
THEY=user@host-running-emacs

if [ "${1#/}" != "$1" ]; then
  # absolute path
  exec ssh $THEY "emacsclient /$ME:$1"
else
  # relative path
  PWD=$(pwd)
  exec ssh $THEY "emacsclient /$ME:$PWD/$1"
fi
```

Share a buffer over http

- Install the impatient-mode package.
- Call M-x httpd-start.
- Configure the firewall to allow incoming connections.
- Put the selected buffer into impatient-mode.
- Share the link with my IP address (form: http://my.ip.ad.dress:8080/imp/)
- Copied from http://sachachua.com/blog/2015/02/emacs-peer-peer-coaching-easier-use-impatient-mode-share-buffer/

How to Write a Emacs Major Mode for Syntax Coloring

- http://ergoemacs.org/emacs/elisp_syntax_coloring.html

Extensions

- https://github.com/balle/emacs/
- http://gabrielelanaro.github.com/emacs-for-python/
- https://github.com/pdee/pdee

Useful links

- http://emacswiki.org/
- http://stackoverflow.com/questions/tagged/emacs

Firefox

Extensions

- NoScript - https://addons.mozilla.org/en-US/firefox/addon/noscript/
- Cookie manager - https://addons.mozilla.org/en-US/firefox/addon/cookie-controller/
- Flashblock - https://addons.mozilla.org/en-US/firefox/addon/flashblock/
- Better privacy - https://addons.mozilla.org/de/firefox/addon/betterprivacy/
- Disable WebRTC - https://addons.mozilla.org/de/firefox/addon/happy-bonobo-disable-webrtc/
- Download Helper - https://addons.mozilla.org/en-US/firefox/addon/video-downloadhelper/
- Session Manager - https://addons.mozilla.org/en-US/firefox/addon/session-manager/
- Firemacs - https://addons.mozilla.org/en-US/firefox/addon/firemacs/
- It's all text - https://addons.mozilla.org/en-US/firefox/addon/4125
- LoL - http://elder-gods.org/lol
- Mouseless browsing - https://addons.mozilla.org/de/firefox/addon/mouseless-browsing/
- Customizable Shortcuts - https://addons.mozilla.org/de/firefox/addon/customizable-shortcuts/
- Tile Tabs - https://addons.mozilla.org/de/firefox/addon/tile-tabs/
- Foxyproxy - https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/
- Live HTTP headers - https://addons.mozilla.org/en-US/firefox/addon/live-http-headers/
- Selenium - http://docs.seleniumhq.org/download/

Shortcuts

- ctrl + l - locationbar
- ctrl + k - searchbar
- ctrl + u - source code
- ctrl + d - bookmark page
- ctrl + shift + o - open bookmark manager
- ctrl + shift + a - open addon manager
- ctrl + j - show downloads
- ctrl + shift + e - switch tabs
- alt + 1-8 - switch to tab 1 - 8
- ctrl + w - close tab
- ctrl + left / right - move tab left / right
- ctrl + page up / down - goto previous / next tab
- ctrl + shift + k - web console
- shift + f4 - open javascript scratchpad
- ctrl + shift + j - open error console
- ctrl + f - find
- alt + left / right - history back / forward
- ctrl + o - open file
- ctrl + n - new window
- ctrl + r - reload page
- ctrl + shift + del - clear history

Privacy

- Use flashbock, adblock, noscript and better privacy addon
- Deactivate Security -> Detect attack / forgery sites
- Block popups
- Deactivate history
- Use privoxy as proxy to filter thinks like browser identity string
- Optionally use tor as forwarding proxy in privoxy

Goto last tab

- browse about:config
- set browser.ctrlTab.previews to true
- Now you can switch to last tab with ctrl + tab

Mouseless browsing

- ctrl + space - show ids
- ctrl + id + enter - click on id
- alt + id - open id in new tab
- ctrl + alt + id - open id in new window

Parameterized bookmarks

- Bookmark a page
- Open bookmark in bookmark manager
- Click on "More" arrow
- Enter a keyword "e.g. g"
- Set %s placeholder somewhere in url (e.g. http://google.com/q=%s)
- Now you can use "g test" in location bar
- See http://johnbokma.com/firefox/keymarks-explained.html

Synchronization

- Tools -> Setup Sync
- Will (depending on config) synchronize tabs, bookmarks, history, settings, addons and passwords
- You can setup your own firefox sync server see http://docs.services.mozilla.com/howtos/run-sync.html

Graphite

Setup graphite-web

- After installation append the following to `/etc/graphite/local_settings.py`

```
from graphite.app_settings import *

SECRET_KEY='eiThoo6/biephe7Fs.o7aiZeu=eD8sho!ahQuah2+yiegh9Ai,6euP8hAequa7ee'
```

- Of course make sure to generate your own secret key
- Adjust apache configuration file `/etc/httpd/conf.d/graphite-web.conf` and add

```
 <Location "/">
    AllowOverride All
    Require all granted
</Location>
```

- If the database was not created by your packagemanagement go to */usr/lib/python2.6/site-packages/graphite* and exec

```
./manage.py syncdb
```

- Last but not least add graphite-web as hostname to localhost in `/etc/hosts` or whatever ip you like

Send data to graphite

- Via Collectd
- Using Statsd
- With Icinga
- By help of Diamond
- Or in Bash script

```
echo "localhost.hello.world `ps ax | wc -l` `date +%s`" | nc <graphite-srv> 2003
```

- For windows systems use SSC Serv

Graphite Web API

- See http://graphite.readthedocs.org/en/latest/render_api.html

Carbon config

- */etc/carbon/carbon.conf* defines limits for caching, flush interval, forwarding and aggregation
- */etc/carbon/storage-schemas.conf* configures how long the data will be stored, in which frequency and when it gets overwritten (remember its a round-robin like RRD)

```
[collectd]
pattern = ^collectd\..*
rententions = 1s:1d,1m:7d,1h:30d,1d:2y
```

Fetch some data

```
whisper-fetch --from=`date +%s -d "2014-12-24 00:00:00"` --until=`date +%s` /var/lib/
↪carbon/whisper/<some_database.wsp>
```

Dump database

```
whisper-dump /var/lib/carbon/whisper/<some_database.wsp>
```

Resize database

```
whisper-resize /var/lib/carbon/whisper/<some_database.wsp> 1s:7d
```

Convert RRD to Whisper

```
rrd2whisper <path_to_rrd>
```

Lsof

Open files by ...

- Pid

```
lsof -p <pid>
```

- User

```
lsof -u <pid>
```

- Program

```
lsof -c <progname>
```

Which command is using this port?

```
lsof -i :<port>
```

Which processes have an open TCP socket to remote-site

```
lsof -i TCP@remote:port
```

Which processes are using this file?

```
lsof /path/to/file
```

What pids does that binary have?

```
lsof -t /path/to/command
```

Find all open files in a directory

```
lsof +D /some/dir
```

Mc

Keyboard shortcuts

- alt + s - search
- ctrl + \ - directory hotlist
- ctrl + h - add current directory to hotlist
- ctrl + x d - compare directories
- + - enter regexp all files matching regexp get marked
- \ - enter regexp all files matching get unmarked
- alt + t - toggle view
- ctrl + o - background mc
- f9 - menus

Use SFTP

- f9
- choose left or right
- Shell connection
- user@host
- or quick way (in term) *cd /#sh:user@host*

Tips

- activate f9 -> options -> configuration -> lynx like actions
- now you can leaver directory with left arrow

Monitoring

Cacti

- Create new host: Console -> Create devices -> Add (right top corner)
- Add host to graph tree: Graph Trees -> Add -> Tree Item Type Host
- Reconfiguring data sources doesnt seem to work. Better delete and recreate.
- Cumulative graphs: Install Aggregate Plugin than Graph Management -> Select graphs -> Action Create Aggregate Graph

Munin

- http://munin-monitoring.org/wiki/Using_SNMP_plugins
- http://munin-monitoring.org/wiki/HowToWriteSNMPPlugins
- value (and only value!) must be returned after config section

```
#!/usr/bin/perl -w

=head1 MAGIC MARKERS

  #%# family=snmpauto
  #%# capabilities=snmpconf

=cut

use strict;
use Munin::Plugin::SNMP;

my $oid = "1.3.6.1.4.1.34097.9.77.1.1.17.1";

if (defined $ARGV[0] and $ARGV[0] eq "snmpconf") {
      print "require $oid.0 [0-9]\n"; # Number
      exit 0;
}

if (defined $ARGV[0] and $ARGV[0] eq "config") {
  my ($host) = Munin::Plugin::SNMP->config_session();
      print "host_name $host\n" unless $host eq 'localhost';
      print "graph_title Power A Total
graph_vlabel A
graph_category power
graph_info Power in A
";

   print "power_total_a.label Total\n";
   print "power_total_a.info Total\n";
   print "power_total_a.draw LINE2\n";

   exit 0;
}

my $session = Munin::Plugin::SNMP->session();

my $power_a = $session->get_single ("." . $oid . ".0") || 'U';

if($power_a ne "U")
{
    $power_a /= 1000;
}

print "power_total_a.value ", $power_a, "\n";
```

- Aggregate graphs http://munin-monitoring.org/wiki/aggregate_examples
- Tons of plugins https://github.com/munin-monitoring/contrib/

Cluster graphing

- http://ganglia.sourceforge.net/
- Install gmond on all nodes
- Install gmetad and php webfrontend on monitor host

Puppet

Overview

- Firewall port tcp 8140
- Sample puppet.conf

```
[main]
  vardir = /var/lib/puppet
  logdir = /var/log/puppet
  rundir = /var/run/puppet
  ssldir = $vardir/ssl

[agent]
  classfile = $vardir/classes.txt
  localconfig = $vardir/localconfig
  listen = True
  runinterval = 3600

[master]
  manifestdir = /etc/puppet/manifests
  modulepath = /etc/puppet/modules
  autoflush = true
  autosign = true
```

- `manifests/site.pp` tells puppet what client configurations to load

```
import 'nodes.pp'
# import 'nodes/*'
$puppetserver = "name-of.master.net"
```

- `manifest/notes.pp` include the client configuration

```
node base {
   include module1, module2
}

node 'full.hostname.tld' inherits base {
  include another_module
}
```

Classes

- Group some code blocks like package, file and service

```
class a_name($param = "default value") {}
```

Definitions

- Definitions are like functions
- Useful because classes are singletons (can be only instanciated once)

```
define cronjob( $hour = '00', $minute = '00' ) {
  cron { "$name":
    command => "/opt/cronjobs/$name",
    hour    => $hour,
    minute  => $minute,
  }
}

cronjob { 'blah':
  hour => 23,
}
```

Modules

- Create dirs

```
mkdir -p /etc/puppet/modules/$modulename/{files,templates,manifests}
```

- Create `manifests/init.pp`

```
class emacs {
  package { emacs:
    ensure => present
  }
}
```

- Search for existing modules on puppetlabs

```
puppet module search <term>
```

- Install / uninstall a module

```
puppet module install puppetlabs-openstack
puppet module uninstall puppetlabs-openstack
```

Install software

```
package { 'emacs': ensure => present }
```

Copy files

```
file { "/root/.emacs":
  owner => "root",
  group => "root",
  mode => 0440,
  source => "puppet://$puppetserver/modules/emacs/.emacs"
  require => Package["emacs"]
}
```

- File must be on master server in `/etc/puppet/modules/emacs/files/.emacs`

Change a file

- Append

```
file_line {
  'a comment':
    path => '/path/to/a/file',
    line => 'append this content';
}
```

- Edit specific entry

```
file_line {
  'a comment':
    path => '/path/to/a/file',
    match => '^#?some content',
    line => 'New content';
}
```

- Or using augeas

```
augeas { "nova.conf":
  context => "/files/etc/nova/nova.conf",
  changes => [
                "set rpc_backend nova.rpc.impl_kombu",
              ],
}
```

Directory

```
file { "/root/.emacs":
  ensure => 'directory',
  owner => "root",
  group => "root",
  mode => 0440,
}
```

Link

```
file { "/link/from/here":
  ensure => 'link',
  target => '/link/to/here',
}
```

Adding users

```
user { "testuser":
  ensure => present,
  uid => 10001,
  gid => 10001,
  shell => "/bin/zsh",
  home => "/home/testuser",
  comment => "Just a test",
  password => "$hash",
  managehome => true,
}
```

- To generate the password hash use

```
openssl passwd
```

SSH keys

```
ssh_authorized_key { "testuser":
  ensure => present,
  type => "ssh-rsa",
  key => "",
  user => "testuser",
  require => User["testuser"],
}
```

Starting services

```
class ssh::service {
  service { "sshd":
    ensure => running,
    hasstatus => true,
    hasrestart => true,
    enable => true,
  }
}
```

- hasstatus and hasrestart tells puppet if the init script understand the parameter status and restart
- A file can trigger a service restart by adding `notify => Class["ssh::service"]`
- To stop a service use `ensure => stopped,`

Cronjobs

```
cron { 'Make Backup':
  command => 'tar cvzf /data/backup,tgz /home',
  hour    => '00',
```

```
  minutes => '00',
}
```

Selective Execution

```
exec { 'apache restart':
  command => 'apachectl restart',
  unless  => 'ps ax | grep apache',
}
```

- with `cwd` one can change working directory
- `path` will set PATH env var
- or exec on file change

```
exec { 'myscript':
  command => 'whatever',
  refreshonly => true,
  subscribe => File['/path/to/some_file'],
}
```

Deleting stuff

```
ensure => absent,
```

Templates

- Templates are used to create files depending on facter and config variables

```
myhostname = <%= hostname %>

<% if a_flag == 1 -%>
  config_a = 123
<% elsif a_flag == 2 -%>
  config_b = 321
<% else -%>
  do something totally different
<% end -%>
```

- Can be included in files using `content = template("template_file.erb")`
- To access variables from other packages use `<%= scope.lookupvar('my_package::some_var') %>`

Config controls

```
if $host == '' {
  $srvname = $title
} else {
  $srvname = $servername
```

```
}
case $operatingsystem {
  'centos', 'redhat', 'fedora': { $vdir    = '/etc/httpd/conf.d'
                                  $logdir = '/var/log/httpd'}
  'ubuntu', 'debian':           { $vdir    = '/etc/apache2/sites-enabled'
                                  $logdir = '/var/log/apache2'}
  default:                      { $vdir    = '/etc/apache2/sites-enabled'
                                  $logdir = '/var/log/apache2'}
}
```

Firewall config

- First install firewall module
- Setup firewall in site.pp

```
resources { "firewall":
  purge => true
}

Firewall {
  before  => Class['my_fw::post'],
  require => Class['my_fw::pre'],
}

class { ['my_fw::pre', 'my_fw::post']: }
class { 'firewall': }

class my_fw::pre {
  Firewall {
    require => undef,
  }

  # Default firewall rules
  firewall { '00000 accept all icmp':
    proto   => 'icmp',
    action  => 'accept',
  }->
  firewall { '00001 accept all to lo interface':
    proto   => 'all',
    iniface => 'lo',
    action  => 'accept',
  }->
  firewall { '00002 accept related established rules':
    proto   => 'all',
    state   => ['RELATED', 'ESTABLISHED'],
    action  => 'accept',
  }
}

class my_fw::post {
  firewall { '99999 drop all':
    proto   => 'all',
    action  => 'drop',
    before  => undef,
  }
}
```

- Make sure pluginsync is enabled in `puppet.conf` in section `[main]`
- If you dont want to delete the firewall or just add rules with puppet set `purge => false`
- If you want to configure bridge interfaces / kvm server see patch for "Could not evaluate: Invalid address from IPAddr.new" on https://github.com/puppetlabs/puppetlabs-firewall/issues/141

```
pluginsync = true
```

- The comment must contain an index to get the order of the rules

```
firewall { "00001 a comment":
  proto => 'tcp',
  iniface => 'eth0',
  dport => 22,
  action => 'accept',
}
```

SELinux

- Boolean

```
selboolean { "a comment":
  name => "httpd_enable_cgi",
  value => 'off',
}
```

- File context

```
class selinux::fcontext ( $context = "", $pathname = "" ) {
  if ( $context == "" ) or ( $pathname == "" ) {
    fail("context and pathname must not be empty")
  }

  exec { "add_${context}_${pathname}":
    command => "semanage fcontext -a -t ${context} \"${pathname}(/.*)?\" &&␣
↪restorecon -RFvv \"${pathname}\"",
    unless  => "semanage fcontext -l|grep \"^${pathname}.*:${context}:\"",
    path    => '/usr/sbin:/bin',
  }
}

class { "selinux::fcontext":
  context => "mysqld_log_t",
  pathname => "/var/log/mysql(/.*)?",
}
```

- Policy module

```
selmodule { "load a policy":
  ensure => present,
  selmoduledir => "/path/to/policy",
  name => "filename_without_pp",
}
```

Checkout from Git / Subverson / CVS...

M#* Install https://github.com/puppetlabs/puppetlabs-vcsrepo

```
vcsrepo { "/path/to/repo":
  ensure => present,
  provider => git,
  source => 'git://example.com/repo.git',
  revision => 'master'
}
```

Executing ruby code

```
require SecureRandom
#...
myparam => inline_template("<%= SecureRandom.hex(20) %>"),
```

Cert handling

- List

```
puppet cert --list
```

- Sign

```
puppet cert --sign <hostname>
puppet cert --sign --all
```

- Delete

```
puppet cert clean <hostname>
```

Environments

- Add the following to puppet.conf

```
[main]
  modulepath = $confdir/modules
  manifest = $confdir/manifests/site.pp

[devel]
  modulepath = $confdir/devel/modules
  manifest = $confdir/devel/manifests/site.pp
```

- Now you can tell a puppet agent to use the devel environment by adding `--environment devel`

Syntaxcheck a manifest

- Check syntax of a file

```
puppet parser validate <some_file.pp>
```

- Test a whole module

```
puppet apply --noop <manifests/init.pp>
```

Getting help

- http://docs.puppetlabs.com/puppet/3/reference/
- http://www.puppetcookbook.com
- get a list of all known resources

```
puppet describe --list
```

- doc about a resource

```
puppet describe -s <keyword>
(+ 1 2)
```

Debugging

- Master

```
puppet agent --no-daemonize --verbose
```

- Agent

```
puppet agent --no-daemonize --verbose --test --noop
```

- Use `--debug` instead of `--verbose` for even more output
- You can use the `notice("foo")` command somewhere to send a log message
- See `/var/lib/puppet/state/last_run_report.yaml` for information update last update

RRD

Overview

- RRDs (Round Robin Databases) consist of DS (Data Source) that will be saved in RRAs (Round Robin Archives)
- Each RRA has a fixed size of slots, that will be automatically rotated
- Every update on a rrd triggers every RRA to be updated by a consolidation function (CF) like MIN, MAX or AVERAGE

Create a rrd

- DS defines Data Source with name load, type GAUGE (just save the value as is), every 60 seconds we expect a value between 0 and unlimited
- The first RRA has 60 slots and saves a single value in it if it greater than 0.1
- The second RRA has 4 slots and saves the MAX value of the last 15 values

```
rrdtool create some.rrd DS:load:GAUGE:60:0:U RRA:AVERAGE:0.1:1:60 RRA:MAX:0.1:15:4
```

Show meta data about a rrd

```
rrdtool info some.rrd
```

Insert values

```
rrdtool update some.rrd `date +%s`:`cat /proc/loadavg| cut -d " " -f 1`
```

Display values

- from yesterday till now

```
rrdtool fetch some.rrd AVERAGE --start `date +%s -d "yesterday"` --end `date +%s`
```

- from specific timespan

```
rrdtool fetch some.rrd MAX --start `date +%s -d "2013-02-15 18:00:00"` --end `date +
↪%s -d "2013-02-15 19:00:00"`
```

Create a graph image

- The graph will be drawn as LINE with a width of 2 pixel and the color red

```
rrdtool graph test.png  --start `date +%s -d "yesterday"` --end `date +%s`␣
↪DEF:myload=some.rrd:load:AVERAGE LINE2:myload#FF0000
```

Calculating stuff

- Graph values multiplied by 10
- Add the following to your graph command

```
CDEF:myvalue=myload,10,\*
```

- To really graph the ne myvalue add

```
LINE2:myvalue#0000ff
```

Aggregate two graphs

- First define two DEFs
- Add both by using a CDEF

```
rrdtool graph test.png --start `date +%s -d "yesterday"` --end `date +%s`␣
↪DEF:load1=some.rrd:load:AVERAGE DEF:load2=another.rrd:load:AVERAGE CDEF:total=load1,
↪load2,\+ LINE2:total#0000ff
```

Export data as XML

- Everything

```
rrdtool dump some.rrd
```

- Or specific

```
rrdtool xport --start `date +%s -d "yesterday"` --end `date +%s` DEF:load1=some.
↪rrd:load:AVERAGE XPORT:load1
```

Resize Databases

```
for DIR in $(ls); do echo "Entering $DIR"; cd $DIR/snmp; for RRD in $(find . |grep␣
↪rrd|grep -v resize); do echo "Resizing $RRD"; rrdtool resize $RRD 12 GROW 10800; mv␣
↪-f resize.rrd $RRD; rrdtool resize $RRD 13 GROW 10800; mv -f resize.rrd $RRD;␣
↪rrdtool resize $RRD 14 GROW 10800; mv -f resize.rrd $RRD; done; cd ../..; done
```

sar

CPU

- Every 10 seconds

```
sar 10
```

- Every 60 seconds for 5 minutes

```
sar 60 5
```

- Show single cores

```
sar -P ALL
```

Load

```
sar -q
```

Mem and swap

```
sar -rS
```

Disk

```
sar -bd
```

Network

```
sar -n ALL
```

Inodes and file descriptors

```
sar -v
```

Processes

```
sar -w
```

Continuos monitoring

```
/usr/lib/sa/sadc 60 -
```

- Logs can now be found in /var/log/sa
- Show all logs from current day

```
sar -f -
```

Screen

Basics

- ctrl+a c - new window
- ctrl+a n - next window
- ctrl+a p - previous window
- ctrl+a a - last window
- ctrl+a <nr> - switch to window <nr>
- ctrl+a A - rename window
- ctrl+a k - kill window

Splitting

- ctrl+a S - split window
- ctrl+a tab - switch between splittet screens
- ctrl+a X - close window

Detatching

- ctrl+a d - detach screen -r # attach
- ctrl+a x - lock screen

Copy & paste

- ctrl+a esc - get into copy mode
- space - set marker begin / end
- W - mark whole word
- Y - mark whole line
- ctrl+a] - paste

Searching

- ctrl+a esc - get into copy mode
- ctrl+r - search backward
- ctrl+s - search forward

Example .screenrc

```
startup_message off
vbell on
autodetach on
defscrollback 10000

bindkey -m ' ' eval 'msgwait 0' 'stuff \040' writebuf 'exec !!! xclip /tmp/screen-
↪exchange' 'msgwait 2'
bindkey -m Y eval 'msgwait 0' 'stuff Y' writebuf 'exec !!! xclip /tmp/screen-exchange
↪' 'msgwait 2'
bindkey -m W eval 'msgwait 0' 'stuff W' writebuf 'exec !!! xclip /tmp/screen-exchange
↪' 'msgwait 2'

bind r eval 'echo "Resize window"' 'command -c resize'""
bind -c resize "+" eval 'resize +1' 'command -c resize'
bind -c resize "-" eval 'resize -1' 'command -c resize'

caption always  "%{=b kw} $LOGNAME@%H %c %D %d/%m/%Y %{=b kr}|%{-} %l %u %{=b kr}|%{-}
↪ %-Lw%{=b kr} %50>%n%f*%t %{-}%+Lw%<"
```

Smartmon Tools

- health check

```
smartctl -H <dev>
```

- short self test

```
smartctl -t short <dev>
```

- long self test (better run at night)

```
smartctl -t long <dev>
```

- test results

```
smartctl -l selftest <dev>
```

Tmux

Overview

Tmux is a terminal multiplexer like screen.

To attach to a running tmux session type

```
tmux attach -t sessionid or name
```

If no session exists a new one will be created.

To start Tmux togehter with your shell you can put the following into your .zshrc or .bashrc

```
[[ $TERM != "screen" ]] \&\& tmux \&\& exit
```

Shortcuts

- Ctrl + b d -> dettach
- Ctrl + b c -> new window
- Ctrl + b 0 -> goto window 0
- Ctrl + b ” -> split window V
- Ctrl + b % -> split window horizontal in 2 pannel spliten
- Ctrl + b w -> show windowlist
- Ctrl + b ! -> close all window
- Ctrl + b n -> next window
- Ctrl + b p -> previous window
- Ctrl + b l -> last window
- Ctrl + b , -> rename window

- Ctrl + b k -> delete window
- ctrl + b x -> delete current pane
- Ctrl + b o -> switch panel
- ctrl + b up/down -> switch to panel up/down
- ctrl + b q <nr> -> show panel numbers and switch to it directly
- ctrl + b ! -> move current pane to new window
- ctrl + b s -> send pane to other window
- Ctrl + b [-> switch to buffer (like emacs)
- Ctrl + Space -> set marker
- Ctrl + w -> cut region
- Alt + w -> copy region
- Esc -> leave buffer
- Ctrl + b] -> paste last copied buffer
- ctrl + b = -> choose paste buffer
- ctrl + b : capture-pane -> copy all visible pane output
- ctrl + b : save-buffer -> write paste buffer to file
- ctrl + space -> toggle different pane layouts
- ctrl + b : resize-pane -D 20 -> shrink down
- ctrl + b : resize-pane -U 20 -> shrink up
- ctrl + b <left/right/up/down> -> resize current pane to left/right/up/down
- ctrl + b : break-pane -> convert pane to window
- ctrl + b : source-file ~/.tmux.conf -> reload config

Session handling

- Create a new session named muh

```
tmux new -s muh
```

- List all sessions

```
tmux ls
```

- Attach to a session

```
tmux attach -t <session-name>
```

- Detach from a session with Ctrl b + d
- Kill a session

```
tmux kill-session -t <name-or-number>
```

Scripting

```
#!/bin/bash

for IP in {1..96}; do
  tmux select-layout tiled
  tmux split-window -h
  tmux send-keys "ssh root@192.168.1.$IP" C-m
  tmux send-keys "top" C-m
done
```

Synchronous input

- ctrl + b : synchronize-panes

Getting help

ctrl b ? - show keys ctrl b : list-commands

Balle Config

```
#!/bin/bash

# Make it use C-a, similar to screen..
unbind C-b
unbind l
set -g prefix C-a
bind-key C-a last-window
bind-key k kill-window
bind-key -n C-M-d set-window-option synchronize-panes off
bind-key -n C-M-c set-window-option synchronize-panes on

# Reload key
bind r source-file ~/.tmux.conf

set -g default-terminal "screen-256color"
set -g history-limit 100000
set -g status-interval 1

#--Status-Bar---------------------------------------------------------------------
# Default colors
set -g status-bg black
set -g status-fg white

# Left side of status bar
set -g status-left-length 20
set -g status-left ''
#set -g status-left '#[fg=green][#[bg=black,fg=cyan]#S#[bg=black,fg=red,dim]:#H
↪#[fg=green]]'

# Inactive windows in status bar
set-window-option -g window-status-format '#[fg=cyan,dim]#I#[fg=blue]:#[default]#W
↪#[fg=grey,dim]#F'
```

```
# Current or active window in status bar
set-window-option -g window-status-current-format '#[bg=red,fg=cyan,bold]#I#[bg=red,
↪fg=cyan]:#[fg=white]#W#[fg=dim]#F'

# Alerted window in status bar. Windows which have an alert (bell, activity or␣
↪content).
#set-window-option -g window-status-alert-fg red
#set-window-option -g window-status-alert-bg white

set -g status-right-length 50
set -g status-right '#[fg=yellow]#(cut -d " " -f 1-3 /proc/loadavg)#[default]
↪#[fg=green]#(whoami)@#h#[default] #[fg=blue]%H:%M:%S %d/%m#[default]'

# enable activity alerts
setw -g monitor-activity on
set -g visual-activity on

# resize screen only for active clients
setw -g aggressive-resize on

bind-key C-s set-window-option synchronize-panes

# bind arrow keys
bind-key -n C-up select-pane -t :.+
bind-key -n C-down new-window

bind-key | split-window -h
bind-key - split-window -v

# pane movement
bind-key j command-prompt -p "join pane from:"  "join-pane -s '%%'"
bind-key s command-prompt -p "send pane to:"  "join-pane -t '%%'"

# pane resize
bind-key C-u resize-pane -U     # Resize window up              (Ctrl+b, u) (i.e.,␣
↪hold Ctrl and alternate hitting 'b' and 'u')
bind-key C-d resize-pane -D     # Resize window down            (Ctrl+b, d) (similar)
bind-key C-l resize-pane -L     # Resize window left            (Ctrl+b, l) (similar)
bind-key C-r resize-pane -R     # Resize window right           (Ctrl+b, r) (similar)

# browsing urls
bind-key u capture-pane \; save-buffer /tmp/tmux-buffer \; new-window -n "urlview" '
↪$SHELL -c "urlview < /tmp/tmux-buffer"'

# Screen lock
bind-key C-x   lock-server
set-option -g   lock-after-time 0
set-option -g   lock-server on
#set-option -g   lock-command "vlock"

# better copy & paste
bind-key C-c run "tmux save-buffer - | xclip -i sel clipboard"
bind-key C-v run "tmux set-buffer \"$(xclip -o sel clipboard)\"; tmux paste-buffer"
bind-key C-y paste-buffer
bind-key M-y choose-buffer

# plugins
```

```
#set -g @tpm_plugins "              \
#  tmux-plugins/tpm                 \
#  tmux-plugins/tmux-copycat        \
#  tmux-plugins/tmux-yank          \
#  tmux-plugins/tmux-open          \
#"
#run-shell ~/.tmux.d/tpm/tpm
```

- For browsing urls in firefox edit `~/.urlview`

```
COMMAND exec >> /tmp/urlview.out 2>&1; set -x; firefox
```

Tmux plugins

- https://github.com/tmux-plugins

Zsh

History

- !13 -> ruft den 13 Befehl der History aus
- !-1 -> genau wie !!-> ruft den gesamten letzten Befehl auf
- !-2 -> ruft den vorletzten Befehl auf usw.
- !:2 -> ruft nur das 2 Argument des letzten Befehls auf
- !:2:h -> ruft den vorderen Teil des letzten Arguments auf
- !-2:2:t -> ruft den hinteren Teil des vorletzten Befehls auf
- !:2:r.pdf -> ändert das Suffix des letzten Befehls (z.B txt) in pdf
- !!:s/z/y/ -> ändere im letzten Befehl das ERSTE z in y
- !!:gs/z/y/ -> ändere im letzten Befehl ALLE z in y
- history -D -> zeigt letzten Befehle an mit Nummer und Startzeit
- history -40 -20 -> zeigt alle letzten Eintrage zwischen dem 20 - 40
- !ca -> letzter Befehl der mit ca anfing z.B cal 12 2011
- !?pwd -> sucht nach dem letzten Befehl in dem pwd vorkam und führt ihn aus
- `history -EDd` shows history with timestamps and how long a command had run
- vom letzten Befehl bestimmte Zeichen löschen oder ersetzen

```
cal 1222 2011
```

- ^12 -> ruft cal 12 2011 auf
- ^cal^echo -> ersetzt das erste cal mit echo und gibt 12 2011 in stdout aus
- Pfade wiederverwenden

```
ls /usr/local/src
!!/blub                -> ls /usr/local/src/blub              -> Wiederverwendung des␣
↪gesamten letzten Befehls und hängt
                                                                 hinten "blub" an
cp !$/something .      -> cp /usr/local/blub/something .      -> wiederverwenden des␣
↪letzten Arguments des letzten Befehls
```

Output redirect

- cat bla > file.txt > error.txt-> redirect output to file.txt and errors to error.txt
- cat bla >| filename -> redirect output to File nur wenn diese NICHT existiert
- mehrere Pfade /Dateien hintereinander ausgeben

```
ls /etc /bin /usr -> gibt hintereinander /etc dann /bin dann /usr aus
cp file file0 file1 file3 /home  -> kopiert alle Dateien nach /home
```

- nach dem Anfang oder Ende eines Befehls oder Datei suchen

```
ls /usr/bin /bin | grep 'sh$'     -> findet alle Dateien die mit sh Enden in /usr/bin␣
↪und /bin
ls /home | grep  test*                  -> findet alle Dateien mit dem Anfang test im␣
↪/home
```

- alles in einem Verzeichnis ausgeben bis auf gestimmte Sachen

```
ls /usr/bin /bin | grep -v 'sh$'      -> gibt alles in den Verzeichnissen /usr/bin␣
↪und /bin aus bis auf das was mit sh endet
```

- Sachen sortiert ausgeben mit sort

```
ls /usr/bin /bin | grep 'sh$' | sort  -> gibt die Dateien von a - z sortiert aus
```

- Ausgabe in mehrere Spalten mit column

```
ls /usr/bin /bin | grep 'sh$'| column -> gibt die Dateien in so vielen Spalten wie␣
↪auf den Bildschirm passen aus
```

- im Befehl cat kann die Spalten Anzahl mitten -c angeben werden

```
cat -c 1-3 bla           -> Ausgabe von bla in 3 Spalten
```

- lange Befehleketten können mittels newline unterteilt werden

```
ls /usr/bin /bin | \
grep 'sh$' | \
sort | \
column \
```

Prozesse eines Terminals

- Ctrg+z -> Prozess im Vordergrund im Hintergrund schlafen legen (suspend)
- Ctrg+c -> Tötet den Prozess im Vordergrund

- Ctrg+-> Tötet den Prozess + Core Dump
- Ctrg+R -> Rückwärts suchen
- jobs -> zeigt alle Prozesse des Terminals mit Nummern an
- bg %Prozessnummer -> Restartet schlafende Prozesse im Hintergrund
- fg %Prozessnummer -> holt Prozesse aus dem Hintergrund in den Vordergrund
- Ctrg+z + bg %Prozessnummer -> Prozesse in den Hintergrund schieben die im Vordergrund laufen

ZSh Bindings

- bindkey -L -> listet alle zsh Bindings auf
- bindkey 'C-w' kill-region -> setzt zsh keybinding für kill-region auf Ctrg+w (für dauerhafte Bindings einfach in die .zshrc eintragen)
- read -> liest einen Buchstaben ein und gibt seine escapde Form aus für ein bindkey
- bindkey -s 'C-ff' "firefox" -> Bindet den String firefox an die Tastenkürtzel Ctrg+ff
- stty -a -> zeigt alle Terminal bindings und mehr an
- Terminal bindings ändern mit
- stty intr '^t' -> Interrupt jetzt nicht mehr Ctrg+c sondern Ctrg+t

Keybindings für BASH ZSH und Emacs

- Alt + b -> Wort zurück springen
- Alt + f -> Wort vor springen
- Alt + Backspace -> Wort vor Cursor löschen
- Alt + d -> Wort nach Cursor löschen
- Alt + h -> manpage des Befehls aufrufen
- Alt + Q -> Befehl für eine Zeile in den Hintergrund schieben
- Strg + l -> clean screen
- Strg + x Strg +x -> Zeile von Cursor bis Anfang der Zeile markieren (EMACS markiert Block)
- Alt + . -> Argumente der vorhergehenden Befehle abrufen (nur BASH und ZSH)
- Strg + Space -> Marke setzen
- Strg + k -> Ende der Zeile löschen von Cursor
- Strg + w -> Anfang der Zeile löschen bis Cursor
- Strg + u -> ganze Zeile löschen
- Strg + s -> suche vorwärts
- Strg + r -> suche rückwärts
- Strg + s -> Output Pause und Strg + Q -> Fortsetzen der Ausgabe

Directory Stack

- dirs -v -> zeigt alle Directory des Stacks an
- ~ -> Home dir
- ~person -> home dir von person
- ~- -> letztes Verzeichnis
- ~3 -> 2 Verzeichnis im Stack

Befehle finden

- type acroread -> zeigt das Verzeichnis vom Adobe Reader an
- which firefox -> "" von Firefox
- whence -M '*fg' -> gibt alle Befehle die auf fg-enden mit vollen Pfad aus
- ls *.{c,h,o} -> gibt alle Dateien aus die auf c, h oder o enden
- echo {1..10} -> gibt alle Zahlen von 1 bis 10 aus

Pattern Matching

- Dateien finden & Pattern Matching (ls ist doch Befehle wie chmod, print oder echo etc. ersetzbar)
- ls * -> alle Dateien und Verzeichnisse in diesem Verzeichnis (0 bis n Zeichen) ausgeschlossen sind . Dateien!
- ls **/ -> alle Dateien und Verzeichnisse in diesem Verzeichnis und allen Unterverzeichnissen ausgenommen . Dateien
- ls /*/ -> -> alle Dateien und Verzeichnisse in allen Unterverzeichnissen ausgenommen . Dateien
- ls .* -> listet alle .Dateien diese Verzeichnisses
- ? -> ein Zeichen
- [abc] -> ein a,b, oder c
- [a-z] -> ein Zeichen zwischen a bis z
- [1A-Z] -> 1 oder ein Zeichen zwischen A bis Z
- [^a-z] -> kein Zeichen zwischen a bis z das selbe wie
- (doc|txt) -> entweder txt oder doc
- <1-9> -> Zahlen zwischen 1 - 9
- pat1~pat2 -> das Pattern vor der ~ soll gesucht werden und danach alle Ergebnisse mit dem Pattern nach der ~ entfernt werden
- # -> 0, 1 oder mehrfaches auftreten von Zeichen oder [] ()
- (i#)read(I#) -> sucht nach allen groß und klein geschriebenen read
- Globale Qualifier müssen immer am ende des Pattern in () Klammern stehen
- . -> nur reguläre Dateien keine Verzeichnisse oder Links
- @ -> nur Links
- / -> nur Verzeichnisse

- – -> ausführbare Dateien (keine Verzeichnisse)
- f:u+rwx,o-rwx: -> Dateirechte (hier User hat read write execute others haben kein read write execute
- Lk+100 -> Filegröße (hier Kilobyte größer 100) m für Megabyte, - für kleiner
- mh-1 -> File Timestamp (hier kleiner eine Stunde) m für Minuten, + für mehr als
- on -> sortierte Ausgabe
- Coluom Modifiers as Qualifiers (müssen in den global Qualifiers ganz am Ende stehen)
- :t -> zeigt nur den Hintern Teil der Ausgabe an (z.b. nur Dateinamen nicht den Pfad)
- :t:s/z/ZED/ -> zeigt nur den hinteren Teil der Ausgabe an und ersetzt in jeder Zeile das erste z mit ZED
- :t:gs/z/ZED/ -> zeigt nur den hinteren Teil der Ausgabe an und ersetzt in jeder Zeile alle z mit ZED

Misc

- zsh oder bash für Windows use Cygwin
- http://www.rayninfo.co.uk/tips/zshtips.html
- http://grml.org/zsh/zsh-lovers.html

CHAPTER 19

Network

Apache

Redirct all HTTP to HTTPS

```
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

- or

```
<VirtualHost *:80>
  RedirectMatch ^/(.*)$ https://$SERVER_HOST/$1
</VirtualHost>
```

New allow all

```
<Location "/content/">
    AllowOverride All
    Require all granted
</Location>
```

Run in foreground

```
LogLevel info
ErrorLog "|cat"
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog "|cat" common
```

Serve Django or WSGI app

- install mod_wsgi

```
<VirtualHost *:80>
    ServerName my-cool-webserver

    WSGIDaemonProcess $APPNAME user=$UNIX_USER group=$UNIX_GROUP threads=42
    WSGIScriptAlias / /path/to/run.wsgi

    <Directory /path/to/app/>
        WSGIProcessGroup $UNIX_GROUP
        WSGIApplicationGroup %{GLOBAL}
    </Directory>

</VirtualHost>
```

Bridging

Overview

- Bridging creates a virtual switch
- A bridge can handle STP (be aware of it!)
- Usually the bridge uses a physical interface like eth0 as uplink

Setup

- Install bridge-utils
- Add a new bridge interface

```
brctl addbr br0
```

- Add uplink

```
brctl addif br0 eth0
```

- Add other ports
- Show config

```
brctl show
```

- Remember to disable STP if you dont need it!

```
brctl stp br0 off
```

- Switch of network-manager
- Here how a bridge can be automatically configured with RHEL/CentOS/Fedora

```
DEVICE=br0
TYPE=Bridge
ONBOOT=yes
DELAY=0
BOOTPROTO=static
IPADDR=192.168.100.1
NETMASK=255.255.255.0
STP=off
```

- And how to add eth0 as uplink

```
BRIDGE=br0
NM_CONTROLLED=no
```

Firewalling

- To disable firewalling between bridge ports check */proc/sys/net/bridge/bridge-nf-call-**

GNS3

Getting a console on a router

- Drag and drop a router onto the working space
- Click on the play button (triangle)
- Right mouse click on the router -> console
- If the setup configruation after the first boot fails or freezes -> Dont panic!
- Delete the router
- Create a new one
- Skip first setup and configure it manually

Configure a Cisco Switch

- switches must be simulated as a router
- cisco catalyst switches must use platform c3600 or c3700
- right click on the router -> configure
- in the slots tab choose NM-16ESW on slot1
- you only be able to start some catalyst images in gns3 (see http://7200emu.hacki.at/viewtopic.php?t=6614)
- c3640-is-mz.123-14.T6.bin is known to be working as well

Configuring hosts

- Install a Qemu Linux (or whatever OS you desire) Image (or download one from http://wiki.qemu.org/Download#QEMU_disk_images)

- Goto Edit -> Preferences -> Qemu
- Check that the path of qemuwrapper.py is right
- Try to execute qemuwrapper.py
- If there are errors try to fix them
- On Arch linux you have to change python into python2 on the first line of the script
- Insert the path to your Qemu image in "Path to quemu-img"
- To test or manipulate Qemu images you can either mount it into your filesystem or start it with Qemu
- mount -n -o loop <path-to-qemu-image> /mnt
- qemu <path-to-qemu-image>
- for more information see http://en.wikibooks.org/wiki/QEMU/Images

Use your own Linux system as host os

- Download the ISO of your Linux distro as distro.iso
- Create a qemu harddisk
- qemu-img create linux.img 2G
- qemu -hda linux.img -cdrom distro.iso -boot -d -m 512

Links

- http://anakappa.blogspot.com - Lots of great video tutorials regarding GNS3

iproute

Simple stuff

- Set device up and give it an ip

```
ip l s <dev> up/down
ip a add <ip> <netmask> dev <dev>
ip a sh dev <dev>
```

- Remove one IP

```
ip a del <ip> dev <dev>
```

- Remove all ips

```
ip a flush dev eth0
```

- Show routing table

```
ip r
```

- Configure default gateway

```
ip route add default via 192.168.1.254
```

- Arp table

```
ip n
```

- Show interface statistics for packets and errors

```
ip -s l sh dev eth0
```

Change MAC

```
ip link set <dev> addr <mac>
```

Promisc mode

```
ip link set dev eth0 promisc on
```

Source routing

- Different default gateway depending on source address

```
ip route add $P1_NET dev $IF1 src $IP1 table T1
ip route add default via $P1 table T1
ip route add $P2_NET dev $IF2 src $IP2 table T2
ip route add default via $P2 table T2
```

Load balancing

```
ip route add default scope global nexthop via $P1 dev $IF1 weight 1 \
  nexthop via $P2 dev $IF2 weight 1
```

Show routes of ipsec tunnel

```
ip xfrm policy
ip xfrm state
```

Create a virtual interface

```
ip link add type veth
ip a add 1.2.3.4/24 dev veth0
```

A network interface with multiple mac addresses

```
ip link add link eth0 dev peth0 type macvlan address aa:aa:aa:aa:aa:aa
```

Network namespaces

- http://blog.scottlowe.org/2013/09/04/introducing-linux-network-namespaces/
- With network namespaces, you can have different and separate instances of network interfaces and routing tables that operate independent of each other.
- Only virtual network interfaces can be assigned to a network namespace and they always come in pairs connected peer-to-peer. One device for the default namespace to be connected to the physical interface by bridge and one to assign to the network namespace

```
ip netns add balle
ip netns list
ip link add veth0 type veth peer name veth1
ip link set veth1 netns balle
brctl addbr balle_br
brctl addif balle_br eth0 veth0
ip netns exec balle ip addr add 192.168.100.1/24 dev veth1
dhclient balle_br
```

- Now you can start a process or a shell if you like to use the new network namespace

```
ip netns exec balle bash
```

- Monitor namespaces

```
ip netns monitor
```

Irssi

Auto join

```
/SERVER ADD -auto -network Freenode irc.freenode.net
/NETWORK ADD -autosendcmd "/^nick yourname;/^msg nickserv identify passwordgoeshere;
↪wait 2000" Freenode
/CHANNEL ADD -auto #emacs Freenode
```

Shortcuts

- alt + 1-0 -> change window

Commands

- /names -> list users of channel
- /window new split

- /window shrink / grow <lines>
- /wc -> close window
- /go <nick/channel> - jump to the specified nick or channel

ICQ / Jabber integration

- install and configure bitlbee

Plugins

- bitlbee_join_notice.pl
- bitlbee_typing_notice.pl
- go.pl
- hilightwin.pl
- trackbar.pl
- urlgrab.pl
- notify - https://raw.github.com/lmacken/irssi-libnotify/master/notify.pl

LDAP

Overview

- Data imports are handled by ldif files
- Data structure is defined by schema files

Basic setup

- Edit /etc/openldap/slapd.conf
- suffix is the start of the directory tree (usually the server name)
- rootdn defines the admin user
- rootpw sets the admins password (generated with slappasswd)

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/inetorgperson.schema

suffix "dc=myserver,dc=mydomain,dc=tld"
rootdn "cn=admin,dc=myserver,dc=mydomain,dc=tld"
rootpw <password_hash>
```

Test config file

```
slaptest -u
```

Migrate passwd / group

- Maybe you need to install migrationtools

```
migrate_base.pl > base.ldif
migrate_group.pl /etc/group > group.ldif
migrate_passwd.pl /etc/passwd > passwd.ldif
```

- Edit ldif files and change `dc=padl,dc=com` to `dc=myserver,dc=mydomain,dc=tld`
- Maybe you have to delete the first entry of base.ldif because the root dn already exists

  ldapadd -x -D "cn=admin,dc=myserver,dc=mydomain,dc=tld" -W -f base.ldif ldapadd -x -D "cn=admin,dc=myserver,dc=mydomain,dc=tld" -W -f group.ldif ldapadd -x -D "cn=admin,dc=myserver,dc=mydomain,dc=tld" -W -f passwd.ldif

Dump database

```
ldapsearch -x -D "cn=admin,dc=myserver,dc=mydomain,dc=tld" -W -b "dc=myserver,
↪dc=mydomain,dc=tld"
```

List user / groups

```
getent passwd
getent group
```

Migrate from 2.3 to 2.4

- Use `/usr/lib/openldap/convert-config.sh` to convert old slapd.conf to new cn=config format

MDNS

Allgemeines

- Für MDNS unter Linux muss avahi installiert sein

Alle MDNS Dienste im Netz anzeigen

avahi-browse -arv

Alle MDNS fähigen Printer anzeigen

avahi-browse -rv _printer._tcp

Nagios

Define a service check for all hosts of a group except one

```
define service{
      service_description     CPU Stats
      servicegroups   sysres
      use             generic
      hostgroup_name  linux
      host_name       !server1
      check_command   check_iostat
}
```

Auto create a host and services configs by scanning ports with nmap

- Get Nmap2Nagios

```
nmap -sS -O -oX nmap.xml myserver.mydomain.com
nmap2nagios.pl -v -r nmap.xml -o new.cfg
```

Define OS of a host

```
define hostextinfo{
    host_name       server1
    icon_image      debian.png
    icon_image_alt  Debian
    vrml_image      debian.png
    statusmap_image debian.gd2
}
```

Check config for errors

```
nagios -v /etc/nagios/nagios.cfg
```

Convert timestamps of nagios.log

```
perl -pe 's/(\d+)/localtime(jumi)/e' nagios.log
```

Nagios statistical graphs

- http://www.pnp4nagios.org/

NetworkManager

List devices

```
nmcli d
```

List connections

```
nmcli c
```

Start / stop a connection

```
nmcli c up/down <connection>
```

Scan for wifi networks

```
nmcli d wifi list
```

Add a new ethernet connection

```
nmcli con add con-name my-eth1 ifname eth1 type ethernet ip4 192.168.100.100/24 gw4␣
↪192.168.100.1
```

Add a new wifi connection

```
nmcli con add con-name MyCafe ifname wlan0 type wifi ssid MyNet
nmcli con modify MyNet wifi-sec.key-mgmt wpa-psk
nmcli con modify MyNet wifi-sec.psk 'password'
```

Applet

```
nm-applet --sm-disable
```

Edit connections

```
nm-connection-editor
```

Nginx

Check config

```
nginx -t
```

Gzip compression

```
gzip  on;
gzip_comp_level 3;
gzip_proxied any;
# what mimetypes to gzip? otherwise only html gets zipped
gzip_types      text/plain text/css application/x-javascript text/xml application/xml␣
↪application/xml+rss text/javascript image/png image/gif image/jpeg image/x-icon␣
↪image/bmp;
gzip_disable "MSIE [1-6]\.";
gzip_min_length 1400; # in bytes
gzip_vary  on;        # allow the client to cache it
```

- If nginx is requested a foo.jpg and finds a foo.jpg.gz it will send that file instead of zipping foo.jpg

Keep-alive

```
# let several request be made in one connection
# hold connection open for max x seconds
keepalive_timeout 60;
```

Htaccess

```
auth_basic            "You shall not pass!";
auth_basic_user_file   /etc/nginx/htpasswd;
```

- Use htpasswd to create the file
- If you would like to require auth except for one ip

```
# allow puppet master to post reports
allow 1.2.3.4;

# require auth from all else
deny all;
satisfy  any;
auth_basic            "You must login";
auth_basic_user_file   /etc/nginx/htpasswd;""
```

SSL config

```
ssl on;
ssl_certificate /etc/nginx/mycert.pem;
ssl_certificate_key /etc/nginx/mycert.pem;

# avoid BEAST attack
ssl_ciphers RC4:HIGH:!aNULL:!MD5;
ssl_prefer_server_ciphers on;

# cache ssl sessions
ssl_session_cache    shared:SSL:10m;
ssl_session_timeout  10m;
```

Redirect

```
location / {
   return       301 https://www.ccc.de$request_uri;
}
```

Rewrite

```
location /old_stuff/ {
   rewrite   ^/old_stuff/(.*)$  /new_stuff/$1  permanent;
}
```

Security tricks

- Dont serve version control files, sql / json dumps

```
location ~ (\.git)|(CVS)|(\.svn)|(\.hg)|(\.ht)|(sql)|(dump)|(json) {
   access_log /var/log/nginx/security.log;
   return 404;
}
```

- Dont serve password files

```
location ~ (\.ht)|(pass) {
   access_log /var/log/nginx/security.log;
   return 404;
}
```

- Dont serve backup files

```
location ~ (\.old$)|(~$)|(^#)|(\.bak$)|(\.orig$)|(Kopievon)|(tmp) {
   access_log /var/log/nginx/security.log;
   return 404;
}
```

- Dont serve logs and docs

```
location ~ /(doc)|(log)|(documentation) {
   access_log /var/log/nginx/security.log;
```

```
   return 404;
}
```

- Dont serve dot files and dirs

```
location ~ /\. {
   access_log /var/log/nginx/security.log;
   return 404;
}
```

- Hide server version number

```
http {
   server_tokens off;
}
```

- Web Application Firewall: http://code.google.com/p/naxsi/

Load-Balancing

```
 upstream myservers {
   server 192.168.1.1;
   server 192.168.1.2;
 }

server {
   location / {
     proxy_pass http://myservers;
   }
 }
```

Traffic shaping

```
limit_rate_after 1g;
limit_rate       50k;
```

Request size

```
client_max_body_size 2M;
```

Debugging

```
# [ debug | info | notice | warn | error | crit ]
error_log  /var/log/nginx.error_log  debug
```

uWSGI Virtualhost for serving Django

```
server {
    listen  80;
    server_name .balle.de;
    root /srv/http/balle/balle;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    location /static {
        alias /srv/http/balle/static;
        gzip on;
        expires 30m;
    }

    location /media {
        gzip on;
        expires 24h;  # otherwise i client wont cache
    }

    location / {
        #uwsgi_pass 127.0.0.1:5050;
        uwsgi_pass unix:///var/run/uwsgi/balle.sock;
        include uwsgi.params;
    }
}
```

uwsgi.params

```
uwsgi_param  QUERY_STRING       $query_string;
uwsgi_param  REQUEST_METHOD     $request_method;
uwsgi_param  CONTENT_TYPE       $content_type;
uwsgi_param  CONTENT_LENGTH     $content_length;

uwsgi_param  REQUEST_URI        $request_uri;
uwsgi_param  PATH_INFO          $document_uri;
uwsgi_param  DOCUMENT_ROOT      $document_root;
uwsgi_param  SERVER_PROTOCOL    $server_protocol;

uwsgi_param  REMOTE_ADDR        $remote_addr;
uwsgi_param  REMOTE_PORT        $remote_port;
uwsgi_param  SERVER_PORT        $server_port;
uwsgi_param  SERVER_NAME        $server_name;
```

Gunicorn as WSGI server

- pip install gunicorn
- gunicorn -w 4 -D –bind unix:/tmp/gunicorn.sock myproject:app

```
server {
  location / {
    #proxy_pass http://localhost:8000;
    proxy_pass unix:/tmp/gunicorn.sock;
```

```
  }
}
```

PHP

- Install php-fpm and start server
- Add the following to your server directive

```
location ~ \.php$ {
   include fastcgi.conf;
   fastcgi_intercept_errors on;
   fastcgi_pass    unix:///var/run/php-fpm/php-fpm.sock;
}
```

Puppet Passenger

- For nginx with buildin passenger RPM see http://passenger.stealthymonkeys.com

```
user  nginx;
worker_processes  1;

pid        /var/run/nginx.pid;

events {
  worker_connections  1024;
}

http {
  include       /etc/nginx/mime.types;
  default_type  application/octet-stream;

  log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
  '$status $body_bytes_sent "$http_referer" '
  '"$http_user_agent" "$http_x_forwarded_for"';

  sendfile        on;
  tcp_nopush      on;

  keepalive_timeout  65;

  # Passenger needed for puppet
  passenger_root  /usr/share/rubygems/gems/1.8/passenger-3.0.21;
  passenger_ruby  /usr/bin/ruby;
  passenger_max_pool_size 15;

  ssl                  on;

  ssl_protocols  SSLv2 SSLv3 TLSv1;
  ssl_ciphers  ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;
  ssl_prefer_server_ciphers   on;


  server {
    listen                    8140 ssl;
```

```
    server_name                 puppetmaster.example.com;

    passenger_enabled           on;
    passenger_set_cgi_param     HTTP_X_CLIENT_DN $ssl_client_s_dn;
    passenger_set_cgi_param     HTTP_X_CLIENT_VERIFY $ssl_client_verify;

    access_log  /var/log/puppet/passenger_access.log  main;
    error_log  /var/log/puppet/passenger_error.log warn;

    root                        /usr/share/puppet/rack/puppetmasterd/public/;

    ssl_certificate      /var/lib/puppet/ssl/certs/puppetmaster.example.com.pem;
    ssl_certificate_key  /var/lib/puppet/ssl/private_keys/puppetmaster.example.com.
↪pem;
    ssl_crl                     /var/lib/puppet/ssl/ca/ca_crl.pem;
    ssl_client_certificate      /var/lib/puppet/ssl/certs/ca.pem;
    ssl_ciphers                 SSLv2:-LOW:-EXPORT:RC4+RSA;
    ssl_prefer_server_ciphers   on;
    ssl_verify_client           optional;
    ssl_verify_depth            1;
    ssl_session_cache           shared:SSL:128m;
    ssl_session_timeout         5m;
  }
}
```

Open vSwitch

Overview

- Consists of a daemon and a database server that stores the switch config in json
- You can use a kernel daemon for better performance
- A port is a bridge

Create a new port

```
ovs-vsctl add-br br0
ovs-vsctl add-port br0 eth0
ovs-vsctl show
```

Connect a vm to a port

- Use *virsh edit* <*vm*> to update network config and set

```
<interface type='bridge'>
  <mac address='52:54:00:71:b1:b6'/>
  <source bridge='br0'/>
  <virtualport type='openvswitch'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

Limit an interface to 1 MBit

```
ovs-vsctl set Interface tap0 ingress_policing_rate=1000
ovs-vsctl set Interface tap0 ingress_policing_burst=100
```

Set a port into a VLAN

```
ovs-vsctl set port <port name> tag=<VLAN ID>
```

Bonding

```
ovs-vsctl add-br ovsbr1
ovs-vsctl add-bond ovsbr1 bond0 eth1 eth3
```

Get information

- About the switch overall

```
ovs-vsctl show
```

- A port

```
ovs-vsctl list port
```

- An interface

```
ovs-vsctl list interface
```

Postfix

Masquerade Domain

```
masquerade_domains = foo.example.com
```

Rewrite an email address

- /etc/postfix/main.cf

```
relocated_maps = hash:/etc/postfix/relocated
```

- /etc/postfix/relocated

```
username@example.com      otheruser@elsewhere.tld
```

Delete all mails in queue

```
for i in `mailq|grep '@' |awk {'print $1'}|grep -v '@'`; do postsuper -d $i ; done
```

PXE

Installation

- DHCP server
- TFTP server
- Syslinux

Setup DHCP

- next-server is the ip address of the tftp server
- filename points to syslinux image

```
option domain-name-servers 8.8.8.8;
default-lease-time 86400;
max-lease-time 604800;
log-facility local7;
authoritative;

subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.100 192.168.1.150;
  filename "pxelinux.0";         # the PXELinux boot agent
  nextserver 192.168.1.23;
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.1.255;
  option routers 192.168.1.1;
}
```

Setup TFTP and Syslinux

```
cp /usr/lib/syslinux/pxelinux.0 /var/tftpboot/
cp /usr/lib/syslinux/menu.c32 /var/tftpboot/
cp /usr/lib/syslinux/memdisk /var/tftpboot
cp /usr/lib/syslinux/mboot.c32 /var/tftpboot/
cp /usr/lib/syslinux/chain.c32 /var/tftpboot
mkdir /var/tftpboot/pxelinux.cfg
mkdir -p /var/tftpboot/images/CentOS
```

- Download ISO image (e.g. CentOS)
- http://mirror.switch.ch/ftp/mirror/centos/6.3/isos/i386/CentOS-6.3-i386-minimal.iso
- Copy kernel and ramdisk from ISO image

```
mount -o loop CentOS-6.3-i386-bin-DVD1.iso /mnt/
cp /mnt/images/pxeboot/* /var/tftpboot/images/CentOS
umount /mnt
```

- Create file /var/tftpboot/pxelinux.cfg/default

```
DEFAULT menu.c32
PROMPT 0
MENU TITLE Balle sein PXE Main Menu

console 0
serial 0 115200 0

LABEL CentOS
  MENU LABEL ^CentOS
  KERNEL images/CentOS/vmlinuz
  APPEND initrd=images/CentOS/initrd.img ksdevice=eth0 console=ttyS0,115200␣
↪earlyprint=serial,ttyS0,115200 ramdisk_size=9025
```

Optionally kickstart setup

- Install a webserver (e.g. nginx)
- Create kickstart file (/srv/http/web/centos-kickstart.cfg)

```
install
url --url http://mirror.centos.org/centos/5/os/i386
lang de_DE.UTF-8
keyboard de
network --device eth0 --bootproto dhcp
rootpw test123
firewall --enabled --ssh
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone --utc Europe/Zurich
bootloader --location=mbr
reboot

# Partitioning
clearpart --all --drives=sda
part /boot --fstype ext3 --size=100 --ondisk=sda
part pv.2 --size=0 --grow --ondisk=sda
volgroup VolGroup00 --pesize=32768 pv.2
logvol / --fstype ext3 --name=LogVol00 --vgname=VolGroup00 --size=1024 --grow
logvol swap --fstype swap --name=LogVol01 --vgname=VolGroup00 --size=256 --grow --
↪maxsize=512

%packages
@core
openssh
openssh-clients
openssh-server
```

- Edit /var/tftpboot/pxelinux.cfg/default and append to the APPEND line

```
ks=http://<ip_of_pxe_server>/centos-kickstart.cfg
```

Install clonezilla images

- Unzip clonezilla zip into `/var/lib/tftpboot/images/clonezilla`
- Edit `/var/tftpboot/pxelinux.cfg/default` and for manual installtion append

```
LABEL Clonezilla
    MENU LABEL Clonezilla
    APPEND initrd=clonezilla/live/initrd.img boot=live config noswap nolocales edd=on␣
↪nomodeset ocs_live_run="ocs-live-general" ocs_live_extra_param="" keyboard-layouts="
↪" ocs_live_batch="no" locales="" vga=788 nosplash noprompt fetch=tftp://<ip_of_pxe_
↪server>/pxe/clonezilla/live/filesystem.squashfs
    KERNEL clonezilla/live/vmlinuz
```

- for full-automatic installion append

```
LABEL Clonezilla
    MENU LABEL Clonezilla
    APPEND initrd=images/clonezilla/live/initrd.img boot=live config noswap nolocales␣
↪edd=on nomodeset ocs_live_run="/usr/sbin/ocs-sr --batch -q -e1 auto -e2 -r -j2 -p␣
↪reboot restoredisk $IMAGENAME sda" ocs_live_extra_param="" ocs_live_keymap="NONE"␣
↪ocs_live_batch="yes" ocs_lang="en_US.UTF-8" vga=788 nosplash noprompt ocs_prerun=
↪"mount -t nfs -o vers=3 <ip_of_pxe_server>:/local/clonezilla /home/partimag" ocs_
↪postrun="$POST_COMMAND && reboot" fetch=tftp://<ip_of_pxe_server>/images/clonezilla/
↪live/filesystem.squashfs
    KERNEL clonezilla/live/vmlinuz
```

- Make sure to replace $POST_COMMAND, $IMAGENAME, /path/to/image and the ip of your pxe server
- Virtio disks can also be a problem make sure to use normal disks in vms

Diskless Redhat

- Install dracut and dracut-network
- Edit `/etc/dracut.conf` and make sure the following is set

```
add_dracutmodules+="nfs"
```

- Also add network kernel modules to `add_drivers+`
- Build initramdisk

```
dracut initramfs-3.10.0-327.4.5.el7.x86_64 3.10.0-327.4.5.el7.x86_64
lsinitrd initramfs-3.10.0-327.4.5.el7.x86_64
```

- Create minimal root filesystem

```
yum groupinstall Base --installroot=/export/diskless-el7
yum install openssh-server openssh-clients --installroot=/export/diskless-el7
```

- Make sure there is a network config file for your card e.g. `/export/diskless-el7/etc/sysconfig/network-scripts/ifcfg-eno0`

```
DEVICE=eno0
NAME=eno0
BOOTPROTO=dhcp
ONBOOT=yes
```

- Create `/export/diskless-el7/etc/fstab`

```
none            /tmp            tmpfs   defaults        0 0
tmpfs           /dev/shm        tmpfs   defaults        0 0
sysfs           /sys            sysfs   defaults        0 0
proc            /proc           proc    defaults        0 0
none            /var/log        tmpfs   defaults        0 0
none            /var/run        tmpfs   defaults        0 0
none            /var/lock       tmpfs   defaults        0 0
none            /var/tmp        tmpfs   defaults        0 0
none            /var/spool      tmpfs   defaults        0 0
```

- Set default target to multi-user instead of graphical

```
rm -f /export/diskless-el7/etc/systemd/system/default.target
ln -s ../../../usr/lib/systemd/system/multi-user.target /export/diskless-el7/etc/
↪systemd/system/default.target
```

- Create some links to files or dirs that otherwise want to be writable

```
ln -s etc/ld.so.cache~ tmp/ld.so.cache~
ln -s etc/udev/hwdb.bin tmp/hwdb.bin
mkdir tmp/catalog; ln -s var/lib/systemd/catalog tmp/catalog
ln -s etc/machine-id /tmp/machine-id
```

- Generate ssh host keys for the image

```
mount -o bind /dev /export/diskless-el7/dev
chroot /export/diskless-el7
ssh-keygen -A
exit
umount /export/diskless-el7/dev
```

- Copy your public key into the image

```
mkdir /export/diskless-el7/root/.ssh
cat ~/.ssh/id_rsa.pub > /export/diskless-el7/root/.ssh/authorized_keys
```

- Delete unneeded services

```
cd /export/diskless-el7/etc/systemd/multi-user.target.wants
rm -f abrt* atd.service crond.service chronyd.service firewalld.service kdump.service␣
↪libstoragemgmt.service rhel-dmesg.service rhsmcertd.service rngd.service sysstat.
↪service
```

- Edit `/var/tftpboot/pxelinux.cfg/default` and add

```
LABEL  rhel7
   KERNEL diskless/vmlinuz-3.10.0-327.4.5.el7.x86_64
   APPEND initrd=diskless/initramfs-3.10.0-327.4.5.el7.x86_64 root=nfs:<ip_to_nfs_
↪server>:/export/diskless-el7:nfsvers=3 ip=dhcp netdev=boot
```

Fedora netinstall

- Download lkrn from https://boot.fedoraproject.org/download
- Edit `/var/tftpboot/pxelinux.cfg/default` and add

```
LABEL  Fedora
MENU LABEL boot.fedoraproject.org
IPAPPEND 2
KERNEL linux/bfo.lkrn
```

SMTP

SMTP (Simple Mail Transfer Protokoll)

Um die Kommunikation mit einem Mailserver zu testen können folgende Befehle in einer telnet Sitzung benutzt werden.

```
telnet mail.server.de 25

helo <rechnername>
auth login:
username (base64 -> printf 'username' | mimencode)
password (base64 -> printf 'password' | mimencode)
mail from: <mein@mail.address>
rcpt to: <empfänger@mail.address>
Data
To: mein Name <mein@mail.address>
From: Empfänger Name <empfänger@mail.address>
Subject: Hallo
Dies ist eine telnet generierte Mail.
.
```

Eine neue Zeile mit einem Punkt beendet die Dateneingabe und versendet die Mail. Jetzt kann überprüft werden ob die Mail beim Empfänger ankommt.

SNMP

get all info

```
snmpwalk -Os -c public -v 1 $host .
```

print numeric OIDs

```
snmpwalk -On -c public -v 1 $host .
```

set a special info

```
snmpset -c public -v 1 192.168.1.1 ipDefaultTTL.0 i 66
```

Exceute command via snmp (in snmpd.conf)

```
exec muh /some/command
```

Install a new MIB file

- Copy the file (e.g. MY-MIB.txt) to /usr/share/snmp/mibs

```
snmpwalk -Of -v 1 -c public -m +MY-MIB 192.168.1.1 .
```

Translate a MIB to its number

- maybe you need to delete the last .0

```
snmptranslate -m ALL -On <mib>
```

Scripting with Perl

```
#!/usr/bin/perl

use strict;
use Net::SNMP;
use Time::HiRes qw(usleep nanosleep);

my $snmp_host = "127.0.0.1";
my $snmp_version = 1;
my $snmp_community = "public";

my $oid = "";

my ($snmp, $error) = Net::SNMP->session(-hostname => $snmp_host,
                                        -version => $snmp_version,
                                        -community => $snmp_community);

die "SNMP connect to $snmp_host failed: $error\n" if $error;

while (1)
{
  my $results = $snmp->get_request(-varbindlist => [ $oid_a_total, $oid_w_total, $oid_
↪kwh_total, $oid_pf_total ]);
  print join("", values(%$results)) . "\n";

  # Request every millisecond
  usleep(1);

  # Request every nanosecond
  #nanosleep(1);
}

$snmp->close();
```

Socket

Closing a zombie socket

```
fuser -k -n tcp 80
```

SSH

Local port forward

- Local port 5555 gets forwarded to 25 on remote machine over ssh-host

```
ssh -Nf -L 5555:remote-machine:25 user@ssh-host
```

Socks forward

```
ssh -D 5555 user@remote-machine
```

- Now connect your firefox to socks proxy port 5555 and it will tunnel everything over it

Ignore Host key

```
ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
```

UPNP

Overview

- Broadcast via 239.255.255.250 UDP port 1900
- Uses SOAP to exchange Information
- Method M-SEARCH to search for other devices
- Method NOTIFY to periodically send information
- GET /ctl/ContentDir returns root dir of a media / file service
- ISG protocol can be used to configure port forwards / nat
- http://www.ethicalhacker.net/content/view/220/1/

Search for devices

```
from scapy.all import IP, UDP, send, sniff
payload = "M-SEARCH * HTTP/1.1\r\n" \
"HOST:239.255.255.250:1900\r\n" \
"ST:upnp:rootdevice\r\n" \
"MAN: \"ssdp:discover\"\r\n" \
"MX:2\r\n\r\n"

send(IP(dst="239.255.255.250") / UDP(sport=1900, dport= 1900) / payload)
sniff(filter="udp and port 1900", prn=lambda p: p["IP"].src + " " + str(p["UDP"].
↪payload))
```

Search for services

```
from scapy.all import IP, UDP, send, sniff
ip = "192.168.1.1"
payload = "M-SEARCH * HTTP/1.1\r\nHost:%s:1900\r\nST: ssdp:all\r\nMan:\"ssdp:discover\
↪"\r\n" % ip
send(IP(dst=ip) / UDP(sport=1900, dport= 1900) / payload)
sniff(filter="udp and port 1900", timeout=10, prn=found_service)
```

Portforward

- untested

```
from scapy.all import IP, UDP, sr1
target="192.168.1.1"
payload = "POST /upnp/control/igd/wanpppcInternet HTTP/1.1\r\n"
payload += "Host: %s\r\n\r\n" % target
payload += """<?xml version="1.0" ?>
<s:Envelope s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:s=
"http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
      <u:AddPortMapping xmlns:u="urn:schemas-upnp-org:service:WANIPConnection:1">
          <NewRemoteHost/>
          <NewExternalPort>2200</NewExternalPort>
          <NewProtocol>TCP</NewProtocol>
          <NewInternalPort>22</NewInternalPort>
          <NewInternalClient>192.168.1.2</NewInternalClient>
          <NewEnabled>1</NewEnabled>
          <NewPortMappingDescription>SSH Tunnel</NewPortMappingDescription>
          <NewLeaseDuration>0</NewLeaseDuration>
      </u:AddPortMapping>
  </s:Body>
</s:Envelope>"""
r = sr1(IP(dst=target) / UDP(dport= 1900) / payload);
r.display();
```

Tools & papers

- https://community.rapid7.com/community/infosec/blog/2013/01/29/security-flaws-in-universal-plug-and-play-unplug-dont-play
- http://finux.co.uk/slides/UPnP-Revisted-BSidesLondon-2012-finux.pdf

- http://code.google.com/p/mirandaupnptool/
- http://www.gnucitizen.org/blog/hacking-the-interwebs/

UWSGI

uwsgi.xml

```
<uwsgi>
  <!-- <socket>127.0.0.1:5050</socket> -->
  <socket>/var/run/uwsgi/balle.sock</socket>
  <chown-socket>www-data</chown-socket>
  <uid>www-data</uid>
  <gid>www-data</gid>
  <master/>
  <!-- <daemonize>/var/log/uwsgi/balle.log</daemonize> -->
  <max-requests>5000</max-requests>

  <!-- max seconds an app will after its being killed -->
  <harakiri>120</harakiri>

  <!-- restart the server after x seconds of inactivity -->
  <inactivity>300</inactivity>

  <!-- name of wsgi app file -->
  <module>django_wsgi</module>

  <!-- reload server after touching that file -->
  <touch-reload>/srv/http/balle/uwsgi/django_wsgi.py</touch-reload>

  <!-- setup python environment -->
  <autoload/>
  <plugins>python</plugins>
  <pythonpath>/srv/http/balle/</pythonpath>
  <pythonpath>/srv/http/balle/uwsgi</pythonpath>
  <pythonpath>/srv/virtualenvs/balle/lib/python2.7/site-packages</pythonpath>

  <!-- use virtualenv -->
  <pyhome>/srv/virtualenvs/balle/</pyhome>


  <!-- Number of processes. Set this to number of cpus not cores -->
  <processes>1</processes>

  <!-- number of threads. Set this to the total number of cores -->
  <enable-threads/>
  <threads>4</threads>
</uwsgi>
```

django_wsgi.py

```
import os
import django.core.handlers.wsgi
```

```
os.environ['DJANGO_SETTINGS_MODULE'] = 'balle.settings'
application = django.core.handlers.wsgi.WSGIHandler()
```

start uWSGI process

```
uwsgi -x uwsgi.xml
```

RFC Sourcebook

CHAPTER 20

Security

Cryptography

- Why using a hash algo multiple times is a bad idea

Cisco

Allgemeines

- com2usb linux driver pl2303.ko
- running config mit passwörtern anzeigen

```
more system:running-config
```

- man kann nur von grösseren security level auf kleinere routen
- mit enable (oder en) kommt man in den enable modus
- mit conf t in den configurations modus
- befehle kann man mit tab vervollständigen
- mögliche optionen erfährt man mit ?
- mit “no” vor einem befehl löscht man eine config zeile
- config speichern mit “copy running-config startup-config”
- system auslastung anzeigen

```
 show cpu usage
 show processes
 show mem


Grundconfiguration
```

- hostname lalala
- domain-name schwubs.tld
- Zeitzone und NTP einstellen

```
clock timezone CET 1
ntp server 192.168.100.1
sh clock
```

- user anlegen

```
username <user> password <password> privilege 15
```

- enable passwort setzen

```
aaa authentication enable console LOCAL
```

- SSH und Consolen User lokal authentifiziert

```
aaa authentication ssh console LOCAL
```

- sh run interface
- interface resetten

```
interface Management0/0
no nameif
no security-level 100
no ip address
no management-only
```

- interface config anschauen

```
sh run interface Management0/0
interface Management0/0
no nameif
no security-level
no ip address

interface Ethernet0/0
description WAN interface
ip address 192.168.103.91 255.255.255.0 standby 192.168.103.92
security-level 0
nameif external
no shut
```

- SSH Config

```
ssh 192.168.103.0 255.255.255.0 external
crypto key generate rsa general-keys modulus 2048
write mem
```

Routing

- Default Gateway setzen (device ip mask gateway)

```
route external 0.0.0.0 0.0.0.0 192.168.103.254
```

- Routing Tabelle anzeigen

```
sh route
```

- source validation anschalten

```
 ip verify reverse-path interface <iface>


VLAN
```

- trunk port = port der in mehreren vlans hängt

```
interface Management0/0.1
description LAN Failover Interface
vlan 8
interface Management0/0.2
description STATE Failover Interface
vlan 9
```

- wenn man kein vlan angibt, dann sind bei einem trunk port per default alle erlaubt
- VLAN config anschauen

```
 sh vlan


ARP
```

- ARP Cache anzeigen lassen

```
 sh arp



NAT
```

- nat-control (alle connections müssen eine nat rule haben)
- alles was aus external raus geht nimm per default die adresse vom external device

```
global (external) 1 interface
```

- alles was aus interface patronas raus kommt natte mit id 1 also auf das externe interface

```
nat (patronas) 1 192.168.109.176 255.255.255.240
```

- Alles was nat id 0 hat wird nicht genattet

```
access-list NO_NAT deny ip any any
nat (patronas) 0 access-list NO_NAT
```

- Statisches NAT (dev intern, dev extern) (192.168.109.215 wird genattet auf 192.168.103.93)

```
 static (axxion,external) 192.168.103.93 192.168.109.215 netmask 255.255.255.255


Logging und Debugging
```

```
logging enable
logging console ?
logging console 6
```

- wenn man via ssh connected is nimmt man monitor und nicht console

```
 logging monitor 7
 term monitor


Packet Filtering / Access lists
```

- Access list anlegen

```
access-list EXTERNAL_IN permit icmp any any source-quench
access-list EXTERNAL_IN permit icmp any any unreachable
access-list EXTERNAL_IN permit tcp any host 192.168.109.215 eq 22
```

- Access liste an ein interface binden

```
access-group EXTERNAL_IN in interface external
```

- bei genatteten verbindung brauch man nur die nat ip erlauben das weiterleiten wird dann automatisch erlaubt

Packet Capturing

- Alles was durch die Access-List gelassen wird, wird aufgezeichnet

```
access-list CAP permit ip any any
capture CAP interface patronas access-list CAP
```

- Aufgezeichnete Pakete anzeigen

```
sh capture CAP
```

- Aufzeichnen stoppen

```
no capture CAP
```

- Allen Traffic auf einem Interface capturen

```
 access-list CAP interface external


Failover
```

- Die Failover IP für LAN muss in einem anderen Netz sein als das für State

```
failover
failover lan unit primary
failover lan interface lan-fo Management0/0.1
failover key Fmjhd3
failover replication http
failover link state-fo Management0/0.2
failover interface ip lan-fo 192.168.109.193 255.255.255.252 standby 192.168.109.194
failover interface ip state-fo 192.168.109.197 255.255.255.252 standby 192.168.109.198
```

- sh run failover (config anzeigen)
- sh failover (status anzeigen)
- monitoring bei logischem device anschalten

```
monitor interface patronas
```

- Die Slave Firewall von der Master aus rebooten

```
 failover reload-standby


Firewall Desaster Recovery
```

- Es muss sichergestellt sein, dass Ethernet 0/2 auf beiden ASA das aktive Interface ist

```
 interface Redundant1
 redundant-interface Redundant 1 active-member Ethernet 0/2


IPSec / VPN
```

- crypto isakmp enable <interface>
- crypto isakmp identity address
- Phase 1 (control connection definieren) * sh crypto isakmp sa detail

```
crypto isakmp policy 1
authentication pre-share
encryption 3des
hash md5
group 5
lifetime 3600
```

- Phase 2 (data connection definieren) * sh crypto ipsec sa peer <$VPN_PEER> * Transform Set definieren (Name für Verschlüsselung / Hashing Optionen für die wirklichen Datentunnel)

```
crypto ipsec transform-set ESP-AES-256-MD5 esp-aes-256 esp-md5-hmac
crypto ipsec transform-set ESP-3DES-MD5 esp-3des esp-md5-hmac

* Optional maximale Timeouts für das Rekeying definieren
```

```
crypto ipsec security-association lifetime seconds 28800
crypto ipsec security-association lifetime kilobytes 4608000

* Name für VPN Peer anlegen
```

```
name 213.23.72.194 VPN_PEER_TEST

* Welcher Traffic getunnelt werden soll, wird über eine Accesslist (CMAP_$VPN_MAP)␣
↪definiert
* Die Src muss immer dem Netz der Firewall entsprechen
```

```
access-list CMAP_TEST_MATCH extended permit ip 192.168.109.176 255.255.255.240 192.
↪168.103.0 255.255.255.0
access-list CMAP_TEST_MATCH extended permit ip 192.168.109.176 255.255.255.240 192.
↪168.100.0 255.255.255.0
access-list CMAP_TEST_MATCH extended permit ip 192.168.109.208 255.255.255.240 192.
↪168.103.0 255.255.255.0
access-list CMAP_TEST_MATCH extended permit ip 192.168.109.208 255.255.255.240 192.
↪168.100.0 255.255.255.0
access-list CMAP_TEST_MATCH extended permit ip host 123.123.122.66 host 192.168.100.3
access-list CMAP_TEST_MATCH extended permit ip host 123.123.122.66 host 192.168.100.1
access-list CMAP_TEST_MATCH extended permit ip 192.168.109.224 255.255.255.240 192.
↪168.100.0 255.255.255.0
access-list CMAP_TEST_MATCH extended permit ip 192.168.109.224 255.255.255.240 192.
↪168.103.0 255.255.255.0

* WICHTIG! Genau die selben Regeln müssen auch in die NO_NAT Access-List eingetragen␣
↪werden

* Eine Cryptomap ist eine Sammlung von Phase2 gebunden an ein Interface
* Identifiziert wird über die Zahl z.B. 20
```

```
crypto map CMAP_STATIC 20 match address CMAP_TEST_MATCH
crypto map CMAP_STATIC 20 set peer VPN_PEER_TEST
crypto map CMAP_STATIC 20 set transform-set ESP-AES-256-MD5

* Optional Perfect Forwarding Secrecy (PFS) einschalten
  * Erzwingt das ein neuer Schlüssel beim Rekeying generiert wird
```

```
crypto map CMAP_STATIC 20 set pfs
```

```
  * Macht immens Probleme zwischen unterschiedlichen Peers
```

- PreShared Key vergeben
- Hier muss immer die IP verwendet werden
- l2l heisst LAN-to-LAN

```
tunnel-group 123.123.11.22 type ipsec-l2l
tunnel-group 123.123.11.22 general-attributes
tunnel-group 123.123.11.22 ipsec-attributes
pre-shared-key <password>
```

- Beispiel für einen neuen Tunnel

```
name 123.124.222.5 VPN_PEER_BLA

access-list CMAP_BLA_MATCH extended permit ip host 192.168.109.213 host 123.222.147.10
access-list NO_NAT extended permit ip host 192.168.109.213 host 123.222.147.10

crypto map CMAP_STATIC 40 match address CMAP_BLA_MATCH
crypto map CMAP_STATIC 40 set pfs
crypto map CMAP_STATIC 40 set peer VPN_PEER_BLA
crypto map CMAP_STATIC 40 set transform-set ESP-3DES-MD5
crypto map CMAP_STATIC 40 set security-association lifetime seconds 28800
crypto map CMAP_STATIC 40 set security-association lifetime kilobytes 4608000

tunnel-group 193.228.147.5 type ipsec-l2l
tunnel-group 193.228.147.5 general-attributes
default-group-policy VPN
tunnel-group 193.228.147.5 ipsec-attributes
pre-shared-key <password>
```

- PPPOE configuration

```
 interface Vlan2
 nameif outside
 security-level 0
 pppoe client vpdn group QSC
 ip address pppoe setroute

 vpdn group ISP request dialout pppoe
 vpdn group ISP localname [your username here]
 vpdn group ISP ppp authentication chap
 vpdn username [your username here] password [your password here ]



Switch Config
```

- ip domain-name patronas.int
- user spass

```
aaa new-model
username <user> password 0 <pass>
```

- ssh server für 15 terminals freischalten

```
line vty 0 15
transport input ssh
```

- switch ports sind standardmaessig in vlan1
- trunk port configurieren

```
switchport mode trunk
switchport trunk allowed vlan 8,9
```

- einen port in ein vlan hängen

```
switchport access vlan 10
```

- eine andere art vlan zu confen (im enable mode)

```
vlan database
vlan <nr> name <name>
int vlan <nr>
ip add x.x.x.x
```

- default gateway einstellen

```
ip default-gateway 192.168.1.1
```

- lacp config

```
 switchport mode access
 channel-protocol lacp
 channel-group 1 mode active


Firmware update on a Cisco device
```

- Setup a TFTP server in the same IP range as the Cisco device to backup

the configs, IOS image and also for later to upload the new IOS image.

```
testrouter# copy startup-config tftp
Address or name of remote host []? 10.10.10.2
Destination filename [startup-config]?
!!
1278 bytes c opied in 0.100 secs
```

- Backup Current IOS Image

```
testrouter# copy flash: tftp:
Source filename []? xxxxx-xx-xx.121-x.XB
Address or name of remote host []? 10.10.10.2
Destination filename [xxxxx-xx-xx.121-x.XB]?
```

- Now, Load the new IOS image from the TFTP onto the flash

```
 ciscorouter#copy tftp: flash:
 Address or name of remote host []? 10.10.10.2
 Source filename []? c3560-ipbasek9-mz.122-40.SE.bin
 Destination filename [c3560-ipbasek9-mz.122-40.SE.bin]?
```

```
 Accessing tftp://10.10.10.2/c3560-ipbasek9-mz.122-40.SE.bin
 Loading c3560-ipbasek9-mz.122-40.SE.bin



How to repair a Cisco with erased flash
```

- copy xmodem: flash:flash_filename
- now from the "transfer" dropdown menu on the hyperterminal,

select "send file" and choose "xmodem" in the subsequent dialog box and browse for the flash_filename (the downloaded IOS bn file) and send.

- boot flash:flash_filename

Firewalld

Allow a port

```
firewall-cmd --add-port 5405/udp
```

Grsecurity

Activate RBAC

```
gradm -P
gradm -P admin
gradm -P shutdown
gradm -E
```

Learn policy for special program

- Add the following to `/etc/grsec/policy`

```
subject /path/of/binary ol
    / h
    -CAP_ALL
    connect disabled
    bind disabled
```

- Start learning mode

```
gradm -L /etc/grsec/learning.logs -E
```

- Switch to admin role and dump learning.log to policy.new

```
gradm -a admin
gradm -L /etc/grsec/learning.logs -O /etc/grsec/policy.new
```

- Review policy.new, add it to policy and reload it

```
gradm -R
```

- Leave admin role

```
gradm -u
```

Explanation of PaX flags

Flax	Description
PAX_NOEXEC	This option enables the protection of allocated pages of memory as non-executable if they are not part of the text segment of the running process. It is needed for PAGEEXEC, SEGMEXEC and KERNEXEC.
PAGE-EXEC	The kernel will protect non-executable pages based on the paging feature of the CPU. This is sometimes called "marking pages with the NX bit" in other OSes. This feature can be controlled on a per ELF object basis by the PaX P and p flags.
SEG-MEXEC	This is like PAGEEXEC, but based on the segmentation feature of the CPU and it is controlled by the PaX S and s flags. Note that SEGMEXEC is only available on CPUs that support memory segmentation, namely x86.
EMU-TRAMP	The kernel will emulate trampolines (snippets of executable code written on the fly) for processes that need them, e.g. nested functions in C and some JIT compilers. Since trampolines try to execute code written by the process itself to memory marked as non-executable by PAGEEXEC or SEGMEXEC, the PaX kernel would kill any process that tries to make use of one. EMUTRAMP allows these processes to run without having to fully disable enforcement of non-executable memory. This feature can be controlled on a per ELF object basis by PaX E and e flag.
MPRO-TECT	The kernel will prevent the introduction of new executable pages into the running process by various techniques: it will forbid the changing of the executable status
RAND-MMAP	The kernel will use a randomized base address for mmap() requests that do not specify one via the MAP_FIXED flag. It is controlled by the PaX R and r flags.

Honeypot

Overview

- http://www.honeynet.org/project

Simulate broken webapp

- http://glastopf.org/

Simulate network services

- http://dionaea.carnivore.it/
- Simulates somes services mainly SMB with known attacks and tries to catch new shell code / exploit techniques with libemu
- http://honeytrap.carnivore.it/

- Sniff for connection requests, start a server and catch everything, if the client shut ups try to communicate to catch the whole exploit.

Simulate broken client

- https://github.com/buffer/thug

Create an attack map

- https://github.com/fw42/honeymap

IPtables

Example script

```
#!/bin/bash

###[ Config ]###

LOGLIMIT=20
IPTABLES=/usr/sbin/iptables
IP6TABLES=/usr/sbin/ip6tables


###[ CLEANUP RULE ]###

# Erstmal alle Rules loeschen...
echo "Deleting old rules"
$IPTABLES -F
$IPTABLES -t nat -F
$IP6TABLES -F


###[ CREATING NEW CHAINS ]###

echo "Creating chains"

# Chain to log and reject a port by ICMP port unreachable
$IPTABLES -N LOGREJECT
$IPTABLES -A LOGREJECT -m limit --limit $LOGLIMIT/minute -j LOG --log-prefix
↪"FIREWALL REJECT " --log-level notice --log-ip-options --log-tcp-options
$IPTABLES -A LOGREJECT -j REJECT --reject-with icmp-port-unreachable


###[ PROC MANIPULATION ]###

# Enable IP Forwarding
#echo "Enabling IP forwarding"
#echo 1 > /proc/sys/net/ipv4/ip_forward

# Dont respond to broadcast pings
echo "Disabling broadcast pings"
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

```
# Halt die Klappe bei komischen ICMP Nachrichten
echo "Enabling bogus ICMP message protection"
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

# Enable SYN Flood protection
echo "Enabling SYN FL00D protection"
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

# Kick all IP Spoofing shit
# (Enable source validation)
echo "Disabling IP Spoofing attacks"
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# Logge seltsame Pakete
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians

# Set default TTL to 61 (default for Linux is 64)
echo "Setting default TTL to 61"
echo 61 > /proc/sys/net/ipv4/ip_default_ttl

# Sende RST Pakete raus, wenn der Buffer voll ist
echo 1 > /proc/sys/net/ipv4/tcp_abort_on_overflow

# Warte max. 30 Sekunden auf ein FIN/ACK.
# Schliesse danach den Socket
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout

# Gib nach 3 SYN/ACK Paketen den Verbindungsaufbau auf
# Default ist 6
echo 3 > /proc/sys/net/ipv4/tcp_synack_retries


###[ MAIN PART ]###

# Set default policy
echo "Setting default policy DROP"
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT ACCEPT
$IP6TABLES -P INPUT DROP
$IP6TABLES -P FORWARD DROP
$IP6TABLES -P OUTPUT ACCEPT

# Be stateful
echo "Be stateful"
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# In the loopback device we trust all other we monitor ;)
echo "Trust loopback"
$IPTABLES -A INPUT -i lo -j ACCEPT
$IP6TABLES -A INPUT -i lo -j ACCEPT

$IPTABLES -A INPUT -i tap0 -j ACCEPT

# ICMP is ok
```

```
echo "ICMP"
$IPTABLES -A INPUT -p icmp -j ACCEPT

# Erlaube SSH Logins
echo "SSH"
$IPTABLES -A INPUT -p tcp --dport 22 -j ACCEPT

# Verbindungsversuche loggen und rejecten
# Der Rest wird eh per Default Policy gedroppt
echo "Reject and log all other packets"
$IPTABLES -A INPUT -p tcp --syn -j LOGREJECT
$IPTABLES -A FORWARD -p tcp --syn -j LOGREJECT
```

Showing rules

- Show all rules with interfaces

```
iptables -L -n -v
```

- Show all NAT rules

```
iptables -L -t nat -n -v
```

OpenBSD

What's missing (in mid 2016)?

- Bluetooth
- TRIM support for SSDs
- Adobe Flash

Filesystem tweaks

- Configure soft updates everywhere (softdep)
- Disable access time logging (noatime)
- If possible mount with noexec, nosuid, nodev

```
<duid> /home ffs rw,nodev,nosuid,noatime,softdep 1 2
```

Ports and packages

- Packages dont get security updates!
- Therefore configure ports to use packages if possible
- And follow the stable ports branch

```
echo "FETCH_PACKAGES=yes" >> /etc/mk.conf
cd /usr
cvs -qd anoncvs@anoncvs.ca.openbsd.org:/cvs get -rOPENBSD_5_9 -P ports
```

- Which packages / ports need to be updated?

```
/usr/ports/infrastructure/bin/out-of-date
```

- Update a port

```
cd /usr/ports/<portname>
make update
```

- Possible binary updates through packages from https://stable.mtier.org/

Update base system

- Follow patch branch

```
cd /usr
cvs -qd anoncvs@anoncvs.ca.openbsd.org:/cvs get -rOPENBSD_5_9 -P src
cd /usr/src/sys/arch/$(uname -m)/conf
config GENERIC
cd /usr/src/sys/arch/$(uname -m)/compile/GENERIC
make clean && make
make install
reboot
rm -rf /usr/obj/*
cd /usr/src
make obj
cd /usr/src/etc && env DESTDIR=/ make distrib-dirs
cd /usr/src
make build
```

Upgrade to a new release

- Refer to the upgrade documentation e.g. http://www.openbsd.org/faq/upgrade59.html
- If you cannot upgrade by CD / USB / PXE use the `Upgrade without the Install Kernel` documentation

Set clock to localtime

```
ln -sf /usr/share/zoneinfo/right/CET /etc/localtime
rdate -ncv time.fu-berlin.de
```

Increase / derease volumne

```
mixerctl outputs.master=100,100
```

WPA-PSK

- Create /etc/hostname.<ifname>

```
nwid <ssid>
wpa
wpakey <passphrase>
dhcp
```

WPA enterprise

- Install wpa-supplicant
- Create /etc/wpa-supplicant.conf

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel

ap_scan=0
eapol_version=1
fast_reauth=1

network={
      key_mgmt=WPA-EAP
      proto=WPA2
      eap=PEAP # or TTLS
      pairwise=CCMP
      group=CCMP
      phase1="peaplabel=0"
#     phase1="tls_disable_tlsv1=1 tls_disable_tlsv1_1=2" # if only sslv3 is supported
      phase2="auth=MSCHAPV2"
      ca_cert="/etc/certs/QV2.pem"
      ssid="<ssid>"
      identity="<username>"
      password="<mysecetpass>"
}
```

- You must setup wiki card before running wpa_supplicant!
- It is better to set the bssid
- wpaakms must be configure otherwise wpa_supplicant will fail!

```
ifconfig urtwn0 nwid <ssid> bssid <mac_of_ap> wpa wpaakms 802.1x up
wpa_supplicant -B -c /etc/wpa_supplicant.conf -D openbsd -i urtwn0
```

UTF-8 system-wide

```
echo 'export LC_ALL="en_US.UTF-8"' >> /etc/profile
echo 'export LC_ALL="en_US.UTF-8"' >> ~/.xsession
```

Adjust max memory size

- Edit /etc/login.conf

```
:datasize-max=1024M:\
:datasize-cur=1024M:\
```

- Or set *infinity:* as value

NTFS

- Built-in NTFS support is read-only
- Install ntfs-3g from ports to get write support

Flash support

- Adobe and Chrome flash plugins do not work on OpenBSD
- But you can use Gnash in Firefox

```
pkg_add gnash
mkdir /home/<user>/.mozilla/firefox/<account_id>.default/plugins
cd /home/<user>/.mozilla/firefox/<account_id>.default/plugins
ln -s /usr/local/lib/mozilla/plugins/libgnashplugin.so
```

Permanently disable kernel features like ACPI

```
mv /bsd /bsd.old
config -e -o /bsd /bsd.old
ukc>disable acpi
ukc>quit
```

Automatically adjust cpufreq

- Edit /etc/rc.conf.local

```
apmd_flags="-A"
```

Linux compatibility (untested yet)

- Currently only works on i386!
- You may need to build a custom kernel

```
cd /usr/src/sys/arch/$(uname -m)/conf
cp GENERIC.MP MYKERNEL
echo "option COMPAT_LINUX" >> MYKERNEL
config MYKERNEL
cd ../compile/MYKERNEL
make depend
make
make install
reboot
```

- Now you can activate it with

```
sysctl kern.emul.linux = 1
```

- And start your Linux program
- If it is dynamically linked you need to provide all libs under /emul/linux (easiest way is to unzip a base package e.g. fedorabase there)
- For more information see http://www.openbsd.org/papers/slack2k11-on_compat_linux.pdf

List all available disks

```
sysctl hw.disknames
```

Ksh config

- ~/.kshrc

```
export PS1='\[\t\] \u@\h:\w\$ '
export EDITOR=/usr/local/bin/zile

set -o emacs

alias cp='cp -i'
alias mv='mv -i'
alias rm='rm -i'
```

- If you use tmux or screen put the following into ~/.profile

```
export ENV=~/.kshrc
```

Which program is listening on port x?

- Install lsof

```
lsof -i :<port>
```

Readmes for packages

- Can be found in /usr/local/share/doc/pkg-readmes

Fix arrow keys in Emacs under Xorg

```
(if (not window-system)                          ;; Only use in tty-sessions.
  (progn
    (defvar arrow-keys-map (make-sparse-keymap) "Keymap for arrow keys")
    (define-key esc-map "[" arrow-keys-map)
    (define-key arrow-keys-map "A" 'previous-line)
    (define-key arrow-keys-map "B" 'next-line)
```

```
    (define-key arrow-keys-map "C" 'forward-char)
    (define-key arrow-keys-map "D" 'backward-char)))
```

Automatic installation over PXE

- Possible with autoinstall
- http://www.bsdnow.tv/tutorials/autoinstall

Tracing kernel calls

- Comparable to strace on Linux

```
ktrace -t cn <program>
kdump | less
```

Building images for cloud and embedded devices

- Read http://stable.rcesoftware.com/resflash/

OpenSSL

Basic stuff

- Check a certificate

```
openssl x509 -in <cert_file> -noout
```

- Show a certificates properties

```
openssl x509 -in <cert_file> -noout -text
```

- Show expiry date of cert

```
openssl x509 -in <cert_file> -noout -enddate
```

- Generate a certificate request (CSR)

```
openssl req -new -newkey rsa:1024 -nodes -keyout key.pem -out cert.pem
```

- Generate CSR with existing key

```
openssl req -new -key key.pem -out cert.pem
```

- Generate a self signed cert

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem
```

- Check a private key

```
openssl rsa -in <key_file> -check
```

- Remove password from a private key

```
openssl rsa -in <key_file> -out <key_file>
```

- Test an SSL port

```
openssl s_client -connect localhost:443 -state -debug
GET / HTTP/1.0
```

- Convert PFX (IIS) to PEM

```
openssl pkcs12 -in mycert.pfx -out mycert.pem
```

- View the details of a certificate revocation list (CRL)

```
openssl crl -in filename.crl  -noout -text
```

- Verify a cert and check crl

```
openssl verify -crl_check -CApath /etc/ssl/certs cert.pem
```

CA stuff

- Build your own CA

```
/usr/lib/ssl/misc/CA.pl -newca

on Arch Linux /etc/ssl/misc/CA.pl
```

- Create a new certificate

```
/usr/lib/ssl/misc/CA.pl -newcert
```

- Sign a certificate

```
/usr/lib/ssl/misc/CA.pl -sign
```

- Create a Certificate Revocation List (CRL)

```
openssl ca -gencrl -keyfile ca_key -cert ca_crt -out my_crl.pem
```

- Revoke a certificate

```
openssl ca -revoke bad_crt_file -keyfile ca_key -cert ca_crt
openssl crl -in crl_file -noout -text
```

Java keystore

- How to convert a PEM cert and RSA key to PKCS12 and import it into a java keystore

```
openssl pkcs12 -export -in mycert.pem -inkey my.key -out mycert.pkcs12
keytool -importkeystore -deststorepass mypassword -destkeystore keystore.jks -
↪srckeystore mycert.pkcs12 -srcstorepass mypassword
```

- add -ext san=dns:www.example.com for alternative names

Generate random bytes

```
openssl rand <nr_of_bytes>
```

PF (OpenBSD)

PF the BSD Firewall

All you will find here may as well work on freeBSD or netBSD but some thinks may work on ‘’openBSD” only.

To let the openBSD a firewall between 2 or more networks you first have to set it up to forward traffic like defined in the routing table this is must off the time referred as a gateway or router.

Exec the following commands to enable IP Forwarding temporarily.

```
sysctl net.inet.ip.forwarding=1
```

To make this setting permanent add the following line to the file /etc/sysctl.conf

```
net.inet.ip.forwarding=1
```

pfctl

This utility is provided with every PF installation and controls the PF demon. The following commands can be used to control the PF firewall.

```
pfctl -e # enable pf
pfctl -ef <file-name> # load PF ruleset in file-name
pfctl -d # disable pf
pfctl -v # verbose output
pfctf -nf <file-name> # compiles PF ruleset but do not load them (got for testing)
```

PF boot up

To start PF at boot up you have to add the following lines to the ‘’/etc/rc.conf.locale’‘

```
pf=YES
```

PF reload

To restart, stop or start PF in a running system use:

```
/etc/rc.dpf restart
```

PF Debugging

PF could be simply debugged via the pflog0 interface that behaves as a normal interface. To see all traffic and its PF state use the command:

```
tcpdump -netttti pflog0
```

pf rules

If not defined other in the ‘’/etc/rc.conf.locale” the PF ruleset is configured in the file ‘’/etc/pf.conf’‘

processing the rules

Since version 4.2 die Option ‘’set rulset-optimization” is set to ‘’basic” which activates PF to optimize the ruleset when loading i

- remove duplicate rules
- remove shadowing rules
- put rules together for easy handling
- resort the quick rules for better passing

If this Option is set to “profile” PF will use the loaded ruleset as a profile for the real network traffic to set the order for the quick rules to be most effective.

simple client rule set

From the client point of view the simplest ruleset would be to allow everything out to the internet but noting in from the internet.

```
block in all
pass out all keep state
```

You could even leave the ‘’keep state” option away, because the PF firewall is keeping the state of the connections in default mode.

lists, macros and tables

To make you PF rules more readability you can us lists, macros and tables.

```
pass proto { tcp http https  } to port 80 # a list is defined in { }
ext_if = bxn0 # defines the macro ext_if
pass proto tcp on $ext_if # uses the macro ext_if
testclient = 192.168.2.13
table <client> persist {192.168.2.0, 192.168.2.5, $testclient} #defines the table␣
↪client with a list of clients
table <server> persist file "/etc/servers" # loads the line separated list of servers␣
↪into the table server
```

To see the input in a table of a running PF firewall use:

```
sudo pfctl -t clients -T -T show
```

keep it simple (IN ON, OUT ON or ALL)

If you change your point of view from the client side to the network as a gateway you will start to think about how to allow traffic between networks. For this you could have a rule like this:

```
pass IN proto tcp ON server1 from server1:network to server2.network
```

If you like to let the traffic into the network of the server2, you will will have to define a second rule like this:

```
pass OUT proto tcp ON server1 from server1:network to server2.network
```

To make your configuration more simple at this point you could define the following rule, which includes bout off the rules above.

```
pass proto tcp from server1:network to server2:network
```

For every rule part that is not defined PF will set in ALL, that could be tricky at some times. SO be careful where and when to reduce you rules.

first small network rule set

```
# First define lists macros and tables
ext_if = em0 # external interface
int_if = em1 # internal interface
int_net = $int_if:network # local network
icmp_echoreq =  "echoreq"

# start the rule set
block ALL # always the block rule first
pass from { lo0, $int_net } # in loopback we trust

# let everything pass from the internal network a nat rule is needed
nat on $ext_if from $int_net to any -> ($ext_if)

# troubleshooting friends

# icmp if not allow all icmp allow echo requests for ping
pass inet proto icmp icmp-type $icmp_echo_echoreq

# allow traceroute to pass
pass out on $ext_if inet proto udp port 33433 >< 33626 # opens the udp ports between␣
↪334433 and 33626 from the local network
```

shaping and bandwidth spliting queues

IN PF you could define 3 difrent types of ALTQ (Alternate Queueing). To see the stats of a queue you could use the following command. IF you like to have the traffic from a pas rule put into a queue you will have to add the queue after the pass rule with ‘queue (queunames)’ breakets

```
systat queues
```

With priq (priority-based queues) ALTQ you could define 16 difrent prioritys from 0 - 1. Every queue rule needs a pass rule to so that the traffic could pass throuw the queue.

```
# define a ALTQ priq
ext_if="bnx0"
 altq on $ext_if priq bandwidth 10M queue { priorityQ, defaultQ }
   queue priorityQ priority 7
   queue defaultQ priority 1 priq(default)

pass quick $ext_if proto tcp queue (defaultQ, priorityQ)
```

With cbq (class-based queues) ALTQ you could define 8 diffrent prioritys with a procentage of the total bandwidth.

```
# define a ALTQ cdq
ext_if="bnx0"
 altq on $ext_if cdp bandwidth 10M { default, ssh, icmp }
   queue default bandwidth 65% cdq(default, borrow red)
   queue ssh bandwidth 30% cdq(borrow red) { ssh_default, ssh_bulk }
      queue ssh_default priority 7 bandwidth 30%
      queue ssh_bulk priority 0 bandwidth 70
   queue icmp bandwidth 5% cbq
```

With the 'borrow' argument the queue could borrow bandwidth from its parent queue while the 'RED' agument avoids to borrow bandwidth from the parent queue. Within a cbq you could also define a other queue like the ssh-default aund ssh_bulk queue.

With hfsc (hierachical Fair Service Curve) ALTQ allows you to define guarantions for minimum bandwidth allocation and hard upper limits. You could define 8 diffrent prioritys from 0 to 7. You even could allocate bandwidth over a spezified time.

```
# define a ALTQ hfsc
ext_if="bnx0"
 altq on $ext_if hfsc queue { default, icmp }
   queue default bandwidth 95% priority 7 qlimit 100 hfsc (realtime 60%, linkshare 90
↪%) { defaultQ, webQ, sshQ, dnsQ }
      queue defaultQ bandwidth 10% priority 3 hfsc (realtime 20%, linkshare 50% red)
      queue webQ bandwidth 75% priority 7 hfsc (realtime 70%, linkshare 10% red)
      queue sshQ bandwidth 10% prioity 5 hfsc /realtime (realtime 50%, linkshare 30%)
      queue dnsQ bandwisth 5% priority 7 qlimit 100 hfsc (realtime (20Kb 3000 6Kb),␣
↪linkshare 5%)
   queue icmp bandwidth 5% priority 0 qlimit 200 hfsc (realtime 0, upperlimit 2%,␣
↪linkshare 90%)
```

With the 'realtime' agument you spezify the minimum bandwidht limit. The 'qlilit' argument spezify how many packets will be queueed if they could not be transmitted directly.

Redundancy and Failover (CARP)

The Common Address Redundancy Protocol (CARP) is open source and under the openbsd license it was developed as an alternative to the HSRP (Hot Standby Router Protocol) RFC 2281 from Cisco and the VRRP (Virtual Router Redundancy Protokol) RFC 3768 from Nikia.

The main funktion of the CARP protocol is to allow tow or more systms to be in charge for the same ip address and share it or to automaticaly take it over. If CARP is in active passive mode, the avtice maschine ist called the master and all passive maschines are called backup.

The CARP protocol like the HSRP and the VRRP protocol, is a multicast protocol. CARp and HSRP usees the mulitcast address asiend be IANA 224.0.0.18 to exchange there informations. This makes it extremly risky to not set a password for the carp comunication for hte security point of view.

To exchange the pf states (mostly TCP states) the tool pfsync is needed. This allowes all systems to handel active connections correctly. The main disadvatage of pfsync is that its traffic is not authentified or encrypted. So the best way to use pfsec is to use dedicated ports and connect them over a cross over cable or use a host to host vpn to encrypt the traffic beetween the firewalls. Any way you shall not use the same IP frame used for your internal or external networks for the pfsync connection so that this inforamtions are not exchanged be mistake over the wrong interface.

To be able to use CARP with OpenBSD you will have to enable the following sysctl values.

```
sysctl net.inet.carp.allow=1
```

To enable it over an reboot you will have to enable it in the /etc/sysctl.conf.

```
net.inet.carp.allow=1
```

Passiv Active Mode

Here we are going to set up a active firewall and a identical secound firewall that should take over if the active firewall failes. The take over will work with no interuption of the active connection and no noticeable downtime.

IF you do not have a console based access to each maschine you should first of all assign a IP address to each interface that is not the virtual CARP IP. Otherweath you could could never know to with maschine you are connected and allows only get accesss to the maschie that is the avtice on in the CARP group and holds the virtual CARP IP.

For the following exaples we assume that we have the internal IP frame 192.168.0/24 and the external IP Frame 10.10.10/24. We will configure the IPs 192.168.0.1 as the internal CARP address and the IPs 10.10.10.1 as the external CARP address. To be able to communicate with the maschines we will give the active firewall the IP 192.168.0.2 and the passiv maschine the IP 192.168.0.3. The external interface we will just give an CARP IP to share between each other. For the pfsync connection we will configure 2 more Interfaces with the IPs 172.16.0.1/30 and 172.16.0.2/30.

Setup pysikal internal interfaces

To set up the IPs for the internal physikal interface add the following line in a file named /etc/hostname.<interface_name> with your faviried editor.

```
zile /etc/hostname.bnx1
up description external interface
zile /etc/hostname.bnx2
192.168.0.2/24 description internal communication
```

On the other maschien we do the same with the other ip

```
zile /etc/hostname.bnx1
up description external interface
zile /etc/hostname.bnx2
192.168.0.3/24 description internal communication
```

To activate the interface execute the following command on both maschienes.

```
/etc/netstart
```

If you do not specify the interface name all interfaces will be configured as discribed in the /etc/hostname.* files. You could also configure all interfaces via the ifconfig command but then the configration will be lost after an reboot.

Setup the CARP virtual interfaces

A CARP interface is configured like other interfaces in /etc/hostname.carp<nummer>. It has needed parameters and optional parameters.

- vhid (virtual host ID) hast to unique within the network broadcast domain. It is needed to identify the interfaces that shared the virtual IP address.
- advbase This is the internal clock pulse generator of the carp connection. Eeach carp memebr sends its helo paket after this counter. The default value is 1.
- advskew This Parameter should be set for each backup. It is added to the advbase parameter so that the backend sends its helo packets slower then the master maschine. For this it also indicates how much less perferred a maschine is to take ofer the master. The higher the value the less likely it is that the maschien takes over the master state. The default value of this parameter is 0.
- pass This specifies a password that if set is needed for all CARP interfaces that have the same vhid

With all this parameters the external and internal carp interface could be configured like this.

```
zile /etc/hostname.carp1
10.10.10.1/24 vhid 2 advskew 20 carpdev bnx1  pass ppppp
description external carp master
zile /etc/hostname.carp2
192.168.0.1/24 vhid 1 advskew 20 carpdev bnx2 pass PPPPP
description internal carp master

zile /etc/hostname.carp1
10.10.10.1/24 vhid 2 advskew 120 carpdev bnx1 pass ppppp
description external carp backup
zile /etc/hostanme.carp2
192.168.0.1/24 vhid 1 advskew 120 carpdev bnx2 pass PPPPP
description internal carp backup
```

To up this interfaces we need again to execute the command /etc/netstart

With this config the master is sending ist helo pakets each 1,20 secounds and the backend is sending the helo packet every 2,20 secounds.

State Synchronization (pfsync)

To have a pf state-table synchronization, you will only need to configure a pysikal interface and a virtual pfsync interface to have the state synchronisation in openbsd, since pfsync is an virtual network interface.

```
zile /etc/hostname.bnx0
172.16.0.1/30
description pysical sync interface
zile /etc/hostname.pfsync0
up syncdev bnx0 syncpeer 172.16.0.1
```

If you put a syncpeer option in the syncdev configuration the sync device will only exapte traffic from this ip and send the sync traffic to this IP. BUt this is only possible if you have only 1 backup maschie in your carp group. A other way to protect your sync traffic if to create a ipsec tunnel between the servers and run the sync traffic over it.

PF Rule Set

Last but not least you will need a pf ruleset to be able to allow the traffic betweenn the interfaces.

SELinux

Overview

- .te -> type enforcement = allow rules, new types, user etc (used most of the time)
- .fc -> file context = define file context rules for module (<regexp> <security_context>)
- .if -> interfaces = macros

Update policy

- Use a unique policy name otherwise it can clash with system internals and result in strange error messages

```
grep qemu-system-x86 /var/log/audit/audit.log | audit2allow -M <policy_name>
semodule -i <policy_name>.pp
```

- Or to allow all since the last policy change

```
audit2allow -alR
```

Show all policy modules

```
semodule -l
```

Get rid of a policy

- Disable

```
semodule -d <policy_name>
```

- Remove

```
semodule -r <policy_name>
```

Write your own policy module

- Allow rules have the definition `allow <from_type> <to_type> : <object_class> {permissions};`
- Every type / attribute / class used and not defined in module must be required
- Choose a good name (not mypol) to avoid clashing with other predefined modules
- Copy Makefile from /usr/share/selinux/devel/

```
policy_module(mypol, 1.0)

require {
  type httpd_t;
}
```

```
type my_type;
allow httpd_t my_type : file { getattr read };
```

- All object classes can be found in `/usr/src/redhat/BUILD/serefpolicy-<version>/policy/flask/security_classes`
- All permissions for a class can be found in `/usr/src/redhat/BUILD/serefpolicy-<version>/policy/flask/access_vectors`

Check policy module

```
checkmodule -m some.te
```

Compile a te file by hand

```
make -f /usr/share/selinux/devel/Makefile some.pp
```

Search a policy rule

```
sesearch -A | grep <whatever>
```

- To see all allow rules with type httpd_t as source

```
sesearch -a -s httpd_t
```

- or to see what a boolean / macro does (needs policy.conf see below)

```
apol
```

Generate a policy skeleton

```
sepolicy generate --application /usr/bin/firefox
sepolicy generate --init /path/to/my/init-service
```

Booleans

- Show all booleans

```
semanage boolean -l
getselbool -a
```

- Set a boolean

```
setsebool -P <boolean> <value>
```

- All local changes are in `/etc/selinux/<policy>/modules/booleans.local`

Write your own boolean

```
bool mybool <defaultvalue>;
tuneable_policy(`mybool', `
  allow statements
');
```

- Name can be combined with || or && and other boolean names to activated this boolean only if condition is true

Managing file contexts

- SE Linux stores the security context for files directly in the filesystem (currently ext{2,3,4}, XFS, JFS, Btrfs)
- Last rule matches
- Show file context

```
ls -Z
```

- Show all context rules

```
semanage fcontext -l
```

- Set new file context rule

```
semanage fcontext -a -t mysqld_db_t '/some/dir(/.*)?'
```

- Reset context rules for dir

```
restorecon -RFvv /some/dir
```

- Copy context

```
chcon -R --reference=/old/dir /new/dir
```

- Permanently set same context as other directory

```
semanage fcontext -a -e /var/www /srv/www
```

- Delete a file context

```
semanage fcontext -d <dir>
```

- Automatically relabel all files on next boot

```
touch /.autorelabel
```

List all roles

```
seinfo -r
```

Change role

```
newrole -r system_r -t unconfined_t
id -Z
```

Start a program in a specific role

```
runcon system_u:system_r:crond_t:s0 /bin/bash
```

Configure users

- List all users

```
seinfo -u
```

- Map Unix user to SELinux user

```
semanage login -a -s user_u <unix_user>
semanage login -l
```

- Map SELinux user to roles

```
semanage user -a -R "user_r sysadm_r" user_u
semanage user -l
```

Log everything

```
semanage dontaudit off
```

- or

```
semanage -DB
```

Reset base policy

```
semodule -B
```

Generate policy.conf (source file of your policy)

- install src rpm of policy

```
rpmbuild -bp selinux-policy.spec
cd BUILD/serefpolicy-<version>
```

- Edit `build.conf` and set type to mcs, name to whatever, distro to redhat and monolithic to y

```
make bare conf
cp ../../SOURCES/boolean-targeted.conf policy/booleans.conf
cp ../../SOURCES/modules-targeted.conf policy/modules.conf
make policy.conf
```

- To make a module policy set MONOLITHIC=n and `make base.pp` instead of make policy.conf
- If apol complains it cannot load policy due to whatever failure just delete those line(s)

Configure Non-executable stack / heap

```
setsebool -P allow_execstack 0
setsebool -P allow_execmem 0
```

Kernel parameter

```
selinux=0|1
enforcing=0|1
autorelabel=0|1
```

Switch to MCS or MLS policy

- Install policy rpm
- Edit `/etc/selinux/config`

```
touch /.autorelabel
reboot
```

- Boot with `enforcing=0`
- Reboot after relabeling

Define new category

- Edit `/etc/selinux/targeted/setrans.conf`

```
s0:c0=NotImportant
s0:c100=VeryImportant
```

- Restart mcstrans

Change category of a user

```
semanage login -a -r <category> <user>
```

Change category of file / dir

- Multiple categories are AND conditions

```
chcat +|-<category> <file|dir>
```

Write your own macro

```
define(`macro_name', `allow $1 $2: file { getattr read }');
```

Domain transition

```
init_daemon_domain(myproc_t, myfile_exec_t)
domain_auto_trans(unconfined_t, myfile_exec_t, myproc_t)
auth_domtrans_chk_passwd(myproc_t)
auth_domtrans_upd_passwd(myproc_t)
```

Mysql config

- Change datadir

```
semanage fcontext -a -t mysqld_db_t '/new/dir/mysql(/.*)?'
restorecon -RFvv /new/dir/mysql/
```

- For more see *man mysqld_selinux*

Apache config

- Allow cgi scripts

```
setsebool -P httpd_enable_cgi 1
```

- Allow webserver scripts to connect to the network

```
setsebool -P httpd_can_network_connect 1
```

- Run apache on non-standard port

```
semanage port -l | grep http
semanage port -a -t http_port_t -p tcp 8888
```

- For more see *man httpd_selinux*

NFS / Mounting

- Specify security context with mount parameter `--context=<security_label>` to have all files / dirs that security label or
- `--defcontext=<security_label>` to define a label just for those unlabeled

Temporarily disable / enable SELinux

```
setenforce [0|1]
```

Audit Framework

- For permanent rules edit `/etc/audit/audit.rules`
- Show current status

```
auditctl -s
```

- Enable / disable audit

```
auditctl -e 0/1
```

- Show all rules

```
auditctl -l
```

- Delete all rules

```
auditctl -D
```

- Log all execve calls of user root

```
auditctl -a exit,always -S execve -F euid=0
```

- Log all executions of a specific program

```
auditctl -A exit,always -F path=/path/to/executable -S execve
```

- Suppress all successful executions of some program

```
auditctl -w /path/to/executable -F success=1
```

- Show all logs of a specific timespan and from a certain user

```
ausearch --start month/day/year time --end month/day/year time -ui 0
```

- Show recent events (last 5 minutes)

```
ausearch -ts recent
```

Documentation

- http://www.selinuxproject.org/page/User_Resources
- http://www.admin-magazin.de/Online-Artikel/Mandatory-Access-Control-MAC-mit-SE-Linux
- http://magazine.redhat.com/2007/08/21/a-step-by-step-guide-to-building-a-new-selinux-policy-module/
- https://www.youtube.com/user/domg4721/videos

CHAPTER 21

Hacking

Layer 2

- ein trunk port auf einer cisco ist nur dazu da, um vlan an einen switch oder router weiter zu reichen
- ein trunk port dient nicht dafuer einen normalen computer in mehrere vlan zu connecten!
- linux kann switch spielen und sich so an einem trunk port in jedes über diesen port geroutetes vlan einklinken

```
vconfig eth0 add <vlan-id>
ifconfig eth0.<vlan-id> <ip_aus_vlan> up
```

- wenn ein switch port auf dynamic desirable oder auto steht (man bekommt dtp packets), kann man den port wahlweise mit yersinia oder scapy in den trunk modus fuer alle vlan schalten

```
from scapy.layers.l2 import Dot3 , LLC, SNAP
from scapy.contrib.dtp import *
scapy> negotiate_trunk(iface="eth0", mymac="7E:D4:DE:A5:45:21")
```

- das anschalten des trunk modes kann ein paar minuten dauern!

Reverse Engineering

2 byte = 1 word 4 byte = 1 dword

ebx = base pointer (e.g. array start) ecx = counter register edx = data register esi = source register edi = destination register

z-flag - zero flag (is zero is last result was zero) cf / of - overflow flags sign flag - indicated if integer is signed or unsigned

wenn zwei gleiche zahlen mit cmp verglichen werden wird z-flag gesetzt wenn beide gleich sind, wenn die erste zahl kleiner als die zweite ist, ist s-flag 1 (bei signed, bei unsigned ist c-flag 1), wenn beide 0 sind ist die erste zahl größer

test = AND (setzt z-flag auf 1 wenn src und dst gleich sind) xor mit zwei gleichen registern schreibt 0 in das register

adressen >= 0x80000000 sind kernel address space

jeder prozess hat mindestens einen thread (es gibt kein fork unter windows) jeder thread hat 2 stacks einen für user und einen für kernel mode, denn ein thread wird via kernel syscalls in den priviledged mode geschaltet

prozess initialisierung: 1. CreateProcess legt ein neues Process Object im Kernel an, mapped die EXE + NTDLL.DLL ins RAM, startet einen thread und allokiert stack space LdrpInitialize (NTDLL.DLL) liest import tables und lädt alle DLLs nach dann wird BaseProcessStart (KERNEL32.DLL) aufgerufen -> startet WinMain

dos header (beinhaltet ein dos programm das ausgibt, dass das programm nicht unter dos läuft) ein zusätzlicher eintrag (e_lfanew) zeigt auf den wirklichen pe header

die struktur eines PE binary auf der platte ist die selbe wie im ram d.h. wenn man etwas interessantes in der datei gefunden hat ist es an der selben virtuellen adresse im ram

sectionheader ist normalerweise zwischen imagebase start und imagebase + 1000 imagebase bekommt man via getmodulehandlea

die adresse im pe header sind alle relativ zur imagebase (und "verkehrt herum" sprich lil endian)

es gibt eine section pro importierter dll und jede section hat eine imported address table (IAT) IAT mapped die dll function adresse auf die virtuelle adresse im ram (wie GOT section im ELF header)

directories sind optionale header mit denen ein PE executable erweitert werden kann z.b. import / export table, - resource table (hier werden bilder und statische strings referenziert), - base relocation table (beinhaltet eine liste von adressen, die neuberechnet werden müssen), - thread local storage table (enthält eine section die privat für den aktuellen thread ist)

wenn eine exception auftrifft ruft der kernel alle functionen aus der exception handler list (in der thread informaion block section) auf und jede funktion entscheidet dann selbst ob sie sich um die exception kümmert oder nicht

KERNEL32.DLL beinhaltet I/O, memory- process- und thread management GDI32.DLL deckt alle low-level gui dienste ab wie linien zeichnen USER32.DLL implementiert alle GUI dienste wie menus, dialoge, window-management etc

switching to kernel mode vor windows2000 wurde die syscall nr in eax geladend und int 2e aufgerufen jetzt wird dieser aufruf hinter dem befehl sysenter versteckt (wird immer nur via SystemCallStub aufgerufen weil sysenter selbst sich nicht den stackpointer merkt)

Scapy

Basics

- Get all protocols

```
ls()
```

- Get all options of a protocol

```
ls(TCP)
```

- Get all commands

```
lsc()
```

- Show description of function

```
help(sniff)
```

=== Sending packets ===

- Just a simple Ping

```
send(IP(dst="192.168.1.1")/ICMP())
```

- send() send on layer 3 (IP)
- sendp() send on layer2 (Ethernet)
- Send a packet on layer 3 and wait for response

```
(resp, unans) = sr(IP(dst="192.168.1.1")/ICMP(), timeout=3)
```

- print destination packet

```
print resp[0][1].show()
```

- sr1 for sending on layer2

Sniffing

```
def handle_packet(packet):
    ip = packet.getlayer(scapy.IP)
    tcp = packet.getlayer(scapy.TCP)

    print "%s:%d -> %s:%d" % (ip.src, tcp.sport
                              ip.dst, tcp.dport)

scapy.sniff(iface=dev, filter="tcp and port 80", prn=handle_packet)
```

- other way to decode the packet

```
print packet[IP].src
```

Useful utils

- Generate random mac / ip

```
RandMAC("*:*:*:*:*:*")
RandIP("*.*.*.*")
```

- Get your own mac / ip

```
get_if_hwaddr("eth0")
get_if_addr("eth0")
```

Awesome oneliners

- Find sniffers on your network

```
promiscping("192.168.1.0/24")
```

- SYN portscan

```
ans, unans = sr(IP(dst="www.chaostal.de")/TCP(dport=range(1,1024), flags="S"),␣
↪timeout=1)
```

- TCP fuzzer

```
send(IP(dst="192.168.1.1") / fuzz(TCP()), loop=1)
```

- mac flooder

```
sendp(Ether(src=RandMAC("*:*:*:*:*:*"), dst=RandMAC("*:*:*:*:*:*")) / IP(src=RandIP(
↪"*.*.*.*"), dst=RandIP("*.*.*.*")) / ICMP(), loop=1)
```

Tcpdump

Show RST packets on all interfaces

```
tcpdump -i any -n -v 'tcp[tcpflags] & (tcp-rst) != 0'
```

Sniff wifi traffic without beacons, probe requests and responses

```
tcpdump -y IEEE802_11_RADIO not subtype beacon and not subtype probereq and not␣
↪subtype proberesp
```

Web Security

Information disclosure

- CVS/Entries .svn/entries .git/index
- hidden parameter debug / test / trace = true / on / 1
- docs / logs / backup / conf / test

Authentication

- Common users: admin, test, demo, guest, $company, $product
- Common passwords: test, test123, password, password1, password123, qwertz, qwerty, letmein
- Look at: Password Recovery, Remember me

ACL bypass techniques

- Try GET instead of POST and vice versa
- Try HEAD instead of GET
- Double-Slash URL
- Hidden Parameter like loggedin / isadmin = true / on / 1
- change userid

Cracking Session Ids

- hex / base64 encoding
- predictable token depending on bad algo e.g. +1 or +time
- weak generation e.g. user AND time -> md5

Fooling Filters

- Double the input <scr<script>ipt>
- Bypass by sequence <scri'pt>
- urlencode (double urlencode)
- unicode
- escape / escape escape

SQL injection

- UNION SELECT NULL, NULL
- ' or double ''
- output von 2 vergleichen mit 1+1
- output vergleichen von order by 1 2 3
- select table_schema, table_name from information_schema.tables
- select into outfile
- select load_file('/etc/passwd')

Ajax

- var request = new XMLHttpRequest();
- request.open('GET', '/muh', true);
- request.send();

```
var request = new XMLHttpRequest();
request.onreadystatechange = handler;
request.open('POST', '/muh', true);
request.send("name=balle&pass=maeh");
```

Bypass Same Origin

- Set header

```
Access-Control-Allow-Origin: *
```

Webworker

- Run javascript in the background

```
var worker = new Worker("worker_script.js");
worker.onmessage = function(e){
  e.data
};
worker.postMessage("start");
```

Client / Session Storage

```
sessionStorage.setItem('key', 'value');
sessionStorage.getItem('key')
sessionStorage.deleteItem('key')
```

Web SQL

```
var db = openDatabase('mydb', '1.0', 'my first database', size)
db.transaction(function (tx) {
 tx.executeSql('CREATE TABLE foo (id unique, text)');
}
```

Misc

- Check negative numbers
- zwei sessions / operationen exakt gleichzeitig ausführen

Wireless

- http://www.aircrack-ng.org/
- http://karmetasploit.com/
- http://www.digininja.org/projects/wifi_honey.php
- http://www.securitytube.net/user/Vivek-Ramachandran
- http://www.kismetwireless.net/

Wireshark

Shortcuts

Wifi

- View -> Wireless Toolbar
- http://sharkfest.wireshark.org/sharkfest.10/B-5_Parsons%20HANDS-ON%20LAB%20-%20WLAN%20Analysis%20with%20Wireshark%20&%20AirPcap%20Exercises.pdf
- hide beacons

```
wlan.fc.subtype != 8
```

- filter by ssid

```
wlan_mgt.ssid == "Spatula City"
```

- filter by channel (e.g. channel 11)

```
radiotap.channel.freq == 2462
```

- only sniff data frames

```
wlan.fc.subtype == 2
```

- sniff probe request / response

```
wlan.fc.subtype==4 or wlan.fc.subtype==5
```

- retransmissions

```
wlan.fc.retry == 1
```

WEP / WPA

- Decrypt WEP / WPA traffic with existing key
- Preferences -> Protocols -> IEEE 802.11 -> Enable decryption + Add decryption keys

SSL

- Edit preferences -> protocols -> SSL
- Put the following into RSA key list

```
192.168.x.x,443,http,/path/to/keyfile.pem;
```

- One could also specify 0.0.0.0 as ip, 0 as port and data as protocol
- Afterwards right click on packet and choose Follow SSL Stream
- Filter SSL handshake

```
ssl.record.content_type==22
```

- Decrypt and display data from dump file

```
tshark -o "ssl.desegment_ssl_application_data: TRUE" -o "ssl.keys_list:,443,http,rsa_
↪private.key" -o "ssl.debug_file:rsa_private.log" -r all.pcap -R "(tcp.port eq 443)"␣
↪-V
```

Detect ARP storms

- Preferences -> Protocols -> ARP -> Detect ARP request storms

Macros

- With Analyze -> Display Filter Macros you can give complex display filter strings an easy name and even use parameters
- E.g. ICMP redirection not from gateway ip and save it under name icmp_redir

```
icmp.type == 5 and ip.src != $1
```

- $1 will get replace by specified ip
- To use it type the following display filter

```
${icmp_redir:192.168.1.1}
```

- Macros are stored in ~/.wireshark/profiles/$profile/dfilter_macros

```
"arp_req","arp.opcode == 0x0001"
"arp_rep","arp.opcode == 0x0002"
"echo_req","icmp.type == 8"
"echo_rep","icmp.type == 0"
"ssl_handshake","ssl.record.content_type==22"
"nobeacons","wlan.fc.subtype != 8"
"ssid","wlan_mgt.ssid == \x22$1\x22"
"probes","wlan.fc.subtype==4 or wlan.fc.subtype==5"
"dns_req","dns.flags.response == 0"
"dns_res","dns.flags.response == 1"
"dns_error","dns.flags.rcode != 0"
"icmp_redir","icmp.type == 5 and ip.src != $1"
```

Frame filter

- You can filter on frame arravile time

```
frame.time == "Jan 01, 2013 00:00:00"
```

- Or on frames that took more than 1 second to the previous frame

```
frame.time_delta > 1
```

GeoIP

- Make a new dir called geoip
- Download http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz and unzip it to that dir
- Add the dir to Preferences -> Name Resolution -> GeoIP database directories
- Restart wireshark
- Statistics -> Endpoints -> IPv4 -> Map
- Edit preferences -> protocols -> ipv4 -> enable geoip (optional to filter on geoip)
- To filter on geoip information use

```
ip.geoip.country == "China"
```

HTTP

- Display filter

```
http.response.code
http.request.method
http.host
http.user_agent
http.referer contains
http.content_type
http.cookie
http contains "password"
```

- Export html pages (File -> Export -> Objects -> HTTP)

Tshark

- Display get requests, dont do dns, dump all packets with payload to all.pcap
- -f “capture filter”
- -R “display filter”
- -s snaplen
- -S decode payload
- -V Display complete packet
- -a <auto-stop-condition>
- -t a (display absolute time)
- -o “tcp.relative_sequence_numbers:FALSE” for displaying absolute sequence numbers

```
tshark -S -n -t a -o "tcp.relative_sequence_numbers:FALSE" -f "port 80"
```

- Show http get requests

```
tshark -S -n -w all.pcap -f "host www.datenterrorist.de" -R "http.request.method==GET"
```

- Capture traffic for 10 seconds, display traffic analysis for all ips

```
tshark -q -a duration:10 -z conv,ip
```

- Sniff cookies

```
tshark -T fields -e http.cookie -R "http.cookie" port 80
```

- FTP logins

```
tshark -R 'ftp.request.command == "USER" || ftp.request.command == "PASS"'
```

- Detect FTP bounce attack

```
tshark -R 'ftp.request.command == "PORT"'
```

- POP logins

```
tshark -R 'pop.request.command == "USER" || pop.request.command == "PASS"'
```

Cheat Sheets

- General filtering http://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf
- 802.11 http://www.willhackforsushi.com/papers/80211_Pocket_Reference_Guide.pdf

CHAPTER 22

Cluster

Ceph

Overview

- http://www.youtube.com/watch?v=OyH1C0C4HzM
- A monitor knows the status of the network and keeps it in its monitor map
- You can have multiple monitors but should have a small, odd number
- MDS are the metadata servers (stores hirarchy of ceph fs + owner, timestamps, permissions etc)
- MDS is only needed for ceph fs
- An OSD is a storage node that contains and servers the real data, replicates and rebalances it
- The OSDs form a p2p network, recognize if one node is out and automatically restore the lost data to other nodes
- The client computes the localization of storage by using the CRUSH algorythm (no need to ask a central server)
- RADOS is the object storage interface
- RBD (RADOS Block Device) creates a block device as RADOS object
- Placement groups (pgs) combine objects into group. Replication is done on pgs or pools not files or dirs. You should have 100 pgs / OSD
- Pool is a seperate storage container that contains its own placement groups and objects (think of mountpoint)

Status

- degraded == not enough replicas
- stuck inactive - The placement group has not been active for too long (i.e., it hasn't been able to service read/write requests).

- stuck unclean - The placement group has not been clean for too long (i.e., it hasn't been able to completely recover from a previous failure).
- stuck stale - The placement group status has not been updated by a ceph-osd, indicating that all nodes storing this placement group may be down.

Manual installation

- Setup a monitor

```
uuidgen
```

- Edit `/etc/ceph/ceph.conf`

```
fsid = <uuid>
mon initial members = <short_hostname>
mon host = <ip_address>
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd pool default size = 2
```

- Generate keys for the monitor and admin user and add the monitor to the monitor map

```
ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon
↪'allow *'
ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n␣
↪client.admin --set-uid=0 --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow'
ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.
↪keyring
monmaptool --create --add <short_hostname> <ip_address> --fsid <uuid> /tmp/monmap
```

- Create the monitor cache filesystem, start the monitor and see if it created the default pools and is running

```
mkdir -p /var/lib/ceph/mon/ceph-<short_hostname>
ceph-mon --mkfs -i <short_hostname> --monmap /tmp/monmap --keyring /tmp/ceph.mon.
↪keyring
ceph-mon -i <short_hostname>
ceph osd lspools
ceph -s
```

- Setup an OSD (note the command ceph osd create returns the osd id to use!)

```
uuidgen
ceph osd create <uuid>
mkdir -p /var/lib/ceph/osd/ceph-<osd_id>
fdisk /dev/sda
ceph-disk prepare /dev/sda1
mount /dev/sda /var/lib/ceph/osd/ceph-<osd_id>/
ceph-osd -i <osd_id> --mkfs --mkkey
ceph auth add osd.<osd_id> osd 'allow *' mon 'allow rwx' -i /var/lib/ceph/osd/ceph-
↪<osd_id>/keyring
ceph-disk activate /dev/sda1 --activate-key /var/lib/ceph/osd/ceph-<osd_id>/keyring
ceph-osd -i <osd_id>
ceph status
```

- Add another OSD for replication

- Setup a metadata server (only needed when using CephFS)

```
mkdir -p /var/lib/ceph/mds/mds.<mds_id>
ceph auth get-or-create mds.<mds_id> mds 'allow ' osd 'allow *' mon 'allow rwx' > /
↪var/lib/ceph/mds/mds.<mds_id>/mds.<mds_id>.keyring
ceph-mds -i <mds_id>
ceph status
```

Adding OSDs the easy way

- With ceph-deploy

```
ceph-deploy osd prepare node1:/path
ceph-deploy osd activate node1:/path
```

- Manually (ssh to new osd node)

```
ceph-disk prepare --cluster ceph --cluster-uuid <fsid> --fs-type xfs /dev/sda
ceph-disk-prepare --fs-type xfs /dev/sda
```

Complete setup of new node

- On new node

```
useradd -d /home/ceph -m ceph
passwd ceph
echo "ceph ALL = (root) NOPASSWD:ALL" | tee /etc/sudoers.d/ceph
mkdir /local/osd<id>
```

- On ceph-deploy node

```
su - ceph
ssh-copyid ceph@<hostname_of_new_node>
ceph-deploy install <hostname_of_new_node>
ceph-deploy osd prepare <hostname_of_new_node>:/local/osd<id>
ceph-deploy osd activate <hostname_of_new_node>:/local/osd<id>
ceph-deploy mon create <hostname_of_new_node>
```

Configure replication

- Edit ceph.conf

```
osd pool default size = 2
```

Access storage

- CEPH FUSE (filesystem access comparable to NFS)

```
ceph-fuse -m <monitor>:6789 /mountpoint
```

- FUSE via fstab

```
id=admin                /mnt  fuse.ceph defaults 0 0
```

- CEPH FS kernel client
- RADOS API for object storage

```
rados put test-object /path/to/some_file --pool=data
rados -p data ls
ceph osd map data test-object
rados rm test-object --pool=data
```

- RADOS FUSE
- Virtual Block device via kernel driver (needs kernel >= 3.4.20)

```
rbd create rbd/myrbd --size=1024
echo "rbd/myrbd" >> /etc/ceph/rbdmap
service rbdmap reload
rbd showmapped
```

- iSCSI interface under development
- Code your own client with librados

Check size

- Of the filesystem

```
ceph df
```

- Of a file

```
rbd -p <pool> info <file>
```

File snapshots

```
rbd -p <pool> snap create <file>
rbd -p <pool> snap ls <file>
rbd -p <pool> snap rollback <file>
rbd -p <pool> snap rm <file>
```

Check health

```
ceph health detail
```

- get continuos information

```
ceph -w
```

Check osd status

```
ceph osd stat
ceph osd tree
ceph osd dump
```

Check server status

```
/etc/init.d/ceph status
```

Pools

- Create

```
ceph osd lspools
ceph osd pool create <pool_name> <num_pgs>
```

- Change number of pgs

```
ceph osd pool get <name> pg_num
ceph osd pool set <name> pg_num <nr>
```

- Create a snapshot

```
ceph osd pool mksnap <name>
```

- Find out nr of replicas per pool

```
ceph osd dump | grep <pool>
```

- Change nr of replicas per pool

```
ceph osd pool set <name> size 3
```

Placement groups

- Overview

```
ceph pg dump
ceph pg stat
```

- What does the status XXX mean?

```
inactive - The placement group has not been active for too long (i.e., it hasn't been␣
↪able to service read/write requests).
unclean - The placement group has not been clean for too long (i.e., it hasn't been␣
↪able to completely recover from a previous failure).
stale - The placement group status has not been updated by a ceph-osd, indicating␣
↪that all nodes storing this placement group may be down.
```

- Why is a pg in such a state?

```
ceph pg <pg_num> query
```

- Where to find an object / file?

```
ceph osd map <pg_name> <object-name>
```

- "Fsck" a placement group

```
ceph pg scrub <pg_num>
```

Editing the CRUSH map

- The CRUSH map defines `buckets` (think storage groups) to map placement groups tp OSDs across a failure domain (e.g. copy 1 is in rack 1 and copy 2 in rack 2 to avoid power outage of one rack to destroy all copies)
- A higher weight will get more load than a lower weight

```
ceph osd getcrushmap -o crushmap
crushtool -d crushmap -o mymap
emacs mymap
crushtool -c mymap -o newmap
ceph osd setcrushmap -i newmap
```

Maintanance

- To stop CRUSH from automatically balance load of the cluster

```
ceph osd set noout
```

Troubleshooting general

- Remove everything (not recommended for production use!)

```
ceph-deploy purge host1 host2
ceph-deploy purgedata host1 host2
ceph-deploy gatherkeys
```

Troubleshooting sudo

- Make sure that visiblepw is disabled

```
Defaults   !visiblepw
```

- Is the /etc/sudoers.d directory really included?

Troubleshooting network

- The name of a osd / mon must be the official name of the host no aliases!
- Make sure you have a `public network = 1.2.3.4/24` in your ceph.conf

Repair monitor

- the id can be found by looking into `/var/lib/ceph/mon/`
- run monitor in debug mode

```
ceph-mon -i <myid> -d
```

- Reformat monitor data store

```
rm -rf /var/lib/ceph/mon/ceph-<myid>
ceph-mon --mkfs -i <myid> --keyring /etc/ceph/ceph.client.admin.keyring
```

Cluster is full

- The easiest way is of course to add new OSDs, but if thats not possible
- Try to reweight automatically

```
ceph osd reweight-by-utilization
```

- Reweight manually free OSDs

```
ceph osd tree
ceph osd crush reweight osd.<nr> <new_weight>
```

- Reconfigure `full_ratio` value and delete objects (DONT FORGET TO CHANGE IT BACK!)

```
ceph pg set_full_ratio 0.99
```

Cannot delete a file

- Check that the cluster is not full otherwise see above

```
ceph health detail
```

- Purge all snapshots

```
rbd -p <pool> snap purge <file>
```

- Check that the file is not locked and maybe remove the lock

```
rbd -p <pool> lock list <file>
rbd -p <pool> lock remove <file> <id> <locker>
```

- I can still not remove the file! (Thats the not so nice and maybe destructive way)

```
rados -p <pool> rm rbd_id.<file_id>
rbd -p <pool> rm <file>
```

Clustershell

Overview

- http://cloud.github.com/downloads/cea-hpc/clustershell/ClusterShell_UserGuide_EN_1.6.pdf

Define node groups

- Edit `/etc/clustershell/groups`

```
node_all: cluster-node[001-999].somewhere.in-the.net
```

Run a command on all nodes

```
clush -w @node_all "my_command with_params"
```

Iterate over nodes

```
for NODE in $(nodeset -e @node_all); do scp some_file root@$NODE:~/; done
```

Diff results

```
clush -w @node_all --diff "dmidecode -s bios-version"
```

Combine results

```
clush -w @node_all -b "uname -a"
```

Copy file

```
clush -v -w @node_all --copy a_file
```

Retrieve a file

- Will create files like `id_rsa.node001`

```
clush -v -w @node_all --rcopy /root/.ssh/id_rsa
```

Clush.conf

- Need ssh password auth? Install sshpass and edit /etc/custershell/clush.conf

```
ssh_user: root
ssh_path: /usr/bin/sshpass -p "password"
ssh_options: -oStrictHostKeyChecking=no
```

- Need to limit number of connection or connection / command timeouts?

```
fanout: 256
connect_timeout: 15
command_timeout: 0
```

Scripting in Python

```
from ClusterShell.Task import task_self, NodeSet

task = task_self()
task.run("/bin/uname -r", nodes="mynode[001-123]")

for output, nodes in task.iter_buffers():
    print NodeSet.fromlist(nodes), output
```

GlusterFS

Network ports

- These are the tcp ports to open in your firewall

Port	Description
616	GlusterFS
38465	GlusterFS
38466	GlusterFS
38468	GlusterFS
38469	GlusterFS
24007	GlusterFSd
49153	Bricks

Installation

- Install GlusterFS repository

```
wget http://download.gluster.org/pub/gluster/glusterfs/LATEST/CentOS/glusterfs-epel.
↪repo
```

- Install packages

```
yum install glusterfs{,-server,-fuse,-geo,-replication}
```

- Start the service

```
service glusterd start
```

Configuration

- Make a server trust another Gluster node

```
gluster peer probe <ip>
```

- Lets asume you have a partition mounted to /export/test to distribute with GlusterFS on node1 and node2
- Setup a volume

```
gluster volume create test replica 2 node1:/export/test node2:/export/test
gluster volume start test
```

- Now you can mount and use the volume

```
mount -t glusterfs node1:/export/test /mnt
```

- For redundant mount insert the following into your `/etc/fstab`

```
$GFS1_NODE:/export/test /mnt glusterfs defaults,_netdev,backupvolfile-server=$GFS2_
↪NODE 0 0
```

Peers

- Add a new one

```
gluster peer probe <ip>
```

- Show status

```
gluster peer status
```

- Remove one

```
gluster peer detach <ip>
```

Volumes

- Create a new one

```
gluster volume create test replica 2 node1:/export/test node2:/export/test
gluster volume start test
```

- List all volumes

```
gluster volume status
```

- Remove one

```
gluster volume remove-brick test node1:/export/test node2:/export/test
gluster volume stop test
gluster volume remove test
```

- Add a new disk to a volume

```
gluster volume add-brick <volname> replica 2 node3:/export/moretest
```

- Manage access by ip

```
gluster volume set testvol auth.allow 192.168.1.1
# or
gluster volume set testvol auth.allow all
gluster volume set testvol auth.reject 192.168.10.*
```

- How many space to reserve for logs / meta data?

```
gluster volume set cluster.min-free-disk 5%
```

- Enable self healing (on by default)

```
gluster volume set cluster.self-heal-daemon on
```

NFS export

- Start rpcbind
- Start nfslock (rpcstatd)
- Start glusterd
- Adjust firewall

Port	Description
2049	GlusterFS (NFS)
111	RPCbind
54539	RCP statd
38003	RPCbind

- Now you can mount it with

```
mount -t nfs -o mountproto=tcp,vers=3 ip1:/testme /mnt
```

Quota

```
gluster volume quota <volname> enable
gluster volume quota <volname> limit-usage <directory> 10GB
gluster volume quota <volname> list
gluster volume quota <volname> remove <directory>
```

Performance tuning

- Performance information

```
gluster volume top <volname> read-perf
gluster volume top <volname> write-perf
```

- Profiling

```
gluster volume profile <volname> start
gluster volume profile <volname> info
gluster volume profile <volname> stop
```

- Setting read cache size (default 32MB)

```
gluster volume set <volname> performance.cache-size 256MB
```

- Stripe block size

```
gluster volume set cluster.stripe-block-size 128KB
```

- I/O threads

```
gluster volume set performance.io-thread-count 32
```

Troubleshooting

- *requested NFS version or transport protocol is not supported* -> you try to mount with UDP or you didnt start rpcbind, nfslock, glusterd in the right order
- *Protocol not supported* -> you try to mount with version 4 instead of 3
- *node is already part of another cluster* -> delete /var/lib/glusterd/peers/*
- *split brain* means that we detected changes to both replicas

```
gluster volume heal <volname> full
gluster volume heal <volname> info
```

- ‘{path} or a prefix of it is already part of a volume ‘ -> you forgot to remove the brick before deleting the volume

```
setfattr -x trusted.glusterfs.volume-id $brick_path
setfattr -x trusted.gfid $brick_path
rm -rf $brick_path/.glusterfs
service glusterd restart
```

Hadoop

Overview

- Name node is the master or controller of the HDFS (currently only one possible!)
- Data nodes are the storage nodes for data
- Mapred nodes are the computing nodes
- HDFS splits files into file block (128 Mb by default) which are split up into storage blocks (512 kb) and keeps distributed copies (3 by default)

- JobTracker is the controller of Map reduce
- A job consists of a number of tasks
- TaskTracker runs on every data node to receive a computation task
- Hadoop moves the computation to the data instead of vice versa
- `bin/slaves.sh` allows you to execute a command on all slave nodes

Configuration files

- Default values can be looked up in `conf/hadoop-defaults.xml`
- Config options can be combined in `conf/hadoop-site.xml`

File	Description
hadoop-env.sh	Environment variables
hadoop-policy.xml	ACL for various Hadoop services
core-site.xml	Hadoop core settings
hdfs-site.xml	Settings for HDFS: namenode, secondary namenode, datanodes
mapred-site.xml	Settings for MapReduce nodes
masters	Contains the hostname of the SecondaryNameNode
slaves	Lists every ndoe which should start TaskTracker and DataNode daemons

Network ports

- These are the tcp ports to open in your firewall

Port	Description	Config parameter
50070	Name node	dfs.http.address
50075	Data node	dfs.datanode.http.address
50090	Secondary Name node	dfs.secondary.http.address
50030	Job tracker	mapred.job.tracker.http.address
50060	Task tracker	mapred.task.tracker.http.address

Installation

- Install Java (at least 1.6.0!)
- Get Hadoop from http://hadoop.apache.org/common/releases.html#Download
- Unzip it e.g. in /opt
- Edit `conf/hadoop-env.sh` to set environment variables

```
export JAVA_HOME=/usr/lib/jvm/jre-1.5.0-gcj
```

- Edit `conf/core-site.xml` to configure tmp dir and location of name node

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
</property>

<property>
```

```
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
  <description>The name of the default file system.  A URI whose
scheme and authority determine the FileSystem implementation.  The
uri's scheme determines the config property (fs.SCHEME.impl) naming
the FileSystem implementation class.  The uri's authority is used to
determine the host, port, etc. for a filesystem.</description>
</property>
<property>
  <name>hadoop.security.authorization</name>
  <value>true</value>
</property>
```

- Edit `conf/mapred-site.xml` to set the locations of the job tracker and its working dir

```
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs
at.  If "local", then jobs are run in-process as a single map
and reduce task.
  </description>
</property>

<property>
  <name>mapreduce.jobtracker.staging.root.dir</name>
  <value>/user</value>
</property>
```

- Edit `conf/hdfs-site.xml` to set working dirs of name and data node and how often a file gets replicated

```
<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
  </description>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/hadoop/data</value>
</property>
<property>
  <name>dfs.name.dir</name>
  <value>/hadoop/name</value>
</property>
```

- Create a hadoop user with an SSH key

```
useradd -d /opt/hadoop hadoop
chown -R hadoop /opt/hadoop
su - hadoop
ssh-keygen
cat .ssh/id_rsa.pub > .ssh/authorized_keys
chmod 400 .ssh/authorized_keys
ssh localhost
```

- Format the HDFS

```
bin/hadoop namenode -format
```

- Start all servers

```
bin/start-all.sh
```

- Test the installation

```
bin/hadoop jar hadoop-examples-1.2.1.jar pi 2 10
```

Configure HDFS

- Config file is `conf/hdfs-site.xml` or `conf/hadoop-site.xml`

Config option	Description
fs.default.name	The URI for the name node e.g. hdfs://namenode:9000
dfs.data.dir	Directory where data node stores its stuff
dfs.name.dir	Directory where name node stores its stuff
dfs.block.size	Changes the file block size
dfs.namenode.handler.count	Nr of threads for name node to handle data nodes

Working with HDFS

- Access to the name node via http://localhost:50070
- Mkdir

```
hadoop dfs -mkdir some_dir
```

- Copy a file to hdfs

```
hadoop dfs -copyFromLocal file.txt some_dir
hadoop dfs -put file.txt some_dir
```

- Copy a large file in parallel

```
hadoop distcp file:///data/bigfile /some_dir
```

- Copy a directory

```
hadoop fs -copyFromLocal src_dir dst_dir
```

- List a directory

```
hadoop dfs -ls some_dir
```

- Copy a file on HDFS

```
hadoop dfs -cp file.txt test.txt
```

- Remove a file

```
hadoop dfs -rm test.txt
```

- Show file contents

```
hadoop dfs -cat file.txt
```

- Retrieve a file

```
hadoop dfs -get file.txt local_file.txt
```

- Remote access

```
HADOOP_USER_NAME=hadoop bin/hdfs dfs -fs hdfs://192.168.1.4:9000 -ls /
```

- In python

```
cat = subprocess.Popen(["hadoop", "fs", "-cat", "/path/to/myfile"], stdout=subprocess.
↪PIPE)
for line in cat.stdout:
  print line
```

- Rebalance HDFS

```
bin/start-balancer.sh
```

- Check filesystem health

```
bin/hadoof fsck
```

Export HDFS via NFS

- Install like describe on https://github.com/cloudera/hdfs-nfs-proxy/wiki/Quick-Start
- Example config

```
<?xml version="1.0" encoding="UTF-8"?>

<configuration>
  <property>
    <name>hdfs.nfs.nfs4.owner.domain</name>
    <value>localdomain</value>
  </property>
  <property>
    <name>hdfs.nfs.data.dir</name>
    <value>/tmp/hdfs-nfs-proxy/data</value>
  </property>
  <property>
    <name>hdfs.nfs.temp.dirs</name>
    <value>/tmp/hdfs-nfs-proxy/tmp</value>
  </property>
  <property>
    <name>hdfs.nfs.security.flavor</name>
    <value>unix</value>
  </property>
  <property>
    <name>hdfs.nfs.security.allowed.hosts</name>
```

```
    <value>
    * rw
    </value>
  </property>
</configuration>
```

Remove a data node

- Add the node name to `conf/dfs-exclude.txt`
- Edit `conf/hdfs-site.xml` and add the following snippet

```
<property>
  <name>dfs.hosts.exclude</name>
  <value>$HADOOP_HOME/conf/dfs-exclude.txt</value>
</property>
```

- Reload DFS Config

```
hadoop dfsadmin -refreshNodes
```

Remove a task tracker

- Add the node name to `conf/mapred-exclude.txt`
- Edit `conf/mapred-site.xml` and add the following snippet

```
<property>
  <name>mapred.hosts.exclude</name>
  <value>$HADOOP_HOME/conf/mapred-exclude.txt</value>
</property>
```

- Reload DFS Config

```
hadoop mapredadmin -refreshNodes
```

Configure Map Reduce

- Config file is `conf/mapred-site.xml` or `conf/hadoop-site.xml`

Config option	Description
mapred.job.tracker.handler.count	Nr of threads for job tracker to handle task trackers
io.file.buffer.size	Read/write buffer size
io.sort.factor	Number of streams to merge concurrently when sorting files during shuffling
io.sort.mb	Amount of memory to use while sorting data
mapred.reduce.parallel.copies	Number of concurrent connections a reducer should use when fetching its input from mappers
tasktracker.http.threads	Number of threads each TaskTracker uses to provide intermediate map output to reducers
mapred.tasktracker.map.tasks.maximum	Number of map tasks to deploy on each machine
mapred.tasktracker.reduce.tasks.maximum	Number of reduce tasks to deploy on each machine

Streaming interface

- Access the JobTracker with http://localhost:50030
- Access TaskTracker with http://localhost:50060
- Example mapper for word counting (data comes from STDIN and output goes to STDOUT)

```
#!/usr/bin/env python

import sys

for line in sys.stdin:
  line = line.strip()
  words = line.split()

  for word in words:
    # This will be the input for the reduce script
    print '%s\t%s' % (word, 1)
```

- Example reducer code

```
#!/usr/bin/env python

import sys

words = {}

# Gets something like
# word1 1
# word1 1
# word2 1
# word3 1
for line in sys.stdin:
  line = line.strip()
  word, count = line.split('\t', 1)

  try:
    words[word] = words.get(word, 0) + int(count)
  except ValueError:
    pass

for (word, count) in words.items():
  print "%s\t%d" % (word, count)
```

- Execute it with the following command

```
bin/hadoop dfs -mkdir /test
bin/hadoop dfs -put some_file /test
bin/hadoop jar share/hadoop/tools/lib/hadoop-streaming-2.4.1.jar -file /full/path/to/
↪mapper.py -mapper /full/path/to/mapper.py -file /full/path/to/reducer.py -reducer /
↪full/path/to/reducer.py -input /test/README.txt -output /myoutput
```

- Get the result

```
bin/hadoop dfs -cat /myoutput/part-00000
```

Mrjob

- Sample word count

```
from mrjob.job import MRJob

class MRWordFrequencyCount(MRJob):

    def mapper(self, _, line):
        yield "chars", len(line)
        yield "words", len(line.split())
        yield "lines", 1

    def reducer(self, key, values):
        yield key, sum(values)

if __name__ == '__main__':
    MRWordFrequencyCount.run()
```

- Or to grep for errors in kern.log

```
class GrepErrors(MRJob):
    def mapper(self, _, line):
        if "error" in line.lower() or "failure" in line.lower():
            yield "lines", line

    def reducer(self, key, values):
        yield key, "\n".join(values)

if __name__ == '__main__':
    GrepErrors.run()
```

- To run it locally run

```
cat input.txt | python mrjob-example.py
```

- To run it on hadoop call

```
python mrjob-example.py -r hadoop hdfs:///mydir/input.txt
```

Pydoop

- Requires boost-python and maybe boost-devel
- Maybe you need to adjust setup.py to install pydoop (search for `get_java_library_dirs` function and return hardcoded path to libjvm.so)
- Simple wordcount

```
#!/usr/bin/python

def mapper(key, value, writer):
  for word in value.split():
    writer.emit(word, "1")

def reducer(key, value_list, writer):
  writer.emit(key, sum(map(int, value_list)))
```

- Run it with

```
pydoop script test-pydoop.py /test/README.txt myout
```

- Accessing HDFS

```
import pydoop.hdfs as hdfs
for line in hdfs.open("/some/file"):
  print line
```

Jobs

- List running jobs

```
bin/hadoop job -list
```

- List all jobs

```
bin/hadoop job -list all
```

- Terminate a job

```
bin/hadoop job -kill <id>
```

- Get status of a job

```
bin/hadoop job -status <id>
```

- Change priority of a job

```
hadoop job -set-priority <id> HIGH
```

Queues

- List queues

```
hadoop queue -list
```

- List ACLs of a queue

```
hadoop queue -showacls
```

- Show all jobs in a queue

```
hadoop queue -info <queue> -showJobs
```

Security

- This is not for user authentication but for authenticating services!
- You can only adapt user permission by setting `security.client.protocol.acl`
- To enable service-level security set `hadoop.security.authorization` to `true` in `conf/core-site.xml`

Config option	Description
security.client.protocol.acl	You must have these permissions to do anything with the API
security.client.datanode.protocol.acl	ACL for ClientDatanodeProtocol, the client-to-datanode protocol for block recovery.
security.datanode.protocol.acl	ACL for DatanodeProtocol, which is used by datanodes to communicate with the namenode.
security.inter.datanode.protocol.acl	ACL for InterDatanodeProtocol, the inter-datanode protocol for updating generation timestamp.
security.namenode.protocol.acl	ACL for NamenodeProtocol, the protocol used by the secondary namenode to communicate with the namenode.
security.inter.tracker.protocol.acl	ACL for InterTrackerProtocol, used by the tasktrackers to communicate with the jobtracker.
security.job.submission.protocol.acl	ACL for JobSubmissionProtocol, used by job clients to communciate with the jobtracker for job submission, querying job status etc.
security.task.umbilical.protocol.acl	ACL for TaskUmbilicalProtocol, used by the map and reduce tasks to communicate with the parent tasktracker.
security.refresh.policy.protocol.acl	ACL for RefreshAuthorizationPolicyProtocol, used by the dfsadmin and mradmin commands to refresh the security policy in-effect.

- Seems like you have to always add root to security.client.protocol.acl
- After altering the policy you have to refresh it for data and task nodes

```
hadoop dfsadmin -refreshServiceAcl
hadoop mradmin -refreshServiceAcl
```

- HDFS has POSIX-like permissions

```
hadoop dfs -chown
hadoop dfs -chmod
hadoop dfs -chgrp
```

- Network encryption can be setup in Hadoop >= 2.0.2-alpha see http://blog.cloudera.com/blog/2013/03/how-to-set-up-a-hadoop-cluster-with-network-encryption/

Multi-User-Hadoop

- Setup a hadoop group in `conf/hdfs-site.xml`

```
<property>
  <name>dfs.permissions.supergroup</name>
  <value>hadoop</value>
</property>
```

- Set jobtracker staging directory in dfs to other than /

```
<property>
  <name>mapreduce.jobtracker.staging.root.dir</name>
  <value>/user</value>
</property>
```

- Change permissions in hdfs

```
bin/hadoop chgrp -R hadoop /
bin/hadoop chmod 777 /user
```

- Adjust tmp directory permission in real filesystem (do NOT change recursively datanodes will blame you for that!)

```
chmod 777 /app/hadoop/tmp
chmod 777 /app/hadoop/tmp/mapred
```

- Add your users to the hadoop group

Restart a single daemon on a slave node

- Connect to the slave node
- Get hadoop user

```
bin/hadoop-daemon.sh start tasktracker
```

Zookeeper Setup

- Edit *conf/zoo.cfg*

```
tickTime=2000
clientPort=2181
initLimit=5
syncLimit=2
dataDir=/local/hadoop/zookeeper/data
dataLogDir=/local/hadoop/zookeeper/log

# be sure to add an odd number of servers!
server.1=node1:2888:3888
server.2=node2:2888:3888
server.3=node3:2888:3888
```

- Start the server on all nodes

```
bin/zkServer.sh start
```

- To test the connection

```
bin/zkCli.sh -server 127.0.0.1:2181
```

- If the slave server wont start up check `myid` file in zookeeper data dir. It must be the same as in `zoo.cfg`
- To get some status about the zookeeper cluster run

```
echo stat | nc 127.0.0.1 2181
```

- To get a shell on a zookeeper cluster do the following

```
cd zookeeper/src/c
./configure
make
./cli_st 127.0.0.1:2181
help
```

- Zookeeper logfile may be in same directory as it was started from

HBase Setup

- Make sure zookeeper is installed
- Edit *conf/hbase-site.xml*

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:54310/hbase</value>
  </property>

  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>

  <property>
    <name>hbase.tmp.dir</name>
   <value>/local/hadoop/hbase</value>
  </property>

  <property>
   <name>hbase.ZooKeeper.quorum</name>
   <value>localhost</value>
  </property>

  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/local/hadoop/zookeeper</value>
  </property>

</configuration>
```

- Edit *conf/regionservers* and add all nodes
- Edit *conf/hbase-env.sh* and set *JAVA_HOME*
- Start HBase server and shell

```
bin/start-hbase.sh
bin/hbase shell
```

- Web interface can be found here *http://localhost:60010*
- If it complains about Class not found make sure to copy some required libs

```
cp /opt/hadoop/hadoop-core*.jar /opt/hadoop/lib/commons-*.jar /opt/zookeeper/
↪zookeeper-*.jar /opt/hbase/lib
```

- If it complains about address already in use in hbase-zookeeper log, set

```
export HBASE_MANAGES_ZK=false
```

Working with HBase

- Create a table (cf is the columfamily)

```
create 'tablename', 'cf'
create 'webtable', 'contents', 'anchors'
```

- Show all tables

```
list
describe 'tablename'
```

- Insert values (can only put 1 value in 1 column at a time!)

```
put 'tablename', 'row index', 'cf:col1', 'value1'
put 'webtable', 'de.codekid.www', 'contents:html', '<html><body>blah blah</body></
↪html>'
put 'webtable', 'de.codekid.www', 'anchors:www.ccc.de', 'Chaos Computer Club'
put 'webtable', 'de.codekid.www', 'anchors:www.chaostal.de', 'Chaostal'
```

- Select values

```
get 'tablename' 'row index'
get 'webtable', 'de.codekid.www'
```

- Check table health

```
scan 'tablename'
```

- Drop a table

```
disable 'tablename'
drop 'tablename'
```

- For more see http://learnhbase.wordpress.com/2013/03/02/hbase-shell-commands/
- REST interface

```
bin/hbase rest start
curl -v -X GET -H "Accept: application/json" 'http://localhost:8080/webtable/de.
↪codekid.www'""
```

- REST with Python (http://blog.cloudera.com/blog/2013/10/hello-starbase-a-python-wrapper-for-the-hbase-rest-api/)

```
#!/usr/bin/python

from starbase import Connection

table = 'webtable'
key = 'de.codekid.www'
column = 'anchors:images.datenterrorist.de'
data = 'Galerie'

c = Connection(host='127.0.0.1', port=8080)
t = c.table(table)
t.insert(key, {column: data})
print t.fetch(key)
rf = '{"type": "RowFilter", "op": "EQUAL", "comparator": {"type":
↪"RegexStringComparator", "value": "^row_1.+"}}'

for row in t.fetch_all_rows(with_row_id=True, filter_string=rf):
  print row
```

- Complete example

```
from BeautifulSoup import BeautifulSoup
from urlparse import urlparse
from urllib2 import urlopen
from starbase import Connection
from mrjob.job import MRJob
import sys

class MRWebCrawler(MRJob):
    def prepare_link(self, link):
        link_url = link
      scheme = urlparse(link_url)[0]

      if not scheme:
          parsed_base_url = urlparse(self.base_url)
          link_url = parsed_base_url[0] + "://" + parsed_base_url[1] + "/" + link_url

        return str(link_url)

    def mapper(self, _, base_url):
      self.base_url = base_url
      table = 'webtable'
      host = urlparse(base_url)[1]
      html = urlopen(base_url).read()
      parser = BeautifulSoup(html)

      conn = Connection(host='127.0.0.1', port=8080)
      table = conn.table(table)
      table.insert(host, {'contents:html': html})


      for link in parser('a'):
          if link.get('href'):
              if len(link.contents[0]) > 1:
                  table.insert(host, {'anchor:' + link.contents[0]: self.prepare_
↪link(link.get('href'))})
              else:
                  for tag in link.contents:
                      if hasattr(tag, 'get') and tag.get('alt'):
                          table.insert(host, {'anchor:' + tag.get('alt'): self.
↪prepare_link(link.get('href'))})
                          break
                      elif hasattr(tag, 'get') and  tag.get('title'):
                          table.insert(host, {'anchor:' + tag.get('title'): self.
↪prepare_link(link.get('href'))})
                          break

if __name__ == '__main__':
    MRWebCrawler.run()

sys.exit(0)
```

Hbase troubleshooting

```
bin/hbase org.apache.hadoop.hbase.util.hbck.OfflineMetaRepair
```

Working with Hive

- Create a table in HBase

```
CREATE TABLE people(key int, name string, age int) STORED BY 'org.apache.hadoop.hive.
↪hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,
↪name:val,age:val") TBLPROPERTIES ("hbase.table.name" = "hive_people");
```

- List / describe tables

```
SHOW TABLES;
DESCRIBE <tablename>
```

- Insert data (format is 1|ballel|31)

```
LOAD DATA LOCAL INPATH 'people.txt' OVERWRITE INTO TABLE people
```

Import data from a database to HDFS

```
bin/sqoop import -m 1 --connect jdbc:mysql://<host>:<port>/dbname --username <dbuser>␣
↪--password <dbpass> --table <tablename> --target-dir </hdfs-dir>
```

Export data from HDFS to a database

```
bin/sqoop export -m 1 --connect jdbc:mysql://<host>:<port>/dbname --username <dbuser>␣
↪--password <dbpass> --table <tablename> --export-dir </hdfs-dir>
```

Addons

- *Hive* <*http://hive.apache.org*> - A SQL-like language to produce map-reduce jobs
- *Pig* <*http://pig.apache.org*> - high-level mapreduce language
- *oozie* <*http://oozie.apache.org*> - job scheduling
- *flume* <*http://flume.apache.org*> - log and data aggregation
- *whirr* <*http://whirr.apache.org*> - automated cloud clusters on ec2, rackspace etc
- *sqoop* <*http://sqoop.apache.org*> - relational data import
- *hbase* <*http://hbase.apache.org*> - realtime processing (based on google bigtable)
- *accumulo* <*http://accumulo.apache.org*> - NSA fork of HBase
- *mahout* <*http://mahout.apache.org*> - machine learning libraries
- *trumpet* <*http://verisign.github.io/trumpet/*> - Trumpet is an highly-available, fault-tolerant, non intrusive and scalable INotify-like building block for Hadoop HDFS.

Documentation

- http://developer.yahoo.com/hadoop/tutorial/
- https://www.youtube.com/watch?v=XtLXPLb6EXs
- http://hadoop.apache.org/docs/stable/commands_manual.pdf
- http://www.wdong.org/wordpress/blog/2015/01/08/hadoop-internals-how-to-manually-assemble-a-file-in-hdfs/
- http://www.aosabook.org/en/hdfs.html - Internals of HDFS

Troubleshooting

- Check all daemons are running

```
jps
```

- Get a list of active task trackers

```
bin/hadoop job -list-active-trackers
```

- Check DFS status

```
bin/hadoop dfsadmin -report
bin/hadoop fschk /
```

- Cannot create directory Name node is in safe mode -> NameNode is in safemode until configured percent of blocks reported to be online by the data nodes.
- DFS not leaving safe mode?

```
bin/hadoop dfsadmin -safemode leave
```

- Start name and data node in foreground

```
bin/hadoop --config conf namenode
bin/hadoop --config conf datanode
```

- java.io.IOException: Incompatible namespaceIDs (namenode was reformated but datanodes not) -> first try to manually update the namespaceID on every data node by editing `/local/hadoop/data-node/current/VERSION` if this doesnt help

```
bin/stop-all.sh
rm -rf /local/hadoop # on all datanodes
bin/hadoop namenode -format
bin/start-all.sh
```

- Zookeeper status

```
cd /opt/zookeeper
bin/zkCli.sh -server <master-node>:2181
[zk: mynode:2181(CONNECTED) 1] ls /
[zk: mynode:2181(CONNECTED) 1] quit
```

- Be sure you have an odd number of server in zoo.cfg
- Be sure there is an zookeeper id

```
cat /local/hadoop/zookeeper/data/myid
```

- Try to start zookeeper in foreground

```
/opt/zookeeper/bin/zkServer.sh start-foreground
```

- HBase status

```
cd /opt/hbase
bin/hbase shell
hbase(main):001:0> list
hbase(main):002:0> status
```

- Read http://hbase.apache.org/book/trouble.html if the error is not one of the below
- ERROR: org.apache.hadoop.hbase.MasterNotRunningException -> Check that HMaster process is running

```
jps
```

Hadoop 2

Overview

Port	Description
50070	hadoop namenode web
54310	hadoop namenode
50010	hadoop datanode
50020	hadoop datanode
50075	hadoop datanode
8030	hadoop resourcemanager
8031	hadoop resourcemanager
8032	hadoop resourcemanager
8033	hadoop resourcemanager
8088	hadoop resourcemanager web
13562	hadoop nodemanager
8040	hadoop nodemanager
34568	hadoop nodemanager
8042	hadoop nodemanager
19888	hadoop job history web
10020	hadoop job history
10033	hadoop job history

Installation

- Edit `etc/hadoop/core-site.xml` to configure tmp dir and location of name node

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/local/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
</property>
```

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://127.0.0.1:54310</value>
</property>
<property>
  <name>hadoop.security.authorization</name>
  <value>true</value>
</property>
<property>
  <name>hadoop.http.staticuser.user</name>
  <value>hdfs</value>
</property>
```

- Edit `etc/hadoop/mapred-site.xml` to set the locations of the job tracker and its working dir

```
<property>
  <name>mapred.job.tracker</name>
  <value>127.0.0.1:54311</value>
</property>
<property>
  <name>mapreduce.jobtracker.staging.root.dir</name>
  <value>/user</value>
</property>
<property>
  <name>mapred.tasktracker.map.tasks.maximum</name>
  <value>16</value>
</property>
<property>
  <name>mapred.tasktracker.reduce.tasks.maximum</name>
  <value>16</value>
</property>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>

<!--
     DEFAULT VALUES LIKELY TO GET ADJUSTED BELOW HERE
-->
<property>
  <name>mapreduce.jobtracker.expire.trackers.interval</name>
  <value>600000</value>
  <description>Expert: The time-interval, in miliseconds, after which
  a tasktracker is declared 'lost' if it doesn't send heartbeats.
  </description>
</property>

<property>
  <name>mapreduce.jobtracker.restart.recover</name>
  <value>false</value>
  <description>"true" to enable (job) recovery upon restart,
               "false" to start afresh
  </description>
</property>

<property>
  <name>mapreduce.task.io.sort.factor</name>
  <value>10</value>
```

```
  <description>The number of streams to merge at once while sorting
  files.  This determines the number of open file handles.</description>
</property>

<property>
  <name>mapreduce.task.io.sort.mb</name>
  <value>100</value>
  <description>The total amount of buffer memory to use while sorting
  files, in megabytes.  By default, gives each merge stream 1MB, which
  should minimize seeks.</description>
</property>

<property>
  <name>mapreduce.tasktracker.http.threads</name>
  <value>40</value>
  <description>The number of worker threads that for the http server. This is
             used for map output fetching
  </description>
</property>

<property>
  <name>mapreduce.task.timeout</name>
  <value>600000</value>
  <description>The number of milliseconds before a task will be
  terminated if it neither reads an input, writes an output, nor
  updates its status string.  A value of 0 disables the timeout.
  </description>
</property>

<property>
 <name>mapreduce.task.tmp.dir</name>
 <value>./tmp</value>
 <description> To set the value of tmp directory for map and reduce tasks.
 If the value is an absolute path, it is directly assigned. Otherwise, it is
 prepended with task's working directory. The java tasks are executed with
  option -Djava.io.tmpdir='the absolute path of the tmp dir'. Pipes and
  streaming are set with environment variable,
  TMPDIR='the absolute path of the tmp dir'
  </description>
</property>

<property>
  <name>mapreduce.output.fileoutputformat.compress</name>
  <value>false</value>
  <description>Should the job outputs be compressed?
  </description>
</property>

<property>
  <name>mapreduce.shuffle.ssl.enabled</name>
  <value>false</value>
  <description>
  Whether to use SSL for for the Shuffle HTTP endpoints.
  </description>
</property>
```

- Edit `etc/hadoop/hdfs-site.xml` to set working dirs of name and data node and how often a file gets replicated

```
 <property>
   <name>dfs.replication</name>
   <value>3</value>
 </property>
 <property>
   <name>dfs.data.dir</name>
   <value>/data/hadoop/data-node</value>
 </property>
 <property>
   <name>dfs.name.dir</name>
   <value>/data/hadoop/name-node</value>
 </property>
 <property>
   <name>dfs.permissions.supergroup</name>
   <value>hadoop</value>
 </property>

   <property>
     <name>dfs.namenode.accesstime.precision</name>
     <value>3600000</value>
     <description>The access time for HDFS file is precise upto this value.
   The default value is 1 hour. Setting a value of 0 disables
   access times for HDFS.
     </description>
   </property>

<!--
      DEFAULT VALUES LIKELY TO GET ADJUSTED BELOW HERE
 -->
 <property>
   <name>dfs.permissions.enabled</name>
   <value>true</value>
   <description>
   If "true", enable permission checking in HDFS.
   If "false", permission checking is turned off,
   but all other behavior is unchanged.
   Switching from one parameter value to the other does not change the mode,
   owner or group of files or directories.
   </description>
 </property>

 <property>
   <name>dfs.namenode.fs-limits.min-block-size</name>
   <value>1048576</value>
   <description>Minimum block size in bytes, enforced by the Namenode at create
     time. This prevents the accidental creation of files with tiny block
     sizes (and thus many blocks), which can degrade
     performance.</description>
 </property>

 <property>
   <name>dfs.blocksize</name>
   <value>134217728</value>
   <description>
     The default block size for new files, in bytes.
     You can use the following suffix (case insensitive):
     k(kilo), m(mega), g(giga), t(tera), p(peta), e(exa) to specify the size (such as␣
↪128k, 512m, 1g, etc.),
```

```
    Or provide complete size in bytes (such as 134217728 for 128 MB).
  </description>
</property>

<property>
  <name>dfs.namenode.fs-limits.max-blocks-per-file</name>
  <value>1048576</value>
  <description>Maximum number of blocks per file, enforced by the Namenode on
      write. This prevents the creation of extremely large files which can
      degrade performance.</description>
</property>

<property>
  <name>dfs.heartbeat.interval</name>
  <value>3</value>
  <description>Determines datanode heartbeat interval in seconds.</description>
</property>

<property>
  <name>dfs.namenode.handler.count</name>
  <value>10</value>
  <description>The number of server threads for the namenode.</description>
</property>

<property>
  <name>dfs.namenode.name.dir.restore</name>
  <value>false</value>
  <description>Set to true to enable NameNode to attempt recovering a
    previously failed dfs.namenode.name.dir. When enabled, a recovery of any
    failed directory is attempted during checkpoint.</description>
</property>

<property>
  <name>dfs.image.compress</name>
  <value>false</value>
  <description>Should the dfs image be compressed?
  </description>
</property>

<property>
  <name>dfs.image.transfer.bandwidthPerSec</name>
  <value>0</value>
  <description>
      Maximum bandwidth used for image transfer in bytes per second.
      This can help keep normal namenode operations responsive during
      checkpointing. The maximum bandwidth and timeout in
      dfs.image.transfer.timeout should be set such that normal image
      transfers can complete successfully.
      A default value of 0 indicates that throttling is disabled.
  </description>
</property>

<property>
  <name>dfs.datanode.max.transfer.threads</name>
  <value>4096</value>
  <description>
      Specifies the maximum number of threads to use for transferring data
      in and out of the DN.
```

```
    </description>
  </property>

  <property>
    <name>dfs.ha.automatic-failover.enabled</name>
    <value>false</value>
    <description>
    Whether automatic failover is enabled. See the HDFS High
    Availability documentation for details on automatic HA
    configuration.
    </description>
  </property>

  <property>
    <name>dfs.webhdfs.enabled</name>
    <value>false</value>
    <description>
    Enable WebHDFS (REST API) in Namenodes and Datanodes.
    </description>
  </property>

  <property>
    <name>dfs.https.enable</name>
    <value>false</value>
    <description>Decide if HTTPS(SSL) is supported on HDFS
    </description>
  </property>
```

- Edit `etc/hadoop/yarn-site.xml`

```
<property>
  <name>yarn.resourcemanager.resource-tracker.address</name>
  <value>[% HADOOP_MASTER %]:8031</value>
  <description>host is the hostname of the resource manager and
  port is the port on which the NodeManagers contact the Resource Manager.
  </description>
</property>

<property>
  <name>yarn.resourcemanager.scheduler.address</name>
  <value>127.0.0.1:8030</value>
  <description>host is the hostname of the resourcemanager and port is the port
  on which the Applications in the cluster talk to the Resource Manager.
  </description>
</property>

<property>
  <name>yarn.resourcemanager.scheduler.class</name>
  <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.
↪CapacityScheduler</value>
  <description>In case you do not want to use the default scheduler</description>
</property>

<property>
  <name>yarn.nodemanager.local-dirs</name>
  <value>/data/hadoop/nm</value>
  <description>the local directories used by the nodemanager</description>
</property>
```

```
<property>
  <name>yarn.nodemanager.address</name>
  <value>127.0.0.1:8040</value>
  <description>the nodemanagers bind to this port</description>
</property>

<property>
  <name>yarn.nodemanager.resource.memory-mb</name>
  <value>10240</value>
  <description>the amount of memory on the NodeManager in GB</description>
</property>

<property>
  <name>yarn.nodemanager.remote-app-log-dir</name>
  <value>/app-logs</value>
  <description>directory on hdfs where the application logs are moved to </
↪description>
</property>

 <property>
  <name>yarn.nodemanager.log-dirs</name>
  <value></value>
  <description>the directories used by Nodemanagers as log directories</description>
</property>

<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
  <description>shuffle service that needs to be set for Map Reduce to run </
↪description>
</property>
```

- Create a hadoop user with an SSH key

```
useradd -d /opt/hadoop hadoop
chown -R hadoop /opt/hadoop
su - hadoop
ssh-keygen
cat .ssh/id_rsa.pub > .ssh/authorized_keys
chmod 400 .ssh/authorized_keys
ssh localhost
```

- Format the HDFS

```
su - hadoop -c '/opt/hadoop/bin/hdfs namenode -format -force'
```

- Start the services

```
su - hadoop -c '/opt/hadoop/sbin/hadoop-daemon.sh start namenode && /opt/hadoop/sbin/
↪hadoop-daemon.sh start datanode' && /opt/hadoop/sbin/yarn-daemon.sh start␣
↪resourcemanager && /opt/hadoop/sbin/yarn-daemon.sh start nodemanager'"
```

Check status

- HDFS

```
/opt/hadoop/bin/hdfs dfsadmin -report
```

- YARN

```
/opt/hadoop/bin/yarn node -list
```

- Test Namenode

```
su - hadoop -c '/opt/hadoop/bin/hadoop fs -mkdir /user'
su - hadoop -c '/opt/hadoop/bin/hadoop fs -mkdir /user/hadoop'
```

- Test Datanode

```
su - hadoop -c '/opt/hadoop/bin/hadoop fs -put /opt/hadoop/etc/hadoop/hadoop-env.sh /
↪user/hadoop/hadoop-env'
```

- Test YARN

```
su - hadoop -c "/opt/hadoop/bin/hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-
↪mapreduce-examples-*.jar pi 2 10"
```

Configure High Availability

- Edit core-site.xml

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://mycluster</value>
</property>
<property>
  <name>ha.zookeeper.quorum</name>
  <value>zookeeper1:2181,zookeeper2:2181,zookeeper3:2181</value>
</property>
```

- Edit hdfs-site.xml

```
<!-- only for single node setup
<property>
  <name>dfs.name.dir</name>
  <value>file:///local/hadoop/name-node</value>
  </property>
-->

<!-- HA settings -->
<property>
  <name>dfs.nameservices</name>
  <value>mycluster</value>
</property>

<property>
  <name>dfs.ha.namenodes.mycluster</name>
  <value>nn1,nn2</value>
</property>

<property>
  <name>dfs.namenode.rpc-address.mycluster.nn1</name>
```

```
  <value>hadoop_master:8020</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.mycluster.nn2</name>
  <value>hadoop_master2:8020</value>
</property>

<property>
  <name>dfs.namenode.http-address.mycluster.nn1</name>
  <value>hadoop_master:50070</value>
</property>
<property>
  <name>dfs.namenode.http-address.mycluster.nn2</name>
  <value>hadoop_master2:50070</value>
</property>

<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>qjournal://hadoop_master:8485;hadoop_master2:8485/mycluster</value>
</property>

<property>
  <name>dfs.client.failover.proxy.provider.mycluster</name>
  <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</
↪value>
</property>

<property>
  <name>dfs.ha.fencing.methods</name>
  <value>sshfence</value>
</property>

<property>
  <name>dfs.ha.fencing.ssh.private-key-files</name>
  <value>/opt/hadoop/.ssh/id_rsa</value>
</property>

<property>
  <name>dfs.ha.automatic-failover.enabled</name>
  <value>true</value>
  <description>
  Whether automatic failover is enabled. See the HDFS High
  Availability documentation for details on automatic HA
  configuration.
  </description>
</property>

<property>
  <name>ha.zookeeper.quorum</name>
  <value>zookeeper1:2181,zookeeper2:2181,zookeeper3:2181</value>
</property>
```

- Edit yarn-site.xml

```
<property>
  <name>yarn.resourcemanager.ha.enabled</name>
  <value>true</value>
</property>
```

```
<property>
  <name>yarn.resourcemanager.cluster-id</name>
  <value>mycluster</value>
</property>
<property>
  <name>yarn.resourcemanager.ha.rm-ids</name>
  <value>rm1,rm2</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname.rm1</name>
  <value>hadoop_master</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname.rm2</name>
  <value>hadoop_master2</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address.rm1</name>
  <value>hadoop_master:8088</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address.rm2</name>
  <value>hadoop_master2:8088</value>
</property>
<property>
  <name>yarn.resourcemanager.zk-address</name>
  <value>zookeeper1:2181,zookeeper2:2181,zookeeper3:2181</value>
</property>
```

- Start Zookeeper Fencing Service additionally to normal Zookeeper

```
/opt/hadoop/sbin/hadoop-daemon.sh --script /opt/hadoop/bin/hdfs start zkfc
```

- Start HDFS Journalnode on Namenode servers

```
/opt/hadoop/sbin/hadoop-daemon.sh start journalnode
```

- Initialize and format Namenode

```
/opt/hadoop/bin/hdfs zkfc -formatZK
/opt/hadoop/bin/hdfs namenode -format -force
/opt/hadoop/sbin/hadoop-daemon.sh start namenode
```

- Check the status

```
bin/hdfs haadmin -getServiceState nn1
bin/hdfs haadmin -getServiceState nn2
bin/yarn rmadmin -getServiceState rm1
bin/yarn rmadmin -getServiceState rm2
```

Convert single namenode to HA

```
/opt/hadoop/bin/hdfs namenode -bootstrapStandby
/opt/hadoop/bin/hdfs namenode -initializeSharedEdits
```

Configure Capacity Scheduler

- The CapacityScheduler is designed to allow sharing a large cluster while giving each organization a minimum capacity guarantee.
- Make sure `org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler` is set as `yarn.resourcemanager.scheduler.class` in `etc/hadoop/yarn-site.xml`
- Configure resources for unix groups a, b and default
- Edit `etc/hadoop/capacity-scheduler.xml`

```
<property>
  <name>yarn.scheduler.capacity.root.queues</name>
  <value>a,b,default</value>
  <description>The queues at the this level (root is the root queue).
  </description>
</property>

<!-- GROUP A -->
<property>
  <name>yarn.scheduler.capacity.root.a.capacity</name>
  <value>30</value>
  <description>Default queue target capacity.</description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.a.user-limit-factor</name>
  <value>1</value>
  <description>
    Default queue user limit a percentage from 0.0 to 1.0.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.a.maximum-capacity</name>
  <value>100</value>
  <description>
    The maximum capacity of the default queue.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.a.state</name>
  <value>RUNNING</value>
  <description>
    The state of the default queue. State can be one of RUNNING or STOPPED.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.a.acl_submit_applications</name>
  <value>group_a</value>
  <description>
    The ACL of who can submit jobs to the default queue.
  </description>
</property>
```

```
<!-- GROUP B -->
<property>
  <name>yarn.scheduler.capacity.root.b.capacity</name>
  <value>30</value>
  <description>Default queue target capacity.</description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.b.user-limit-factor</name>
  <value>1</value>
  <description>
    Default queue user limit a percentage from 0.0 to 1.0.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.b.maximum-capacity</name>
  <value>100</value>
  <description>
    The maximum capacity of the default queue.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.b.state</name>
  <value>RUNNING</value>
  <description>
    The state of the default queue. State can be one of RUNNING or STOPPED.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.b.acl_submit_applications</name>
  <value>group_b</value>
  <description>
    The ACL of who can submit jobs to the default queue.
  </description>
</property>


<!-- GROUP DEFAULT -->
<property>
  <name>yarn.scheduler.capacity.root.default.capacity</name>
  <value>40</value>
  <description>Default queue target capacity.</description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.default.user-limit-factor</name>
  <value>1</value>
  <description>
    Default queue user limit a percentage from 0.0 to 1.0.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.default.maximum-capacity</name>
```

```
  <value>100</value>
  <description>
    The maximum capacity of the default queue.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.default.state</name>
  <value>RUNNING</value>
  <description>
    The state of the default queue. State can be one of RUNNING or STOPPED.
  </description>
</property>

<property>
  <name>yarn.scheduler.capacity.root.default.acl_submit_applications</name>
  <value>*</value>
  <description>
    The ACL of who can submit jobs to the default queue.
  </description>
</property>
```

- Refresh queues

```
bin/yarn rmadmin -refreshQueues
bin/hadoop queue -list
```

- Submit a test job

```
bin/hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar pi -
↪Dmapred.job.queue.name=a 2 10
```

Configure Fair Scheduler

- All jobs get, on average, an equal share of resources over time
- Make sure `org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler` is set as `yarn.resourcemanager.scheduler.class` in `etc/hadoop/yarn-site.xml`
- A pool has the same name as the user
- Create file `etc/hadoop/fair-scheduler.xml`

```
<?xml version="1.0"?>

<allocations>
  <pool name="hadoop">
    <minMaps>5</minMaps>
    <maxMaps>90</maxMaps>
    <maxReduces>20</maxReduces>
    <weight>2.0</weight>
  </pool>
  <user name="hadoop">
    <maxRunningJobs>3</maxRunningJobs>
  </user>
```

```
  <userMaxJobsDefault>2</userMaxJobsDefault>
</allocations>
```

- Restart resource manager

HDFS NFS Gateway

- Edit `hdfs-site.xml`

```
<property>
  <name>dfs.nfs3.dump.dir</name>
  <value>/data/hadoop/.hdfs-nfs</value>
</property>

<property>
  <name>dfs.nfs.exports.allowed.hosts</name>
  <value>* rw</value>
</property>
```

- Edit `core-site.xml`

```
<property>
  <name>hadoop.proxyuser.nfsserver.groups</name>
  <value>*</value>
  <description>
         The 'nfsserver' user is allowed to proxy all members of the 'nfs-users1' and
         'nfs-users2' groups. Set this to '*' to allow nfsserver user to proxy any␣
↪group.
  </description>
</property>

<property>
  <name>hadoop.proxyuser.nfsserver.hosts</name>
  <value>*</value>
  <description>
         This is the host where the nfs gateway is running. Set this to '*' to allow
         requests from any hosts to be proxied.
  </description>
</property>
```

- Start the daemons

```
systemctl stop rpcbind
sbin/hadoop-daemon.sh start portmap
su - nfsserver -c "sbin/hadoop-daemon.sh start nfs3"
```

- Test nfs mount

```
rpcinfo -p localhost
showmount -e
mount -t nfs -o vers=3,proto=tcp,nolock localhost:/ /mnt
```

Troubleshooting

- `java.lang.IllegalArgumentException: Illegal capacity of -1.0 for queue ->`

You dont have defined a capacity for the queue like yarn.scheduler.capacity.root.$QUEUENAME.capacity

- `org.apache.hadoop.util.Shell$ExitCodeException: chmod: cannot access `/user/myuser1544460269/.staging/job_local1544460269_0001': No such file or directory` -> set mapreduce.framework.name to yarn in mapred-site.xml
- datanode says `Initialization failed for Block pool` -> Namenode got formated / changed afterwards, delete the contents of dfs.data.dir

MPI

Overview

- MPI is an parallel programming API to execute processes over multiple cores / computers
- You must install mpich2 and run `mpd`
- Remote processes are usually executed via ssh

Source sample

- pip install mpi4py
- Run with `mpiexec -n 4 ./test.py`

```
#!/usr/bin/python

from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

print "Hello world from process ", rank, " of ", comm.Get_size(), " processes"

if rank == 1:
  data = [1, 5, 7, 13]
  comm.send(data, dest=0, tag=11)
  print rank, " sending data to process 0"

elif rank == 0:
  data = comm.recv(source=1, tag=11)
  print rank, " got ", data
  comm.bcast("thanks for all the fish!", root=0)
```

Openstack

Overview

- Tenant is something like an organization or mandant (can have multiple users) –> In new versions called project

Subsystem	Ports	Description
Keystone	5000, 35357	Identity service (manages user, roles, tenants, services and endpoints) Port 5000 for auth, 35357 for service registry
Glance	9191, 9292	Image service (manages kvm, qemu, vmware, amazon s3 etc images)
Nova		Compute service (manage startup / life of virtual machines using libvirt)
Nova Scheduler	59229	Decide where to create a new instance
Nova API	8773, 8774, 8775	Access Nova functionality
Nova Network		Configure network if Neutron is not in use
Nova Compute		Management controller for virtual machines
Nova Conductor		Encapsulate database api for nova
Nova ObjectStore	3333	File-based Storage system (can be replaced by Swift or Ceph)
Nova Cert		Nova CA service for x509 certificates
Nova Console		VNC access support
Nova Consoleauth		VNC authorization
Cinder	8776	Volumne service (manages additional storage and replication via LVM / iSCSI / Ceph)
Neutron		Network configuration service - install network client on each vm
Swift		Distributed File Storage Service (can be used to store images for Glance)
Ceph		Ceph is a distributed storage system (object store, block store and POSIX-compliant distributed file system)
Horizon	80	Admin webfrontend in Django

- How an instances is started http://ilearnstack.wordpress.com/2013/04/26/request-flow-for-provisioning-instance-in-openstack/?preview=true&preview_id=410&preview_nonce=3618fe51b7

/D

Installation

- Out-of-the-box http://openstack.redhat.com/Quickstart

```
packstack --allinone

# multi node setup with nova network
packstack --install-hosts=node1,node2,node3 --os-neutron-install=n

# alternativly generate an answer file and edit it
packstack --gen-answer-file /root/answers.txt

# iterative changes (edit answerfile in home dir)
packstack --answer-file=<answerfile>
```

- Manually http://docs.openstack.org/install/
- By using puppet https://wiki.openstack.org/wiki/Puppet-openstack

Configfile to source

```
export OS_USERNAME=admin
export OS_PASSWORD=<whatever>
export OS_TENANT_NAME=<whatever>
export OS_AUTH_URL=http://127.0.0.1:35357/v2.0
```

User management

- Either source configfile above or use *–token <your_secret> –endpoint http://127.0.0.1:35357/v2.0/*
- List

```
keystone tenant-list
keystone user-list
```

- Create

```
keystone user-create --name USERNAME --pass PASSWORD
keystone user-role-add --user-id <user_id> --role-id <role_id> --tenant-id <tenant_id>
```

- The privileges of a role are defined in `/etc/keystone/policy.json`

Create images

- Install your system with libvirt
- Install cloud-init
- Take the disk image
- For more information http://docs.openstack.org/image-guide/content/centos-image.html

Adding images

- List

```
glance image-list
```

- Create

```
glance image-create --name="arch linux" --is-public true --disk-format raw --
↪container-format bare --file "arch_linux.img"
```

- Share an Image with another tenant (–can-share defines it can be reshared)

```
glance member-create --can-share <image> <tenant>
```

- Download an image (e.g. for testing purpose)

```
glance image-download <image>
```

Flavors

- List

```
nova flavor-list
```

- Create

```
nova flavor-create &lt;name&gt; &lt;id&gt; &lt;ram&gt; &lt;disk&gt; &lt;vcpus&gt;
```

Host Aggregates

- Group hypervisors and assign metadata to it to combine it with a flavor so you can start e.g. some vms on monster machines and some on slow ones
- Create a new group

```
nova aggregate-create <name>
nova aggregate-add-host <group_name> <hypervisor>
nova aggregate-list
nova aggregate-details <group_name>
```

- Assign metadata to group

```
nova aggregate-add-metadata <group_name> key=value (e.g. highspec=1)
```

- Assign metadata to flavor

```
nova flavor-key <flavor> set highspec=true
```

- To isolate tenants in a certain host aggregation use `AggregateMultiTenancyIsolation` as `scheduler_default_filters` in `/etc/nova/nova.conf` and set metadata `filter_tenant_id=<tenant_id>` to your aggregation

```
nova aggregate-add-metadata <group_name> filter_tenant_id=<tenant_id>
```

Cells (untested)

- Seperate compute nodes into independent groups with its own db, amqp, network and scheduler servers which share single services like nova-api, keystone, glance, cinder, ceilometer and heat
- Useful to avoid clustering amqp and db servers if load gets to high on very large deployments
- Activated in `/etc/nova/nova.conf` in section `[cells]`

```
[cells]
enable = true
name = MyCellName
```

Configure networking (old style nova networking)

- FlatManager only connects vms to bridge device *no ip configuration!*
- FlatDHCPManager configure network ip on bridge and starts dnsmasq dhcp server on that ip

- VlanManager creates separate VLANs for each tenant
- http://www.mirantis.com/blog/openstack-networking-flatmanager-and-flatdhcpmanager/
- Configure network in */etc/nova/nova.conf*
- flat_network_bridge - bridge interface
- flat_interface - where bridge ends up
- public_interface - used for natting floating (public) ips to private (fixed) ips

```
network_manager=nova.network.manager.FlatDHCPManager
fixed_range=192.168.100.0/24
public_interface=eth0
flat_interface=eth0
flat_network_bridge=br100
```

- Check network settings

```
nova-manage network list
```

- Setup floating ip range manually

```
nova-manage floating create --pool=nova --ip_range=10.10.100.0/24
```

- To automatically assign floating ip add the following to nova.conf

```
auto_assign_floating_ip=True
```

- For manually assigning a floating ip to a vm

```
nova floating-ip-create
nova add-floating-ip <machine_id> <ip_address>
```

Configure Neutron

- Most of the time based on Open vSwitch (http://openvswitch.org/)
- Uses network namespaces and gre tunnel or vlan to seperate tenants (projects)
- You need an interface for host and one for neutron
- Flat network is like nova network flat dhcp network (doesnt seperate tenants)
- Create a new network and subnet

```
neutron net-create <name>
neutron subnet-create --name bastiSubnet --no-gateway --host-route destination=0.0.0.
↪0/0,nexthop=10.10.1.1 --dns-nameserver 8.8.8.8 <net_uuid> 10.10.1.0/24
```

- List existing networks

```
neutron net-list
```

- Get ips / mac of vms

```
neutron port-list
```

- Routing between two nets

```
neutron router-create <name>
neutron router-interface-add <router_name> <net_name_1>
neutron router-interface-add <router_name> <net_name_2>
neutron router-list
```

- Delete an interface from a router

```
neutron router-interface-delete <router_name> <net_name>
```

- Create a floating net

```
neutron net-create --router:external=True floatingNet
neutron subnet-create --name floatingNet --allocation-pool start=192.168.1.2,end=192.
↪168.1.100 --enable_dhcp=False floatingNet 192.168.1.0/24
neutron router-gateway-set <router_name> floatingNet
```

- Find agent hosting a network

```
neutron dhcp-agent-list-hosting-net <net_name>
```

- Find network namespace of a vm

```
nova show <vm_id> # get tenant id
neutron net-list --tenant-id <tenant_id>
neutron dhcp-agent-list-hosting-net <net_name> # find host where net is served
ip netns exec <net_id> # on serving host
```

- Find fixed ips for tenant

```
neutron port-list -f csv -c fixed_ips --tenant_id <tenant_id> | grep subnet | cut -d
↪' ' -f 4 | sed 's/["}]//g'
```

- Release a floating ip

..code-block:: bash

    neutron floatingip-list –tenant-id $TENANT_ID neutron floatingip-disassociate $ID neutron floatingip-delete $ID

- Release all floating ips of a tenant

```
for ID in $(neutron floatingip-list --tenant-id $TENANT_ID -c id -f csv |grep -v␣
↪float | sed 's/"//g'); do neutron floatingip-disassociate $ID; neutron floatingip-
↪delete $ID; done
```

- Firewall rule handling

```
neutron security-group-list
neutron security-group-create --protocol ICMP --direction ingress <group_id>
neutron security-group-rule-list
```

- Quota (independent from nova network quotas!)

```
neutron quota-update --network 0 --router 0 --floatingip 5 --tenant-id <tenant_id>
neutron quota-list
```

- Complete example

```
neutron net-create external --router:external=True
neutron subnet-create --disable-dhcp external 10.10.10.0/24
neutron net-create net0
neutron subnet-create --name net0-subnet0 --dns-nameserver 8.8.8.8 net0 192.168.100.0/
↪24
neutron router-create extrouter
neutron router-gateway-set extrouter external
neutron router-interface-add extrouter net0-subnet0
neutron security-group-rule-create --protocol icmp default
neutron security-group-rule-create --protocol tcp --port-range-min 22 --port-range-
↪max 22 default
ip netns exec qdhcp-<subnet_uuid> ssh <user>@<machine_ip>
ip a add 10.10.10.1/24 dev br-ex
iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -j MASQUERADE
```

Managing security groups

- Security groups define access rules for virtual machines

```
nova secgroup-list
nova secgroup-create mygroup "test group"
nova secgroup-add-rule mygroup tcp <from-port> <to-port> 0.0.0.0/0
nova secgroup-list-rules mygroup
```

Injecting SSH keys

```
nova keypair-list
nova keypair-add --pub_key ~/.ssh/id_dsa.pub a_name
```

Handling instances

- Instances can be found in */var/lib/nova/instances*
- Create a new machine

```
nova flavor-list
nova image-list
nova boot --poll --flavor <flavor_id> --image <image_id> --key_name <key_name> --
↪security_group mygroup <machine_name>
nova list --all-tenants
```

- Logfile */var/log/nova/compute.log*
- Get console output

```
nova console-log <machine_id>
```

- Remove a machine

```
nova delete <machine_id>
```

- If it cannot be removed use

```
nova force-delete <machine_id>
```

- Start / stop / suspend existing machine

```
nova [start|stop|suspend] <machine_id>
```

- Show details about a machine

```
nova show <machine_id>
```

- Connect to machines display

```
nova get-vnc-console <machine_id> novnc
```

- Show all vms and where they are running

```
nova-manage vm list
```

- Connect to a neutron network

```
nova boot --nic net-id=<subnet_id>
```

- Execute a script after creation (image needs to support cloud init and nova metadata must be running)

```
nova boot --user-data ./myscript.sh --flavor ...
```

- In user-data scripts cloud-config can be used to configure the machine in yaml or by invoking puppet (see http://docs.openstack.org/user-guide/content/user-data.html)

VNC access

- First install requirements *novnc* and *openstack-nova-novncproxy*
- Edit /etc/nova/nova.conf

```
novnc_enabled=true
vnc_keymap="de-de"
```

- Make sure *nova-consoleauth* is running

```
nova-manage service list
```

- `vncserver_proxyclient_address` must contain the official ip of the compute node
- Get an access url to throw in your browser

```
nova get-vnc-console <machine_id> novnc
```

Adding additional storage

- Cinder uses LVM2 (or Ceph, NetApp, ...) + ISCSI
- Can only attach a block device to one vm
- Activate Cinder in /etc/nova/nova.conf (restart nova-api and cinder-api afterwards)

```
volume_api_class=nova.volume.cinder.API
enable_apis=ec2,osapi_compute,metadata
```

- Create and attach a new columne

```
cinder create --display_name test 1
cinder list
nova volume-list
nova volume-attach <device_id> <volume_id> auto
```

- Create a snapshot

```
nova volume-detach <machine_id> <volumne_id>
cinder snapshot-create --display-name <name> <volumne_id>
```

- Restore a snapshot

```
cinder snapshot-list
cinder create <size> --snapshot-id <snapshot_uuid> --display-name <name>
```

- Boot from image in cinder

```
cinder create <size> --display-name <name> --image-id <glance_image_id>
nova boot --block-device-mapping vda=<volume_id> --flavor ...
```

- Resize a volumne offline

```
cinder extend <volumne_id> <new_size>
```

- QoS

```
cinder qos-create standard-iops consumer="front-end" read_iops_sec=400 write_iops_
↪sec=200
cinder qos-associate <qos_id> <volumne_id>
```

Quotas

- A value of -1 means unlimited
- Show all quotas of a tenant / project

```
nova quota-show --tenant <tenant>

 * To configure default quota for all tenants edit ``/etc/nova/nova.conf`` and set␣
↪the desired quota like
```

```
quota_instances=100

 * To update the quota of just one tenant execute
```

```
nova quota-update <tenant-id> --instances 100
```

Ceilometer

- Collects data for statistics, alarmings (“monitoring as a service”) or interaction with Heat

- Compute agent polls libvirt, central agent polls Openstack infrastructure, collector collects data in ampq or database, alarm evaluator decides if an alarm should take place, alarm notifier sends the alarm
- QuickStart guide http://openstack.redhat.com/CeilometerQuickStart
- List all what can be monitored

```
ceilometer meter-list
```

- List collected data

```
ceilometer sample-list --meter cpu
```

Heat

- http://docs.openstack.org/developer/heat/template_guide/hot_guide.html
- http://docs.openstack.org/developer/heat/template_guide/openstack.html
- Examples can be found on https://github.com/openstack/heat-templates/tree/master
- Execute a heat template with parameters from console

```
heat stack-create mystack --template-file=<filename> --parameters="Param1=value;
↪Param2=value"
```

- Example script

```
heat_template_version: 2013-05-23

description: Create a network and an instance attached to it

parameters:
  public_net_id:
    type: string
    description: >
      ID of floating network

resources:
  private_net:
    type: OS::Neutron::Net
    properties:
      name: Privatenet

  private_subnet:
    type: OS::Neutron::Subnet
    properties:
      network_id: { get_resource: private_net }
      cidr: 192.168.1.0/24
      gateway_ip: 192.168.1.1
      allocation_pools:
        - start: 192.168.1.2
          end: 192.168.1.254

  router:
    type: OS::Neutron::Router

  router_gateway:
    type: OS::Neutron::RouterGateway
```

```
    properties:
      router_id: { get_resource: router }
      network_id: { get_param: public_net_id }

  router_interface:
    type: OS::Neutron::RouterInterface
    properties:
      router_id: { get_resource: router }
      subnet_id: { get_resource: private_subnet }

  server1:
    type: OS::Nova::Server
    properties:
      name: Server1
      image: Test Image
      flavor: m1.small
      networks:
        - port: { get_resource: server1_port }

  server1_port:
    type: OS::Neutron::Port
    properties:
      network_id: { get_resource: private_net }
      fixed_ips:
        - subnet_id: { get_resource: private_subnet }
```

Savana

- Register an image in glance found in the plugin page e.g. http://docs.openstack.org/developer/sahara/userdoc/spark_plugin.html
- Register image in Data Processing -> Image Registry as described on the plugin page e.g. for Spark the user is ubuntu and tag is Spark version 1.0.0
- Create at least one Node Group Template (better one for master and one for slave nodes)
- Create a Cluster Template to combine the Node Group Templates and define number of nodes per template
- Click on Cluter -> Create Cluster

Automatically backup instances

- You can choose weekly instead of daily

```
nova backup <device_id> <backup_name> daily <keep_x_copies>
```

Live migration

- Setup as described in http://docs.openstack.org/grizzly/openstack-compute/admin/content/configuring-migrations.html
- Migrate a vm to another hypervisor

```
nova live-migration <machine_id> <new_hypervisor>
```

Where to find which service?

```
nova host-list
nova hypervisor-list
```

Where to find which instance?

- Get hypervisor of an instance

```
nova show <machine_id> | grep OS-EXT-SRV-ATTR:host
```

- List all instances of a hypervisor

```
nova hypervisor-servers <host>
```

Statistics

```
nova hypervisor-stats
```

Updating to a new version

- Every service has a db sync command

```
keystone-manage -vvv db_sync
```

Logging & Debugging

- Get an overall overview about the status of openstack

```
openstack-status
```

- Every manage command like *nova-manage* or *cinder-manager* has a parameter *logs errors*
- You can add the following lines to all *[DEFAULT]* config sections of all subsystems like nova or keystone etc

```
verbose=True
debug=True
```

- Every command has a *–debug* parameter

```
nova --debug list
```

- Configure logging e.g. open /etc/nova/nova.conf and add the following line in *[DEFAULT]* secion

```
log-config=/etc/nova/logging.conf
```

- Now create /etc/nova/logging.conf with the following content (syntax is *python logging <http://docs.python.org/3/library/logging.html>*)

```
[logger_nova]
level = DEBUG
handlers = stderr
qualname = nova
```

- Got a *Malformed request url (HTTP 400)* -> Check keystone (user / service / endpoint configuration) and service config for *auth_strategy=keystone*

```
keystone service-list
kestone endpoint-list
```

- Got a *ERROR n/a (HTTP 401)* -> thats an auth failure check service and api config for same as above + tenant / user / password

Compute node crashed

- If the did not crash completely but openstack-nova-compute service is broken, the machine will still be running and you can ssh into them but not use vnc
- If you decide to nevertheless migrate all vms first halt them otherwise the disk images will get crushed

```
ssh <HOSTNAME_OF_CRASHED_NODE>
for VM in $(virsh list --uuid); do virsh shutdown $VM; done
sleep 10
for VM in $(virsh list --uuid); do virsh destroy $VM; done
```

- Maybe you can use *nova evacuate* <*server*> <*vm*> instead of plain sql
- Connect to the master node and execute the following (dont forget to replace the two variables!)

```
echo "select uuid from instances where host = 'HOSTNAME_OF_CRASHED_NODE' and deleted␣
↪= 0;" | mysql --skip-column-names nova > broken_vms
echo "update instances set host = 'HOSTNAME_OF_NEW_NODE' where host = 'HOSTNAME_OF_
↪CRASHED_NODE' and deleted = 0;" | mysql nova
for VM in $(cat broken_vms); do nova reboot $VM; done
```

- The following command should return no results

```
nova list --host <HOSTNAME_OF_CRASHED_NODE>
```

Disable a service on a host

- For example disable a compute node

```
nova-manage service disable <host> nova-compute
```

Troubleshooting Keystone

- SSL error SSL_CTX_use_Privatekey_file:system lib -> Check permission of /etc/keystone/ssl (maybe chown keystone)
- User / services etc doesnt appear in the database -> edit /etc/keystone/keystone.conf section *[catalog]*

```
driver = keystone.catalog.backends.sql.Catalog
```

- Unable to communicate with identity service "Invalid tenant" "Not authorized" -> check that the os-username and -tenant you use have a corresponding admin role

```
keystone user-role-add --role-id <id_of_admin_role> --user-id <userid> --tenant-id
↪<tenantid>
```

- Select role in db

```
select m.data from user u join user_project_metadata m on u.id=m.user_id join project␣
↪p on p.id=m.project_id where u.name="nova";
select * from role where id="a4b2afdf62baifgafaifga7f";
```

- Check `token_format` in keystone.conf should be `UUID` by default
- '' 'Client' object has no attribute 'auth_tenant_id' ''

```
export SERVICE_TOKEN=
export SERVICE_ENDPOINT=
```

- Manually receive an auth token by executing `keystone token-get` or

```
curl -i 'http://127.0.0.1:5000/v2.0/tokens' -X POST -H "Content-Type: application/json
↪" -H "Accept: application/json"  -d '{"auth": {"tenantName": "admin",
↪"passwordCredentials": {"username": "admin", "password": "admin"}}}'
```

Troubleshooting Neutron

- What is for what? l2-agent (DHCP), l3-agent (floating ips and routers)
- Check the neutron metadata agent is running and accessible (lives on 169.254.0.0/16)

```
nova console-log <machine_id>
```

- Status overview

```
neutron agent-list
```

- Make sure the short hostname is not on loopback ip in `/etc/hosts`
- Check br-int and br-ext exist and br-tun for gre tunnel setup

```
ovs-vsctl show
```

- Check `/var/log/neutron` logs and that iproute tool support netns
- Get a shell in the network namespace

```
ip netns list
ip netns exec <namespace> bash
```

- `Timeout while waiting on RPC response - topic: "network"` -> check neutron config in `/etc/nova/nova.conf` on your compute nodes
- `Error: Local ip for ovs agent must be set when tunneling is enabled` -> network device is not up / configured or name used is not in dns / /etc/hosts

Troubleshooting Glance

- Invalid OpenStack identity credentials -> Comment out *flavor=keystone*

Troubleshooting Cinder

- Check the LVM volumne group

```
vgdisplay cinder-volumes
```

- Check that tgtd is running
- HTTP 401 Permission denied? -> Edit /etc/cinder/api-paste.ini section *[filter:authtoken]*

```
admin_tenant_name=service
admin_user=cinder
admin_password=cinder
```

- Cannot connect to AMQP server -> Edit /etc/cinder/cinder.conf

```
rpc_backend = cinder.rpc.impl_kombu
```

- Check nova is using cinder (edit /etc/nova/nova.conf)

```
volume_api_class=nova.volume.cinder.API
```

Troubleshooting Instances

- Check nova logs for errors

```
nova-manage logs errors
```

- Get information about the instance

```
nova show <device_id>
nova diagnostics <device_id>
```

- Instance in an broken task state?

```
nova reset-state <device_id>
nova reset-state --active <device_id>
```

- Qemu disk image is broken?

```
qemu-img check check <disk_file>
```

Troubleshooting Nova

- Read *Nova disaster recovery process <http://docs.openstack.org/trunk/openstack-compute/admin/content/nova-disaster-recovery-process.html>*
- `Instance instance-XXXXXXXX already exists` –> the instance is running check with `virsh list --all`

- Use *virsh* / *virt-manager* or *virt-viewer* for debugging purpose
- Check nova services (ensure ntp is running on all nova nodes)

```
nova-manage service list
```

- Restart all nova services

```
for svc in api objectstore compute network volume scheduler cert; do service␣
↪openstack-nova-$svc restart ; done
```

- Check cpu properties / kernel

```
egrep '(vmx|svm)' /proc/cpuinfo
lsmod | grep kvm
```

- No valid hosts found and log file says `Unexpected vif_type=binding_failed` -> check local_ip setting in [ovs] section in file /etc/neutron/plugins/ml2/ml2_conf.ini
- libvirtError: internal error no supported architecture for os type 'hvm'

```
modprobe kvm
```

- xxx in server list / Unable to connect to amqp server -> check that rabbitmq or qpid server is running
- RabbitMQ config in */etc/nova/nova.conf*

```
rpc_backend = nova.rpc.impl_kombu
rabbit_host=127.0.0.1
```

- Unable to connect to AMQP server client: 0-10 -> rpc_backend in nova.conf doesnt match used server
- AMQP server is unreachable: Socket closed -> Check credentials if socket is reachable

```
rabbitmqctl list_users
rabbitmqctl change_password guest guest
```

- or configure user / pass for rabbitmq access in */etc/nova/nova.conf*

```
rabbit_userid=guest
rabbit_password=guest
```

- nova image-list returns *HTTP 401* -> thats auth failed check */etc/nova/api-paste.ini* section *[filter:authtoken]* for

```
admin_tenant_name=service
admin_user=nova
admin_password=nova
```

- All nova commands return *Malformed request url (HTTP 400)* -> check that openstack-nova-compute is running
- compute manager *nova [-] list index out of range* -> you're doomed with the nova-compute cannot restart because you have machine in ERROR state bug. only way is to manually delete the machine from the database nova (table instances and all constraints)
- *libvirt unable to read from monitor* -> check vnc settings in */etc/nova/nova.conf*
- nova list returns *[Errno 111] Connection refused* -> Check that nova-compute is running, maybe configure its port in /etc/nova/nova.conf

```
[nova.service]
osapi_compute_listen_port=8774
```

Troubleshooting Horizon

- Disable SeLinux *setenfore 0*
- Permission denied -> Check httpd.conf, add the following to Directory directive

```
Require all granted
```

- Command node not found -> Install http://www.nodejs.org

Programming

- Overview about Openstack APIs http://www.ibm.com/developerworks/cloud/library/cl-openstack-pythonapis/index.html
- Keystone

```
import keystoneclient.v2_0.client as ksclient
conn = ksclient.Client(auth_url="http://127.0.0.1:35357/v2.0", username="nova",␣
↪password="nova", tenant_name="services")
print conn.auth_token
```

- Nova

```
import sys
import time
import novaclient.v1_1.client as nvclient

username = "admin"
password = "admin"
tenant = "admin"
auth_url = "http://127.0.0.1:5000/v2.0/"

def get_hypervisor_for_host(hostname):
  try:
    hypervisor = nova.hypervisors.search(hostname, servers=True)[0]
  except Exception:
    hypervisor = None

  return hypervisor

nova = nvclient.Client(username, password, tenant, auth_url)
hypervisor = get_hypervisor_for_host(sys.argv[1])

if not hypervisor:
  print "Hypervisor " + sys.argv[1] + " cannot be found"
  sys.exit(1)

if hasattr(hypervisor, "servers"):
  waiting_for_migrations = True

  for vm_dict in hypervisor.servers:
```

```
    vm = nova.servers.get(vm_dict.get('uuid'))
    print "Migrating " + vm.name
    vm.live_migrate()

  # wait for migration to complete
  sys.stdout.write("\nWaiting for migrations to finish ...")

  while waiting_for_migrations:
    sys.stdout.write(".")
    hypervisor = get_hypervisor_for_host(sys.argv[1])

    if not hypervisor or not hasattr(hypervisor, "servers"):
      waiting_for_migrations = False
      sys.stdout.write("\n")
    else:
      time.sleep(1)
else:
  print "Hypervisor " + sys.argv[1] + " serves no vms"
```

Cool addons

- http://zerovm.org

Pacemaker

Setup

- Run these commands on all nodes

```
yum install pacemaker pcs
passwd hacluster
corosync-keygen
```

- Copy `/etc/corosync/authkey` to all nodes
- Open UDP port 5405
- Setup the cluster

  systemctl start corosync systemctl start pacemaker systemctl start pcsd pcs cluster auth node1 node2 node3 pcs cluster setup –name my-cluster node1 node2 node3

- Edit `/etc/corosync/corosync.conf` and set

```
secauth: on
crypto_cipher: aes256
crypto_hash: sha512
```

- Restart corosync

```
systemctl restart corosync
pcs cluster start
pcs cluster enable
pcs cluster status
```

Create a shared floating ip

```
pcs resource create ha-ip ocf:heartbeat:IPaddr2 params ip="192.168.3.3" cidr_netmask=
↪"32" op monitor interval="10s"
```

Show all resources

```
pcs resource show
```

Set mandatory fencing off

```
pcs property set stonith-enabled=false
```

Configure fencing via IPMI

- Install `fence-agents` on every node

```
pcs stonith create node1 fence_ipmilan params auth="password" ipaddr="1.2.3.4" login=
↪"root" passwd="secret" pcmk_host_list="node1" op monitor interval="30s"
pcs stonith create node2 fence_ipmilan params auth="password" ipaddr="1.2.3.4" login=
↪"root" passwd="secret" pcmk_host_list="node2" op monitor interval="30s"
```

- To test it execute

```
pcs stonith fence another-node
```

Configure fencing for libvirtd

- On libvirt node install `fence-virtd`

```
mkdir /etc/cluster
dd if=/dev/urandom of=/etc/cluster/fence_xvm.key bs=4k count=1
semanage boolean -m --on fenced_can_network_connect
systemctl enable fence_virtd
systemctl start fence_virtd
```

- On all nodes install `fence-virt` and copy `/etc/cluster/fence_xvm.key` to them
- On all nodes and host allow tcp and udp to port `1229`
- On one node test fencing and enable it afterwards

```
fence_xvm -o list -a 225.0.0.12
pcs stonith create ha-fence fence_xvm multicast_address=225.0.0.12 pcmk_host_list=
↪"node1 node2 node3"
```

Enable / disable a resource

```
pcs resource enable/disable <resource>
```

Maintenance a node

- On the node exec

```
pcs cluster standby
```

Reset logs of a resource

```
pcs resource cleanup <resource>
```

Debug a resource

```
pcs resource debug-start <resource>
```

Move a resource to another node

```
pcs resource move <resource> <node>
```

Define a service

- Cloned means the service runs on all nodes, no clone only one node
- mongodb is a name for the resource, mongod the name of the init script / systemd service

```
pcs resource create mongodb lsb:mongod --clone
```

- For systemd services

```
pcs resource create myhaproxy systemd:mongod
```

Define a mountpoint

```
pcs resource create my-nfs Filesystem device=192.168.1.1:/export/something directory=/
↪mnt fstype=nfs options=nolock
```

Group resources

```
pcs resource group add my-group resource1 resource2
```

Define constraints

- Start order

```
pcs constraint order start <resource1> then <resource2>
```

- Ensure both resources are on the same node

```
pcs contraint colocation add <resource1> with <resource2>
```

- Prefered node

```
pcs contraint location <resource> prefers <node>
```

- Delete a contraint (without with or then or prefers...)

```
pcs contraint [colocation|...] remove <resource1> <resource2>
```

Setup a 2 node cluster

- Normally an odd number greater than 1 is used to build a cluster to form a valid quorom

```
pcs resource clone lsb:httpd globally-unique=true clone-max=2 clone-node-max=2
```

Spark

Overview

- Configures like Hadoop 1
- Executing engine completly in RAM therefore lightning fast
- Can integrate with YARN
- Can use HDFS, S3, HBase, Cassandra, local files...
- Easier programming interface using resilient distributed dataset (RDD) - an immutable list distributed over the cluster

Installation

- Unzip latest Spark prebuild for latest Hadoop (can also be used standalone)

```
echo "root@some-slave-host" >> /opt/spark/conf/slaves
/opt/spark/bin/start-all.sh
```

Status Overview

- Point your web browser to http://<yourmasternode>:8080
- Run example

```
bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master spark://127.0.0.1:7077 \
--executor-memory 1G \
--total-executor-cores 1 \
lib/spark-examples-1.1.1-hadoop2.4.0.jar \
1000
```

Get a Python shell on the cluster

```
/opt/spark/bin/pyspark
```

Sample code

- Grep for failures in kern.log

```
log = sc.textFile("/var/log/kern.log")
rrd = log.filter(lambda x: "error" or "failure" in x.lower())
for x in rrd.collect(): print x
```

- Sum up bytes of an apache acces log

```
log = sc.textFile("/var/log/httpd/access.log")
log.map(lambda x: x.split(" ")[9]).filter(lambda x: "-" not in x).map(lambda x:␣
↪int(x)).sum()
```

Add slave nodes to a running cluster

- On the slave node execute

```
/opt/spark/sbin/start-slave.sh some-worker-id spark://master-node:7077
```

Troubleshooting

- Be sure if remotely submiting jobs to use the DNS name and not IP
- `TaskSchedulerImpl: Initial job has not accepted any resources; check your cluster UI to ensure that workers are registered and have sufficient memory`

Monitoring

- http://www.hammerlab.org/2015/02/27/monitoring-spark-with-graphite-and-grafana/

Torque

Overview

- Controller runs `pbs_server` and `pbs_sched`
- Edit `/var/lib/torque/server_priv/nodes` to add all compute nodes (np = number of processors, gpus = number of gpus)

```
pulsar np=4
pulsar-mobile np=4
```

- Compute nodes run `pbs_mom` after editing `/etc/torque/mom/config`
- Verify that all nodes are connected to the controller

```
pbsnodes -a
```

Submitting jobs

```
qsub <my_cool_tool>
```

- Check the job queue with `qstat -n`

CHAPTER 23

Databases

Cassandra

Overview

- Cassandra is a partitioned row store database (Db split over whole cluster)
- Replication for HA
- Designed from the ground up as a distributed database with peer-to-peer communication (No Master node)
- Feels lot like MySQL
- Cassandra Demo App https://github.com/snazy/barker

Specifics

- Update of non-existent data row will insert it as new, Insert of existing data row will update it
- Select WHERE clause only possible for indexed fields (or by appending ALLOW FILTERING)

Start CQL Client

```
bin/cqlsh
```

Create new database

```
cqlsh> CREATE KEYSPACE app WITH replication = {'class': 'SimpleStrategy',
↪'replication_factor': 1};
cqlsh> use app;
```

Create new table

```
cqlsh:app> CREATE TABLE users (username TEXT PRIMARY KEY, firstname TEXT, surname␣
↪TEXT, password BLOB, last_login TIMESTAMP);
```

INSERT, UPDATE, SELECT, DELETE

```
cqlsh:app> INSERT INTO users (username, firstname, surname) VALUES ('balle',
↪'Sebastian', 'Ballmann') IF NOT EXISTS;
cqlsh:app> UPDATE users SET firstname='Bastian' WHERE username = 'balle';
cqlsh:app> SELECT username, last_login FROM users WHERE surname='Ballmann' ALLOW␣
↪FILTERING;
cqlsh:app> DELETE FROM users WHERE username='balle';
```

Create a cluster

- Edit `conf/cassandra.yaml`
- Set at least the Cluster name and listen_address
- Include all nodes in seed provider list
- Start cassandra on all nodes
- Check cluster status

```
bin/nodetool status
```

Import data from MySQL

- Install squoop (http://sqoop.apache.org/)

```
sqoop import --connect jdbc:mysql://127.0.0.1/dev --username root --cassandra-
↪keyspace dev --cassandra-create-schema
```

Elasticsearch

Overview

- http://elasticsearchtutorial.blogspot.ch/

Install browser plugin

```
/usr/share/elasticsearch/bin/plugin install mobz/elasticsearch-head
```

- Now point your browser to http://localhost:9200/_plugin/head/

Dump database

```
curl -XPUT 'http://localhost:9200/_snapshot/my_backup' -d '{ "type": "fs", "settings
↪": { "location": "/mount/backups/my_backup", "compress": true }}'
```

Configure cluster

- Edit `/etc/elasticsearch/elasticsearch.yml`
- Set cluster.name, node.name, network.host and discovery.zen.ping.unicast.hosts
- Make sure TCP port 9300 is open

Cluster autodiscovery is not working

- Edit `/etc/elasticsearch/elasticsearch.yml`
- Make one node the master node

```
node.master: true
node.data: true
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ["node1.example.com"]
```

- Let the others connect to the master node

```
node.master: false
node.data: true
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ["node1.example.com"]
```

Add authentication to Elasticsearch

- Does not work in cluster mode

```
bin/plugin install license
bin/plugin install shield
```

- Edit elasticsearch.yml

```
action.auto_create_index: .security
```

- Restart elasticsearch and add a user

```
bin/shield/esusers useradd admin -r admin
```

Insert data manually

```
curl -XPUT 'http://localhost:9200/dept/employee/1' -d '{ "empname": "emp1"}'
```

Configure Rsyslog to log to Logstash

- Create file /etc/rsyslog.d/logstash.conf

```
*.* @127.0.0.1:5544
```

Configure Rsyslog to log to Fluentd

- Create file /etc/rsyslog.d/fluentd.conf

```
*.* @127.0.0.1:42185
```

Configure Rsyslog to log directly to Elasticsearch

- For RHEL7 / CentOS 7 the rsyslog-elasticsearch plugin is included
- For RHEL6 use repo http://rpms.adiscon.com/v5-stable/rsyslog.repo

```
yum install rsyslog-elasticsearch
```

- Now edit `/etc/rsyslog.conf`

```
module(load="imuxsock")             # for listening to /dev/log
module(load="omelasticsearch") # for outputting to Elasticsearch
# this is for index names to be like: logstash-YYYY.MM.DD
template(name="logstash-index"
  type="list") {
    constant(value="logstash-")
    property(name="timereported" dateFormat="rfc3339" position.from="1" position.to="4
↪")
    constant(value=".")
    property(name="timereported" dateFormat="rfc3339" position.from="6" position.to="7
↪")
    constant(value=".")
    property(name="timereported" dateFormat="rfc3339" position.from="9" position.to=
↪"10")
}

# this is for formatting our syslog in JSON with @timestamp
template(name="plain-syslog"
  type="list") {
    constant(value="{")
      constant(value="\"@timestamp\":\"")     property(name="timereported" dateFormat=
↪"rfc3339")
      constant(value="\",\"host\":\"")        property(name="hostname")
      constant(value="\",\"severity\":\"")    property(name="syslogseverity-text")
      constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
      constant(value="\",\"tag\":\"")   property(name="syslogtag" format="on")
      constant(value="\",\"message\":\"")    property(name="msg" format="on")
    constant(value="\"}")
}
# this is where we actually send the logs to Elasticsearch (localhost:9200 by default)
action(type="omelasticsearch"
    template="plain-syslog"
    searchIndex="logstash-index"
    dynSearchIndex="on")
```

Let Elasticsearch listen only on loopback

- Edit /etc/elasticsearch/elasticsearch.yml

```
network.host: 127.0.0.1
```

Use logstash as log aggregator

- Create /etc/logstash/conf.d/10-syslog.conf

```
input {
  syslog {
    type => syslog
    port => 5544
  }
}

filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %
↪{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?: %
↪{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

- Create /etc/logstash/conf.d/30-elasticsearch-output.conf

```
output {
  elasticsearch { host => localhost }
  stdout { codec => rubydebug }
}
```

Use fluentd as log aggregator

- Can collecd and parse log from many sources (200+)
- Is written in Ruby and needs no Java like Logstash
- Can output to many directions including files, mongodb and of course elasticsearch
- For installation see http://docs.fluentd.org/categories/installation
- Install Elasticsearch plugin

```
gem install fluent-plugin-elasticsearch
```

- If your ruby version is too old or buggy install fluentd inside rvm

```
curl -sSL https://get.rvm.io | bash -s stable --ruby
source /usr/local/rvm/scripts/rvm
gem install fluentd
gem install fluent-plugin-elasticsearch
```

- Regular expressions for parsing logs can be tested on http://rubular.com/
- Time format options can be looked up here http://www.ruby-doc.org/core-1.9.3/Time.html#method-i-strftime
- Example config

```
# live debugging agent
#<source>
#  type debug_agent
#  bind 127.0.0.1
#  port 24230
#</source>

# Listen to Syslog
<source>
  type syslog
  port 42185
  tag system.raw
</source>

# Apache Access Logs
<source>
  type tail
  format apache2
  path /var/log/httpd/access_log
  pos_file /var/log/fluentd/httpd.access.pos
  tag httpd.access
</source>

# Apache Error Logs
<source>
  type tail
  format apache_error
  path /var/log/httpd/error_log
  pos_file /var/log/fluentd/httpd.error.pos
  tag httpd.error
</source>

# Tag kernel messages
<match system.raw.**>
  type rewrite_tag_filter
  rewriterule1 ident ^kernel$  kernel.raw # kernel events
  rewriterule2 ident .* system.unmatched     # let all else through
</match>

# Identify iptables messages
<match kernel.raw.**>
  type rewrite_tag_filter
  rewriterule1 message ^IN=.* OUT=.+$ iptables.raw  # iptables events
  rewriterule2 message .* kernel.unmatched      # let all else through
```

```
</match>

 # Parse iptables messages
 # IN=eno1 OUT= MAC=aa:bb:cc:aa:bb:cc:aa:bb:cc:aa:bb:cc:aa:00 SRC=192.168.10.42␣
↪DST=192.168.10.23 LEN=148 TOS=0x00 PREC=0x00 TTL=255 ID=53270 DF PROTO=UDP SPT=5353␣
↪DPT=5353 LEN=128
 <match iptables.raw.**>
   type parser
   key_name message # this is the field to be parsed!
   format /^IN=(?<iface>.*) OUT=(?<oface>.*) MAC=(?<mac>.*?) (SRC=(?<srcip>.*))?␣
↪(DST=(?<dstip>.*))? LEN=(?<pkglen>.+) TOS=(?<pkgtos>.+) PREC=(?<pkgrec>.+) TTL=(?
↪<pkgttl>.+) ID=(?<ipid>.+) \w{0,2}\s?PROTO=(?<pkgproto>.+)( SPT=(?<srcport>.+)␣
↪DPT=(?<dstport>.+) LEN=(.*))?$/
   time_format %b %d %H:%M:%S
   tag iptables.parsed
 </match>

 # write to file
 #<match iptables.parsed>
 #  type file
 #  path /var/log/td-agent/iptables.log
 #</match>

 # Write to elasticsearch
 <match *.**>
     type elasticsearch
     host localhost
     port 9200
     include_tag_key true
     tag_key _key
     logstash_format true
     flush_interval 10s
 </match>

 # Log to stdout for debugging
 #<match *.**>
 #    type stdout
 #</match>
```

- Last but not least configure your systlog to send messages to fluentd

```
*.* @127.0.0.1:42185
```

- Start fluentd in foreground for testing purpose

```
fluentd -c /etc/fluent/fluent.conf -vv
```

Kibana Web Frontend

- Install it http://www.elasticsearch.org/overview/kibana/installation/
- Run bin/kibana
- Or use this systemd service file

```
[Service]
ExecStart=/opt/kibana4/bin/kibana
```

```
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=kibana4
User=root
Group=root
Environment=NODE_ENV=production

[Install]
WantedBy=multi-user.target
```

- Have a look at https://www.youtube.com/watch?v=hXiBe8NcLPA&index=4&list=UUh7Gp4Z-f2Dyp5pSpLO3Vpg
- For Dashboards see https://github.com/search?utf8=%E2%9C%93&q=kibana+dashboard&type=Repositories&ref=searchresults

Mongo Db

Overview

- You dont need to create users / passwords / dbs or tables just use it
- Programming language is Javascript

List all databases

```
show dbs
```

Connect to a db

```
use <db>
```

List all collections

```
show collections
```

Insert data

```
db.<collection>.save({name: "wurst", firstname: "hans"})
```

- Import in json format

```
cat json_file | /usr/bin/mongoimport --type json -d mydb -c mycollection
```

Select from collection

- all

```
db.<collection>.find()
```

- filter by field

```
db.<collection>.find(field: "value")
```

- filter by regexp

```
db.<collection>.find(field: /\d+/)
```

- select specific fields

```
db.<collection>.find({},{"field_to_select": 1})
```

Iterate over all results

```
var cursor = db.<collection>.find();
while (cursor.hasNext()) printjson(cursor.next());
```

- Or better

```
db.<collection>.find().forEach(printjson)
```

Sorting

- Lowest first

```
db.<collection>.find().sort({"field": 1})
```

- Highest first

```
db.<collection>.find().sort({"field": -1})
```

Update data

```
db.<collection>.update({"_id": 1}, {$set: {"field": "new value"}})
```

Delete data

- Remove complete collection

```
db.<collection>.drop()
```

- Remove some entries

```
db.<collection>.remove({"name": "wurst"})
```

- Delete a field

  db.<collection>.update({“_id”: 1}, {$unset: {“field”: “”}})

```

```

- Delete whole database

```
db.dropDatabase()
```

Working with timestamps

- Get all entries from 1.12. till 6.12.

```
db.snmp.find({'time': {'$gt': ISODate('2012-12-01T00:00:00'), '$lt': ISODate("2012-12-
↪05T23:59:59")}})
```

Select distinct values

```
db.<collection>.distinct('field')
```

Create index on collection field

- sparse will only create index if a value for that field exists

```
db.<collection>.ensureIndex( { myfield: 1 }, {sparse: true, background: true} );
```

- Unique constraint

```
db.<collection>.ensureIndex( { myfield: 1 }, {unique: true} );
```

Show indexes

- Of collection

```
db.<collection>.getIndexes()
```

- All

```
db.system.indexes.find()
```

Using references

```
some_field: new DBRef('collection_of_target', target_document._id)
```

User administration

- To setup authentication edit `/etc/mongodb.conf` and set

```
auth = true
```

- Now execute the following in mongo shell

```
use <db>
db.addUser({user: "foo", pwd: "bar", roles: ["readWrite", "dbAdmin"]})
```

- use role `read` for read-only access
- role `root` will give overall access
- other useful roles “userAdminAnyDatabase”, “readWriteAnyDatabase”, “dbAdminAnyDatabase” (use db admin in this case)
- give `clusterAdmin` roles if user should be able to configure clustering
- To log into a database

```
db.auth("user", "pass")
```

- List all users

```
db.getUsers()
```

- List one user

```
db.getUser("foo")
```

- To change a users password

```
db.changeUserPassword("user", "newpassword")
```

- Update a users roles

```
db.grantRolesToUser("vip_user", ["userAdminAnyDatabase", "readWriteAnyDatabase",
↪"dbAdminAnyDatabase", "clusterAdmin"])
```

Replication

- All nodes share the same data
- You need at least 3 server
- Edit `/etc/mongodb.conf` and set a replSet

```
replSet = rs0
```

- On the master node execute the folowing in mongo shell

```
rs.initiate()
rs.conf()
rs.add("mongodb1.example.net")
rs.add("mongodb2.example.net")
rs.status()
```

- You can define one as master and the others as slaves

Clustering / Sharding

- Sharding means to spread the database horizontally across multiple servers
- config servers store the complete metadata of a cluster (you should have 3)

```
mongod --configsvr --dbpath <path>
```

- mongos are query servers (configdb must define configsvr nodes in same order!)

```
mongos --configdb node1:27019,node2:27019,node3:27019
```

- shard servers store the data

```
mongo --host querysrv --port 27017
```

- You can add the new shard to a replica set

```
sh.addShard( "rs0/newnode:27017" )
```

- or to a whole

```
sh.addShard( "newnode:27017" )
```

- You can set `configsvr` or `shardsvr=true` in `/etc/mongodb.conf` to make the server a config, query or shard server
- configdb = <cfgsrv1>, <cfgsrv2>, <cfgsrv3>
- To enable sharding for a database run

```
sh.enableSharding("<db>")
sh.status()
```

Backup

```
#!/bin/bash

$BACKUP_DIR="/my/backup/path"

cd $BACKUP_DIR
/usr/bin/mongodump

/bin/tar cvjf `hostname`_`date +%Y%m%d`.tbz dump && /bin/rm -rf $BACKUP_DIR/dump
```

Show real data size

- With indizes

```
db.<collection>.totalSize()
```

- Only data

```
db.<collection>.dataSize()
```

Defrag

- Whole database

```
db.repairDatabase()
```

- Single collection

```
db.<collection>.compact()
```

Load collection into memory

```
db.runcommand({ touch: "collection_name", data: true, index: true})
```

Round robin collections

If you've not heard of capped collections before, they're a nice little feature of MongoDB that lets you have a high-performance circular queue. Capped collections have the following nice features:

- They "remember" the insertion order of their documents
- They store inserted documents in the insertion order on disk
- They remove the oldest documents in the collection automatically as new documents are inserted

```
db.create_collection(
  'capped_collection',
  capped=True,
  size=size_in_bytes,      # required
  max=max_number_of_docs, # optional
  autoIndexId=False)       # optional
```

- One can tail for new data in capped collections

```
cur = db.capped_collection.find(
      tailable=True,
      await_data=True)
```

- For more see http://blog.pythonisito.com/2013/04/mongodb-pubsub-with-capped-collections.html

Troubleshooting

- I always get `not master` errors

```
rs.status()
rs.initiate()
```

- Or if you really want to access a slave node

```
db.setSlaveOk()
```

Getting help

```
db.help()
```

MySQL

Create new user

```
grant all privileges on db.* to user@'%' identified by 'passwd';
```

Show extended table information

```
show table status;
```

Show table structure

```
describe <table>
```

Show indices

```
show index from <table>
```

Show all running processes

```
show full processlist;
```

Kill a process

```
kill <pid>;
```

Repair a table

```
repair table <table>;
```

Backup whole database

```
mysqldump <db> > backup.sql
```

Selective backup

```
select * into dumpfile "backup.sql" from table where foo="bar";
```

Create database with utf-8 charset

```
create database <db_name> default character set utf8;
```

- Or edit my.cnf

```
[mysqld]
default-character-set = utf8

[mysql]
default-character-set = utf8
```

Use InnoDB tables instead of MyISAM

- Edit my.cnf

```
default-storage-engine=INNODB
```

Change db charset

```
alter database <db_name> character set utf8;
```

Add Foreign Key Contraint

```
alter table add constraint <contraint_name> foreign key <column> references <table>
↪<column> on delete cascade;
```

Add check constraint

```
alter table add constraint <name> CHECK (some_column > 0 and other_column != "");
```

Add index

```
create index <name> on <table> (<column>);
```

Delete entries older than 30 days

```
DELETE FROM <table> WHERE DATE_SUB(CURDATE(),INTERVAL 30 DAY) <= <column>;
```

Temporary tables

```
CREATE TEMPORARY TABLE table2 AS (SELECT * FROM table1)
```

Reset root password

- Restart db with ‘‘ –skip-grant-tables –skip-networking‘‘

Active active cluster

- Install `MariaDB-Galera-server` instead of mysql-server
- Edit `/etc/my.cnf` on all nodes

```
wsrep_cluster_address=gcomm://master-node
wsrep_node_address='<ip_of_this_node>'
wsrep_node_name='<name_of_this_node>'
```

- On all other nodes than master node init a base db

```
mysql_install_db --user=mysql --ldata=/var/lib/mysql
```

- On master node start

```
service mysql bootstrap
```

- On all other nodes

```
service mysql start
```

Postgres

Create a database

```
createdb mydb -E UTF8 -O myuser
```

- or

```
CREATE DATABASE mydb WITH OWNER mypuser;
```

- By default this will copy template1 db

Create a user

```
createuser --password myuser
```

- or

```
CREATE USER myuser WITH password 'secret';
```

Change user password

```
ALTER USER myuser WITH PASSWORD 'moresecret';
```

Change user permissions

```
ALTER USER myuser CREATEDB;
```

Delete user

```
dropuser myuser
```

- or

```
DROP USER myuser
```

List databases

```
\l
```

Connect to a database

```
\c <db>
```

List all tables

```
\dt
```

Describe table

```
\d+ table
```

List user and permissions

```
\du
```

Show active connections

```
SELECT * FROM pg_stat_activity;
```

Export select as CSV

```
copy(select * from table) to '/some/file' with csv header;
```

Import CSV

```
copy table from '/some'file' with csv header;
```

Redis

Connect

- TCP socket

```
redis-cli -h my-host -p 1234 -a mypassword
```

- UNIX socket

```
redis-cli -s <path/to/unix.sock>
```

Select a database

- Default db is 0

```
redis> SELECT <index>
```

List all keys

```
redis> KEYS *
```

- With types

```
for KEY in $(redis-cli KEYS "*" | cut -d " " -f 2); do echo -en "$KEY - "; redis-cli␣
↪TYPE "$KEY"; done
```

Show type of keys

```
redis> type "key"
```

Read keys

- Normal key

```
redis> GET "key"
```

- set key

```
redis> SMEMBERS "key"
```

- hash key

```
redis> HGETALL "key"
```

Write keys

- Normal key

```
redis> SET "key" "value"
```

- set key

```
redis> SADD "key" "value"
```

- hash key

```
redis> HSET "key" "field" "value"
```

Delete a key

```
redis> DEL "key"
```

Drop database

```
redis> FLUSHDB
```

Save changes to disk

```
redis> SAVE
```

- You can define a periodic interval in `redis.conf`

```
save 60 99999
```

Monitor realtime requests

```
redis> MONITOR
```

CHAPTER 24

Virtualization

Docker

Install a new image

```
docker search <whatever>
docker pull <image>
```

Run image

- Runs a new container of image in interactive mode and starts a bash

```
docker run -i -t <image> bash
```

- Start an existing container

```
docker start -i <container_id>
```

List installed images

```
docker images
```

List running containers

```
docker ps
```

Save changes

```
docker commit <container_id> <image_name>
```

Export images

```
docker save <image> > <archive_file>
docker load -i <archive_file>
```

Update images

- Only one

```
docker pull <image>
```

- All images

```
docker images | awk '{print $1}' | xargs -L1 docker pull
```

Port forward

- Starts in daemon mode and forwards container port 80 to host port 8888 but only on loopback interface

```
docker run -d -p 127.0.0.1:8888:80 <image>
```

- Automatically forward all ports

```
docker run -P <image>
```

Set fixed IP for container

```
docker run --ip=<container_ip> --default-gateway=<gw_ip>
```

Get IP of container

```
docker inspect <container_id> | grep IPAddress
```

Share directory between host and container

- Via Dockerfile

```
VOLUME        ["/var/volume1", "/var/volume2"]
```

- Via command-line

```
-v /path/on/host:/path/in/container
```

Allow docker container to access DISPLAY

```
xhost +local:docker
```

Display CPU / RAM usage of container

```
docker stats <container_id>
```

Get STDOUT / STDERR from container

```
docker log <container_id>
```

Get a shell on a running container

Example docker file

```
#
# base image is latest official redhat rhel7
#
from rhel7:latest

# Update the system
RUN yum update -y

# Install web server
RUN yum install -y httpd

# Copy files to image
#COPY ./public-html/ /usr/local/apache2/htdocs/
#COPY ./my-httpd.conf /usr/local/apache2/conf/httpd.conf

# Start the service
EXPOSE 80
CMD ["-D", "FOREGROUND"]
ENTRYPOINT ["/usr/sbin/httpd"]
```

Troubleshooting

- Couldn't create Tag store: unexpected end of JSON input

```
rm /var/lib/docker/repositories
```

KVM / Qemu

Create an image

```
qemu-img create -f qcow2 disk.img 4G
```

Install a machine

```
qemu-kvm -m 1024 -boot -once=d -cdrom cd.iso disk.img
```

Start a machine

```
qemu-kvm -m 1024 disk.img
```

Start qemu monitor

```
qemu-kvm -m 1024 disk.img -monitor stdin
```

Downloading disk images

- Grab an image from http://pvirtualboxes.org or http://virtual-machine.org or http://virtualboximages.com/

Convert virtualbox or vmware image

```
qemu-img convert -O qcow2 Platte.(vdi|vmdk) Platte.img
```

- http://qemu-buch.de/de/index.php/QEMU-KVM-Buch/_Speichermedien/_Festplatten-Images_anderer_Virtualisierungssoftware

Convert to older Qcow version

```
qemu-img convert disk.img -O qcow2 -o compat=0.10 -c disk-2.img
```

Resize disk

- Switch off machine
- Resize only disk image (filesystem left alone)

```
qemu-img resize disk.img +10G
```

- Or create a new bigger image

```
qemu-img create -f qcow2 new-disk.img 20G
```

Disk encryption

- Create

```
qemu-img create -e -f qcow2 disk.img 10G
```

- Convert

```
qemu-img convert -e -O qcow2 disk.img disc-enc.img
```

Libvirt

Installation

```
yum install libvirt libvirtd python-virtinst qemu-kvm
```

Set up bridged network for direct access

- Edit `/etc//etc/sysconfig/network-scripts/ifcfg-eno1`

```
DEVICE=eno1
ONBOOT=yes
BRIDGE=br0
NM_CONTROLLED=no
```

- Create `/etc//etc/sysconfig/network-scripts/ifcfg-br0`

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
DELAY=0
NM_CONTROLLED=no
```

- Restart network

```
systemctl restart network
```

Connect to a remote libvirtd via ssh

```
virsh -c qemu+ssh://username@host/system
```

Quickinstall

- This needs `libguestfs-tools`
- Build a new disk

```
virt-builder centos-7.0 -o mydisk.img --format qcow2 --size 20G \
             --root-password file:/tmp/rootpw \
             --update \
             --run-command 'rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/
↪epel-release-6-8.noarch.rpm' \
             --install cloud-utils,cloud-init
```

- Install a whole cluster

```
for i in {1..3}; do cp centos7.img node$i.img; virt-install --import --name "Node$i" -
↪-os-type=linux --ram=512 --disk path=node$i.img,size=2; done
```

Create a machine

- Configure bridged interface br0
- Create a virtual machine with 20 gb disk and 512 mb ram
- Via ISO file

```
virt-install --name="TestVM" --os-type=linux --ram=512 --disk path=test-vm.img,
↪size=20 --cdrom <path-to-iso-file>
```

- Using PXE

```
virt-install --name="TestVM" --os-type=linux --os-variant=rhel6 --network bridge=br0,
↪mac=aa:bb:cc:aa:bb:cc --ram=512 --disk path=test-vm.img,size=20 --pxe
```

Remove a machine

```
virsh undefine <machine-name>
```

List all virtual machines

```
virsh list
```

Start or stop a machine

```
virsh start <machine-name>
virsh shutdown <machine-name>
virsh destroy <machine-name>
```

- To shutdown a machine it must have acpid running

Autostart a machine

```
virsh autostart <machine-name>
```

Info about a machine

```
virsh dominfo <machine-name>
```

Info about host system

```
virsh nodeinfo
```

Connect to a machine

- `virt-viewer` or `virt-manager`

Rename a machine

```
virsh dumpxml <machine-name> > muh.xml
<edit muh.xml, change the name>
virsh undefine <machine-name>
virsh define muh.xml
```

Attach a cdrom image

```
virsh attach-disk <machine-name> <iso-file> hdc --type cdrom --mode readonly
```

Update boot order

- First dump machine settings as XML

```
virsh dumpxml <machine-name> > blah.xml
```

- Edit XML file
- Update machine settings

```
virsh define blah.xml
```

Configure RAM

```
virsh setmem <machine-name> <kbyte>
```

Configure number of CPUs

```
virsh setvcpus <machine-name> <nr>
```

Resize disk

- Switch off the machine
- and convert old image to new one and expand sda2 to max size

```
virt-resize --expand /dev/sda2 old-disk.img new-disk.img
```

Update a machines config

```
virsh edit <machine-name>
```

Backup

- Save a machines RAM state to a file

```
virsh save <machine-name> <file>
```

- Take a snapshot of disk and ram (must be supported by disk image format e.g. qcow2 and this will PAUSE the machine if ram is backuped! use –disk-only to avoid this)

```
virsh snapshot-create-as <machine-name> <snapshot-name>
```

- or by using qemu tools (but only when vm is off!)

```
qemu-img snapshot -c my-backup disk.img
qemu-img snapshot -l disk.img
```

- Extract snapshot of qcow2 image file

```
qemu-img convert -f qcow2 -s <snapshot> -O qcow2 <image_file> <backup_file>
```

- Another possibility is to install libguestfs-tools and create a tar archive of /

```
virt-tar -z -x <machine_name> / machine-backup.tgz
```

Restore

```
virsh snapshot-list <machine_name>
virsh snapshot-revert <machine_name> <snapshot>
```

Disk tricks

- Install libguestfs-tools

```
virt-df
virt-df -d <machine_name>
```

- Get content of a file

```
virt-cat -d <machine_name> <filename>
```

- Edit a file (vm must be off)

```
virt-edit -d <machine_name> <filename>
```

- Or even get a shell on a disk image

```
guestfish \
          add disk.img : run : mount /dev/vg_guest/lv_root / : \
                    write /etc/resolv.conf "nameserver 8.8.8.8"
```

Change root password with guestfish

- Generate a new password hash

```
openssl passwd -1 topsecretpassword
```

- Edit shadow file on image

```
guestfish --rw -a <imagefile>
><fs> run
><fs> list-filesystems
/dev/sda1: ext4
><fs> mount /dev/sda1 /
><fs> vi /etc/shadow
```

Convert VirtIO to IDE disk and vice versa

- Make sure the machine is powered off

```
virsh edit <machine-name>
```

- For IDE disk

```
<disk type='file' device='disk'>
   <driver name='qemu' type='raw'/>
   <source file='/whatever.img'/>
   <target dev='hda' bus='ide'/>
   <address type='drive' controller='0' bus='0' target='0' unit='0'/>
 </disk>
```

- For VirtIO disk

```
<disk type='file' device='disk'>
  <driver name='qemu' type='raw' cache='writethrough'/>
  <source file='/whatever.img'/>
  <target dev='vda' bus='virtio'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
</disk>
```

- Afterwards update the systems `/etc/fstab`

```
virt-edit /path/to/image-file /etc/fstab
```

Cloning

- Will copy a whole machine and its properties and gives it a new mac address
- The machine must be switched off

```
virt-clone -o <machine-name> -n <new-machine-name>
```

Migration

- By default, migration only transfers in-memory state of a running domain (memory, CPU state, ...). Disk images are not transferred during migration but they need to be accessible at the same path from both hosts.
- Live migration needs shared network storage via NFS, iSCSI, GFS2 or Fibre Channel

```
virsh migrate --live <machine-name> qemu://example.com/system
```

Performance overview

- Use `virt-top`

Performance tuning

- Use virtio driver for disk and net this will give a machine direct hardware access (no emulation - only for linux guests)
- Maybe you have to load the kernel modules

```
modprobe virtio_blk
modprobe virtio_net
modprobe virtio_pci
```

- If one dont want to use snapshots use *raw* as image type
- Use *Writethrough* as caching type
- Use MacVtab bridge as network device with virtio model
- Use Spice and QXL driver for display

Grant normal user permission to qemu:///system

- Create file `/etc/polkit-1/localauthority/30-site.d/libvirt.pkla`

```
[User update perms]
Identity=unix-user:basti
Action=org.libvirt.unix.manage
ResultAny=no
ResultInactive=no
ResultActive=yes
```

Scripting with Python2

- Just a sample script to shutdown all active instances and boot all that were inactive

```
import libvirt

#conn=libvirt.open("qemu:///system")
conn = libvirt.open("qemu+ssh://root@127.0.0.1/system")

print "Active instances"
active_instances = []

for id in conn.listDomainsID():
  instance = conn.lookupByID(id)
  instance_name = instance.name()
  active_instances.append(instance_name)
  print "Deactivating ", instance_name
  instance.destroy()

print "Activating inactive instances"
inactive_instances = [instance for instance in conn.listDefinedDomains() if instance␣
↪not in active_instances]

for instance_name in inactive_instances:
  print "Activating ", instance_name
  instance = dom = conn.lookupByName(instance_name)
  instance.create()

conn.close()
```

- A script to create / delete a new instances

```
import sys
import libvirt

dom_name = "testme"
dom_mem = 512
dom_cpu = 1
dom_disk = "/data/virtualbox/centos64.img"
qemu_disk_type = "raw"

if len(sys.argv) < 2:
  print sys.argv[0], " up/down"
  sys.exit(0)

conn = libvirt.open("qemu+ssh://root@127.0.0.1/system")
```

```
if sys.argv[1] == "down":
  dom = conn.lookupByName(dom_name)

  if dom:
      dom.undefine()
  else:
      print "Cannot find domain ", dom_name
      conn.close()
      sys.exit(1)
else:
  xml = """<domain type='kvm'>
    <name>""" + dom_name + """</name>
    <memory unit='KiB'>""" + str(dom_mem * 1024) + """</memory>
    <vcpu placement='static'>""" + str(dom_cpu) + """</vcpu>
    <os>
      <type arch='x86_64' machine='rhel6.4.0'>hvm</type>
      <boot dev='hd'/>
    </os>
    <features>
      <acpi/>
      <apic/>
      <pae/>
    </features>
    <clock offset='utc'/>
    <on_poweroff>destroy</on_poweroff>
    <on_reboot>restart</on_reboot>
    <on_crash>restart</on_crash>
    <devices>
      <emulator>/usr/libexec/qemu-kvm</emulator>
      <disk type='file' device='disk'>
        <driver name='qemu' type='""" + qemu_disk_type + """' cache='none'/>
        <source file='""" + dom_disk + """'/>
        <target dev='hda' bus='ide'/>
        <address type='drive' controller='0' bus='0' target='0' unit='0'/>
      </disk>
      <disk type='block' device='cdrom'>
        <driver name='qemu' type='raw'/>
        <target dev='hdc' bus='ide'/>
        <readonly/>
        <address type='drive' controller='0' bus='1' target='0' unit='0'/>
      </disk>
      <controller type='usb' index='0'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
      </controller>
      <controller type='ide' index='0'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
      </controller>
      <interface type='bridge'>
        <mac address='52:54:00:f8:56:2a'/>
        <source bridge='br0'/>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
      </interface>
      <serial type='pty'>
        <target port='0'/>
      </serial>
      <console type='pty'>
        <target type='serial' port='0'/>
```

```
      </console>
      <input type='mouse' bus='ps2'/>
      <graphics type='spice' port='5900' autoport='yes' listen='127.0.0.1'>
        <listen type='address' address='127.0.0.1'/>
        <clipboard copypaste='no'/>
        <image compression='auto_glz'/>
      </graphics>
      <video>
        <model type='qxl' ram='65536' vram='65536' heads='1'/>
        <alias name='video0'/>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
      </video>
      <memballoon model='virtio'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
      </memballoon>
    </devices>
  </domain>"""

  #print xml
  conn.createXML(xml, libvirt.VIR_DOMAIN_START_AUTODESTROY)

conn.close()
```

- Accessing virtual disks with guestfs

```
import guestfs

gfs = guestfs.GuestFS()
gfs.add_drive_opts(dom_disk)
gfs.launch()
root_device = None

for root in gfs.inspect_os():
  for mountpoint in gfs.inspect_get_mountpoints(root):
    if mountpoint[0] == "/":
      root_device = mountpoint[1]
      break

if root_device:
  gfs.mount(root_device, "/")
  gfs.sh("dhclient eth1 && yum install puppet")
else:
  print "Cannot find root device :("

gfs.umount_all()
```

Failover

- http://code.google.com/p/ganeti/

Troubleshooting

- Intel virtualisation support must be activated in bios to use kvm
- Maybe Vbox modules should be unloaded

- Use *virt-rescue* (from libguestfs-tools) as live-rescue system

```
virt-rescue -d <machine_name>
```

Ovirt

Setup controller node

- Leave all default settings except password and ISO domain (5 after password)

```
yum localinstall -y http://resources.ovirt.org/pub/yum-repo/ovirt-release35.rpm
yum install ovirt-engine-reports-setup ovirt-engine-dwh-setup ovirt-engine
engine-setup
```

- Click on Cluster -> Edit and Check "Enable Gluster Service"
- Choose cpu architecture x86_64
- Click on Storage -> New Domain and add an NFS path

Setup hypervisor node

```
yum localinstall -y http://resources.ovirt.org/pub/yum-repo/ovirt-release35.rpm
```

- Login to Ovirt Engine Web frontend as admin user
- Open Cluster tab and click on Default -> Hosts -> New
- Fill out General form
- Click in the right lower corner of the arrows to see event logs

Import an ISO image

```
engine-iso-uploader -r <ovirt-server> -i iiScratch_iso upload <ISO_FILENAME>
```

Vagrant

Installation

- Via RPM / Deb see http://downloads.vagrantup.com/
- Via GEM

```
gem install Vagrant
```

Libvirt support

- You need at least version 1.1
- Install libvirt plugin

```
vagrant plugin install vagrant-libvirt
```

- Edit Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.define :test_vm do |test_vm|
    test_vm.vm.box = "centos64"
    test_vm.vm.network :private_network, :ip => '10.20.30.40'
  end

  config.vm.provider :libvirt do |libvirt|
    libvirt.driver = "qemu"
    libvirt.host = "localhost"
    libvirt.connect_via_ssh = true
    libvirt.username = "root"
    libvirt.storage_pool_name = "default"
  end
end
```

- To start it

```
vagrant up --provider=libvirt
```

- Make libvirt default

```
export VAGRANT_DEFAULT_PROVIDER=libvirt
```

Create custom libvirt box

- https://github.com/pradels/vagrant-libvirt/tree/master/example_box

Get boxes

- http://www.vagrantbox.es

Virtualbox

Running on Grsec kernel

If someone out there should face the same problems as I did here are the Grsecurity features I had to disable to get VirtualBox running correctly:

- grsecurity -> address space protection -> hide kernel symbols
- grsecurity -> filesystem protections -> proc restrictions
- pax -> non-executable pages -> enforce non-executable kernel pages
- pax -> miscellaneous hardening features -> prevent invalid userland pointer dereferences

Headless Server

- http://0x7e.org/blog/2011/10/15/virtualbox-on-a-headless-server/

PXE Boot

- Needs extenstions
- On older versions only via bridged network support

Automation

- http://vagrantup.com/

Download disk images

- http://virtualboximages.com/

CHAPTER 25

Programming

AWK

Useful variables

Variable	Description
FS	Field seperator
RS	Records seperator
NR	nr of records
NF	nr of fields
OFS	output field seperator
ORS	output record seperator

Split lines in file on delimiter after matching it to regexp

```
cat some_file.txt | awk -F ': ' '/^regexp/ {print $2 " - " $1}'
```

Match a colum

```
awk '$1 ~ /^regexp/ {print $2}'
```

Multiple matches

```
awk '/regexp1/ {print $1} /regexp2/ {print $2}'
```

Conditional statements

```
awk '{if($1 == "muh") && ($3 != "maeh") { print $2 } }'
```

Counter

```
cat some_file | awk 'BEGIN { x=1 } { print x; x += 1 }'
```

- Or just use the variable *NR* the line counter (counts all lines not only matching ones)

Sum up a column

```
cat some_file.txt| awk '{ sum+=$3} END {print sum}'
```

String replacement

```
cat some_file.txt | awk 'sub(/regexp/, "replace", $1)'
```

Defining external variables / Limiting

```
cat some_file.txt| awk -v LIMIT=30 '{a=$1; sub(/user/, "", a); if(a <= LIMIT){print}}'
```

CSS3

Rounded corners

```
input {
  border-radius: 5px;
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
}
```

Box Shadow

```
#content {
  -moz-box-shadow: 5px 5px 5px #ccc;
  -webkit-box-shadow: 5px 5px 5px #ccc;
  -o-box-shadow: 5px 5px 5px #ccc;
  -box-shadow: 5px 5px 5px #ccc;
}
```

Text Shadow

```
#content {
  text-shadow: 2px 2px 2px #ccc;
}
```

Rotations

```
#content {
  -moz-transform: rotate(-2.3deg)
  -o-transform: rotate(-2.3deg)
  -webkit-transform: rotate(-2.3deg)
  -ms-transform: rotate(-2.3deg)
  -transform: rotate(-2.3deg)
}
```

Transparency

```
#content {
   background-color: rgba(255, 255, 255, 0.80)
}
```

Striped tables

```
tr:nth-of-type(even) {
  background-color: white;
}

tr:nth-of-type(odd) {
  background-color: black;
}
```

Last line bold

```
tr:last-child {
  font-weight: bolder;
}
```

Two column content

```
#content {
  -moz-column-count: 2;
  -webkit-column-count: 2;
  -moz-column-gap: 20px;
  -webkit-column-gap: 20px;
}
```

Django

Admin

Prefill an admin field

- In ModelAdmin do

```
```

prepopulated_fields = {“slug”: (“title”,)}

WYSIWYG-TextField-Editor

- In admin.py

```
ModelAdmin:
    class Media:
        js = ('scripts/libs/tinymce/tiny_mce.js',
              'scripts/textareas.js',)
```

- Copy http://www.tinymce.com/ to static/scripts/lib
- Create static/scripts/textareas.js

```
tinyMCE.init({
      mode : "textareas",
      theme : "advanced",
  theme_advanced_buttons1 : "bold,italic,underline,separator,undo,redo,link,unlink,
↪anchor,separator,cleanup,help,separator,code",
  theme_advanced_buttons2 : "",
  theme_advanced_buttons3 : ""
});
```

- In the template remember to use |safe filter

AJAX

- http://dajaxproject.com/

CMS

Dynamically create a menu

- Create menu.py with following content

from menus.base import NavigationNode from cms.menu_bases import CMSAttachMenu from menus.menu_pool import menu_pool from django.core.urlresolvers import reverse from myapp.models import Category

class CategorieMenu(CMSAttachMenu): name = “Categories Menu”

def get_nodes(self, request): nodes = []

for category in Category.objects.all():

nodes.append(NavigationNode(category.name,

reverse(“category_list”, kwargs={“category”: category.name}), category.pk,

))

return nodes

menu_pool.register_menu(CategorieMenu)

Forms

- Overwrite only the widget of a modelform

```
class MyForm2(forms.ModelForm):
   def __init__(self, *args, **kwargs):
       super(MyForm2, self).__init__(*args, **kwargs)
       self.fields['myfield1'].required = True
       self.fields[‘myfield1’].widget = forms.Textarea(attrs {‘class’:’myclass’,})
   class Meta:
       model = models.MyModel
```

- Custom error message

```
class MyForm2(forms.ModelForm):
   def __init__(self, *args, **kwargs):
       super(MyForm2, self).__init__(*args, **kwargs)
       self.fields['rating'].error_messages['required'] = 'I require that you fill␣
↪out this field'
```

- Custom validator

```
def clean_myfield(self):
  if self.cleaned_data["myfield"] < 10:
    raise forms.ValidationError("MyField must be greater than 10")

  return self.cleaned_data["myfield"]
```

- Loop over all form fields

```
{% for field in form %}
  {{ field.label_tag }} {{ field }}
{% endfor %}
```

- Display form value without None

```
{{ form.field.value|default_if_none:"" }}
```

- Request if a field is required

```
{% if form.field_name.field.required %}
```

- Display error without li-tags

```
{{ field.errors|striptags }}
```

- Create Form from Model with https://github.com/bradleyayers/django-tables2/

Misc

- http://dabodev.com/ - Django for desktop
- http://www.b-list.org/weblog/2006/jun/13/how-django-processes-request/
- Another CMS http://ridethepony.org/
- django-annoying render_to("template") decorator

Modelling

Automatic created and modified timestamps

```
from django_extensions.db.models import TimeStampedModel
class MyModel(TimeStampedModel)
```

Tree-based models

- https://github.com/django-mptt/django-mptt
- get ancestors / get_descendants

```
mptt_model.get_ancestors()
mptt_model.get_descendants()
```

Get UML for models

```
django-admin.py graph_models app -o app.png
```

Introspection

- Get all attributes

```
User._meta.fields()
```

- Check attribute type

```
isinstance(field, models.ForeignKey)
```

- Where is the foreign key pointing to?

```
field.rel.to
```

- More on http://www.b-list.org/weblog/2007/nov/04/working-models/
- Filter dynamically

```
search_column = "name_" + lang
search_value = self.kwargs.get("category")
Category.objects.filter(**{ search_column: search_value })
```

Multilang

Convert PO-Files to CSV

- http://translate.sourceforge.net/wiki/toolkit/using_csv2po

```
po2csv -i locale/en/LC_MESSAGES/django.po -o translations.csv
iconv -f UTF-8 -t WINDOWS-1252 translations.csv > translations_new.csv
```

Make Messages without Django-Cms etc

```
django-admin.py makemessages -l de -e html,txt -i *cms/* -i *menus/* -i *cmsplugin_
↪filer_file/* -i *cmsplugin_filer_folder/* -i *cmsplugin_filer_image/* -i *cmsplugin_
↪filer_video/* -i *easy_thumbnails/* -i *filer/* -i *mptt/*
```

- django-modeltranslation
- PO-File Admin Editor https://github.com/mbi/django-rosetta

REST

- https://github.com/toastdriven/django-tastypie

Search

- http://haystacksearch.org/

Security

- http://pypi.python.org/pypi/django-passwords/

Create a new user

```

```

from django.contrib.auth.models import User user = User.objects.create_user('bart', 'bart@simpsons.com', 'eatmyshorts')

User profile

- member/models.py

```

```

class UserProfile(models.Model): user = models.OneToOneField(User) language = models.CharField(max_length=2, default=settings.LANGUAGE_CODE)

def user_registered_handler(sender, **kwargs): user = kwargs.get('user') language = kwargs.get("language")

if not language: language = settings.LANGUAGE_CODE

if user: profile, created = UserProfile.objects.get_or_create(user=user) profile.language = language

profile.save()

user_registered_signal.connect(user_registered_handler)

- Register profile in settings

AUTH_PROFILE_MODULE = “member.UserProfile”

Edit user profile in admin

- member/admin.py

class UserProfileInline(admin.StackedInline): model = UserProfile max_num = 1 can_delete = False

class UserAdmin(AuthUserAdmin): inlines = [UserProfileInline]

unregister old user admin admin.site.unregister(User) # register new user admin admin.site.register(User, UserAdmin)

Custom Login

- urls.py

from django.contrib.auth.views import login, logout

url(r’member/login/$’, login, name=”login”), url(r’member/logout/$’, logout, {“template_name”: “mymodule/logout.html”}, name=”logout”),

- settings.py

LOGIN_REDIRECT_URL = “/secure” LOGOUT_REDIRECT_URL=”/”

South

copy one field to another

- django-admin.py datamigration app migration_name

*. edit migration

```
def forwards(self, orm):
    "Write your forwards methods here."
    for mod in Model.objects.all():
        mod.field_new = mod_field_old
        med.save()
```

```
def backwards(self, orm):
    "Write your backwards methods here."
    raise RuntimeError("Cannot reverse this migration.")
```

Testing

- https://github.com/ericholscher/django-test-utils
- https://github.com/kmmbvnr/django-jenkins
- https://github.com/gregmuellegger/django-autofixture
- http://readthedocs.org/docs/django-test-utils/en/latest/testmaker.html
- Get emails to stdout

```
EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
```

- Change language for test client

```
self.client.cookies["django_language"] = 'fr'
```

- Test an AJAX request

```
client.get('/url/',{},HTTP_X_REQUESTED_WITH='XMLHttpRequest')
```

Erlang

Setup a cluster on a LAN

- Make sure UDP / TCP port 4369 is open

```
erl -name muh -setcookie somesecret
erl -name maeh -setcookie somesecret
```

Setup a cluster over SSH

- On master node (make sure your user can connect to the slaves by passwordless ssh key e.g. ssh-copy-id)

```
erl -rsh ssh -sname master
```

- Compile and load cluster module on master

```
-module(cluster).
-export([slaves/1]).

slaves([]) -> ok;
slaves([Host|Hosts]) ->
    {ok, Node} = slave:start_link(Host, "slave", "-rsh ssh"),
    io:format("Erlang node started = [~p]~n", [Node]),
    slaves(Hosts).
```

- Start the cluster from master node

```
erl> cluster:slaves(["node1", "node2", "node3"]).
```

List all processes

```
i().
regs().
```

List all connected nodes

```
nodes().
```

Spawn process on remote node

```
rpc:spawn(Node, Mod, Fun, Args).
```

Call function on remote node

```
rpc:call(Node, Pid, Fun, Args).
```

Call a function on all connected nodes

```
rpc:multicall(nodes(), Mod, Fun, Args).
```

Concurrent program template

```
-module(ctemplate).
-compile(export_all).

start() -> spawn(ctemplate, loop, []).

rpc(Pid, Request) ->
    Pid ! {self(), Request},
    receive
        {Pid, Response} -> Response.
    end.

loop() ->
    receive
        Any ->
            io:format("Help me do something"),
            loop()
     end.
```

Git

Repository

- Normal

```
git init
```

- Bare repository (without working tree / used to clone)

```
git init --bare
```

Commit

- Append to last commit

```
git commit --amend
```

- Revert a commit

```
git revert <version>
```

Edit a commit

```
git rebase -i <version>
```

- This opens an editor -> change the pick into edit

```
git add -A
git commit -C HEAD
git rebase --continue
git push
```

Logging

- Show changes

```
git log -p
```

- Show changes for one file or dir

```
git log -- *.py
```

- Show only the last 3 entries

```
git log -3
```

- Show only commits from the last 2 weeks

```
git log --since=2.weeks
```

- Unpushed commits

```
git log origin..
```

- Show log for one file

```
git log -- [filename]
```

- Show files of a Commit

```
git show --name-only <revision>
```

Branching

- Create branch

```
git checkout -b <branch>
git push origin <branch>
```

- Checkout a branch

```
git pull
git checkout <branch>
```

- Delete branch

```
git push origin :branch
```

- Show diff between two branches

```
git diff master..<branch> --raw
```

- List all branches on remote

```
git branch -a
```

Merging

- Merge everything

```
git checkout <branch>
git merge master
git checkout master
git push origin <branch>
```

- Merge just one commit

```
git cherry-pick <commit-id>
```

Tagging

- Create a tag

```
git tag <tag_name>
```

- Create a tag with a comment

```
git tag -m <comment> <tag_name>
```

- Show all tags

```
git tag
```

- Show one tag

```
git show <tag_name>
```

- Delete a tag

```
git tag -d <tag_name>
```

Working with older versions

- Get latest version of one file

```
git checkout <file>
```

- Show specific version of one file

```
git show <version>:<file>
```

- Get specific version of one file

```
git checkout <version> <file>
```

- Delete all changes over a specific version

```
git reset --hard <version>
```

- Delete just the changes of a specific commit

```
git revert <commit-id>
```

Using the stash

- Save changes to the stash

```
git stash
```

- Show stashes

```
git stash list
```

- Show changes of a stash

```
git stash show stash@{0}
```

- Apply latest stash changes and delete the stash

```
git stash pop
```

- Apply a specific stash without deleting it

```
git stash apply stash@{0}
```

- Delete a stash

```
git stash drop stash@{0}
```

- Wipe all stashes

```
git stash clear
```

Handling remote repositories

- Add a remote

```
git remote add origin git://domain.tld/repo.git
```

- Show infos about remotes

```
git remote show
git remote show origin
```

Ignore existing file (if gitignore doesnt ignore)

```
git update-index --assume-unchanged <file>
```

Check consistency

```
git fsck --progress
```

Dangling commits

- A commit that is not linked to a branch or tag

```
git reflog expire --expire=now --all
git gc --prune=now --aggressive
```

Convert normal repo to bare

```
git clone --bare -l <normal_repo> <bare_repo>
```

Git over HTTP

```
git clone --bare /git/test
touch git-daemon-export-ok
↪                                                                    |
git config --file config http.receivepack true
↪                                                                    |
git config core.sharedRepository
↪                                                                    |
chown apache:apache -R /git/test
```

Apache config for gitweb

```
<VirtualHost *:80>
  ServerName git.server.net
  ServerAlias git

  DocumentRoot "/var/www/git"
  Timeout 2400

  LogFormat   combinedssl
  LogLevel    info
  ErrorLog    /var/log/httpd/git-error.log
  TransferLog /var/log/httpd/git-access.log

  RewriteEngine On
  RewriteLog "/var/log/httpd/git-rewirte.log"
  RewriteLogLevel 5
  RewriteCond %{QUERY_STRING} ^.*p=(.*?)(\.git|;|&|=|\s).*
  RewriteRule (.*)/$ http://git.server.net$1?

  SetEnv GIT_PROJECT_ROOT /git
  SetEnv GITWEB_CONFIG /etc/gitweb.conf
  Alias /git/static/ /var/www/git/static/
  AliasMatch ^/git/(.*/objects/[0-9a-f]{2}/[0-9a-f]{38})$          /var/www/git/$1
  AliasMatch ^/git/(.*/objects/pack/pack-[0-9a-f]{40}.(pack|idx))$ /var/www/git/$1
  ScriptAliasMatch \
                  "(?x)^/git/(.*/(HEAD | \
                  info/refs | \
                  objects/info/[^/]+ | \
                  git-(upload|receive)-pack))$" \
                  /usr/bin/git-http-backend/$1
  ScriptAlias /git/ /var/www/git/gitweb.cgi/
</VirtualHost>
```

Subversion over git

- You can use a subversion repo like a remote git repo
- Clone it

```
git svn clone <svn-url>
```

- Pull and push changes

```
git pull origin master
git svn push origin master
```

Misc

- Diff with meld http://nathanhoad.net/how-to-meld-for-git-diffs-in-ubuntu-hardy
- Code Review with ReviewBoard http://ericholscher.com/blog/2011/jan/24/using-reviewboard-git/
- Webfrontend http://gitorious.org/ or https://github.com/takezoe/gitbucket

HTML5

Basics

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5 / CSS3 Spielplatz</title>
  </head>
  <body>
    <header>Dies ist die &UUml;berschrift</header>
    <nav><a>Link1|</a><a>Link2</a></nav>

    <section>
      <header>Artikel&uuml;berschrift</header>
      <p>Ganz viel tolles Bla Bla</p>
      <aside>Eine Randnotiz</aside>
    </section>

    <footer>Dies ist die Fussnote</footer>
  </body>
</html>
```

Forms

```
<form>
  <fieldset>
    <legend>HTML5 form</legend>
    <p>
      <label for="email">E-Mail</label>
      <input type="email" placeholder="E-Mail" id="email" />
    </p>
    <p><input type="url" placeholder="URL" /></p>
    <p><input type="tel" placeholder=="Phone"/></p>
    <p><input type="search" placeholder="Searchfield" /></p>
    <p><<input type="range" placeholder="Range" /></p>
    <p><input type="number" placeholder=="Number" /></p>
    <p><input type="date" placeholder="Date" /></p>
    <p><input type="datetime" placeholder="DateTime" /></p>
    <p><input type="color" placeholder="color" autofocus /></p>
```

```
  </fieldset>
</form>
```

Audio

```
<audio controls>
  <source src="track.mp3" type="audio/mpeg" />
</audio>
```

Video

```
<video controls>
  <source src="movie.mp4" />
</video>
```

Jenkins

Extra tmp dir for jenkins

- -Djava.io.tmpdir=/var/lib/jenkins/tmp/

Lisp

Basics

Overview

- (print “Hello world”)
- (format “%s” “Hello world”)
- create a global variable with `defvar` or `defparameter`
- (setq name “balle”) is the same as (set ‘name “balle”)
- local variables

```
(let ((var1 value) (var2 value2) do-something)
```

- use `let*` if vars must know each other during declaration
- var returns the value of var
- ‘var returns a reference (the symbol) of var
- symbols are always treated like uppercase
- quote or ‘ suspresses evaluation
- ‘ suspresses evaluation for all expression but prefixed with ,

- nil and () are the same
- with type-of you get the type of an object

Lists

- `car` returns the first element of a list
- `cdr` returns all but the first element of a list
- `cons` adds a element in front of list by creating a new list
- `nthcdr` exec cdr nth times on list
- `nth` returns the nth element of the list staring with 0
- `setcar` replace first element of list
- `setcdr` replace all but first element of list
- `append` adds to a list by copying the first list
- `nconc` adds the second, third etc list to the first
- `push` insert an element at the beginning of the list
- `pop` returns and removes the first element
- `(member what list)` check if what is in list
- `(position what list)` get position of what in list
- `(remove what list)` remove element what from list (returns new list)
- `(delete what list)` remove element what from list directly
- `(number-sequence 1 9)` returns a list with numbers from 1 to 9
- `first` or `car` returns the first element of the list
- `rest` or `cdr` returns the rest of the list
- `cadr` is the same as (car (cdr alist))
- `cdar` is the same as (cdr (car alist))
- (:muh 1 :maeh 2) is a property list
- plist (property list) is a list with (:key value) pairs
- (getf list :keyword) returns value of keyword in a plist
- alist is a plist where you can also lookup by value using (assoc 'what-to-find my-list)
- plists and alists are still handled sequencially
- `set-difference` tells which items are in one list but not in another
- `intersection` tells which items are in both Lists
- `remove-duplicates` creates a unique list out of two or more lists
- `(mapcar #'function alist)` applys function on every list element and returns new list
- `(mapc #'function alist)` applys function on every list element without returning a new list

Sequence functions

- `length` returns the number of the lists elements
- `(map 'type #'function aseq)` to loop through a sequence create a new 'type and apply function on ever element
- `(reduce function aseq)` calls function with next item and previous return value of function and thus reduces a sequence to one value
- `(apply function aseq)` Call function with remaining args, using last arg as list of args

hashes

- #s(hash-table size 30 data (key1 val1 key2 300))
- gethash key table &optional default
- puthash key value table

Arrays

- make a resizeable array (init size is 5)

```
(make-array 5 :fill-pointer 0 :adjustable t)
(vector-push-extend 'new-stuff my-array)
(aref my-array 3)
```

Structures

```
(defstruct person surname firstname age)
(defvar hans (make-person :surname wurst :firstname hans :age 35))
(person-age hans)
```

functions

```
(defun hello (name)
"function to say hello to someone or something"
  (print (concat "Hello " name)))
(hello "world")
```

- parameter after `&optional` are optional
- default values for parameters

```
(defun some-func (a &optional (b 10)))
```

- define keyword arguments

```
(defun hello (&key name "world" by default))
```

- #' or function suspresses evaluation of functions (aka returns pointer)
- use lambda to define anonymous functions

- flet declares local functions
- labels command is for flet what let* is for let (functions know each other during definition)

control structures

- equal check euqalness eq identity
- (eq “abc” “abc”) -> nil
- (equal “abc” “abc”) -> t
- check numbers with =
- (= 1 1) -> t
- check symbols with eq
- check everything else with equal
- if else

```
(if (eq "abc" "bcd")
  (progn do-this-if-cond-is-true)
  (progn do-this-if-cond-is-false)
)
```

- `when` is an if without else that can handle multiple statements
- cond is a list of checks like if, else if, else if, else

```
(cond ((equal var value)
       (do-something))

      ((equal var value2)
        (do-something))

      (t
        (do-something))
)
```

- there is also a switch case

```
(case person
     ((hans)
        '(give him some food))
     ((wurst)
        '(run away screaming))
     ((otherwise)
        '(be cool)))
```

- to compare strings in a case form

```
(case (find-symbol (string-upcase person) :keyword)
  (:hans
     '(give him some food))
  (:wurst
     '(run away screaming))
  (otherwise
     '(be cool)))
```

Loops

- simple while

```
(while (< (count) 10)
  do-something
)
```

- iterate each item of a list

```
(dolist (item list)
   (print item))
```

- or

```
(loop for i in '(1 2 3) do
  (print i))
```

```
(loop for i from min to max by step)
```

- iterate over key, value pairs of a hash

```
(loop for k being the hash-key using (hash-value v) of h do (format t "~a ~a~%" k v))
```

Store state of interpreter in file

- SBCL

```
(SAVE-LISP-AND-DIE "foo.core")
```

- Load with

```
sbcl --core foo.core
```

- CLISP

```
(saveinitmem "foo.mem")
```

- Load with

```
clisp -M foo.mem
```

Scripting

- SBCL

```
#!/usr/bin/sbcl --script

(require ".sbclrc")
```

- Disable style warnings in SBCL

```
(declaim #+sbcl(sb-ext:muffle-conditions style-warning))
```

- CLISP

```
#!/usr/local/bin/clisp

(require ".clisprc.lisp")
```

Installing modules

- Install http://www.quicklisp.org/beta/

```
(ql:quicklib "module")
```

Loading modules

- load is used to load a single lisp file
- require is used to load modules that can consist of more than one file

Whats the difference between packages, systems and modules?

- http://weitz.de/packages.html
- Packages are namespaces (like in Perl)
- A system is a bunch of code with instructions to install them plus their dependencies
- A module is something you can load to your lisp code

Channel

- *standard-output*
- *error-output*, *debug-io* and *trace-output*
- *query-io* for user input

Redirect stdout

```
(let ((*standard-output* (make-broadcast-stream)))
  (app:noisy-code))
```

Debugging

- (trace) will trace function calls
- (step) through function calls
- (break) sets a break point

Links

- http://ghostopera.org/blog/2012/06/24/the-newbie-guide-to-common-lisp/
- http://psg.com/~dlamkins/sl/contents.html - Successful lisp

CLOS - Common Lisp Object System

Overview

- Its class based, but
- Methods are not bound to objects
- Objects are bound to methods
- It supports multiple inheritance
- There is no enforce encapsulation (like in python) everyone can directly access all slots (properties) of an object

Example

```
(defclass vehicle ()
  ((wheels)
   (color
    :initarg :black)))

(defclass automobile (vehicle)
  ((wheels
    :initarg :4)))

(defclass motorbike (vehicle)
  ((wheels
    :initarg :2)))

(defmethod drive ((vehicle automobile))
  (format *standard-output* "Driving a car~%"))

(defmethod drive ((vehicle motorbike))
  (format *standard-output* "Flying over the street with a motorbike~%"))

(defvar ferrari (make-instance 'automobile))
(defvar kawasaki (make-instance 'motorbike))

(drive ferrari)
(drive kawasaki)
```

Define Getter / Setter

- If you dont like to access slot directly using `(slot-value)`

```
(defclass vehicle ()
  ((wheels)
   (color
    :initarg :black
    :reader color
    :writer (setf color)
    :accessor color)))

(setf (color ferrari) 'red)
(print (color ferrari))
```

Print all slots

- SBCL

```
(mapcar #'sb-pcl:slot-definition-name (sb-pcl:class-slots (class-of object)))
```

- Clisp

```
(mapcar #'clos:slot-definition-name (clos:class-slots (class-of object)))
```

Database

CLSQL

- (ql:quickload "clsql")

```
(require "clsql")

(setq *conn*
    (clsql:connect '("127.0.0.1" "mydb" "user" "secret") :database-type :mysql))
(setq query "SELECT timestamp, value FROM data LIMIT 100;")

(loop for row in (clsql:query query :field-names t) do
   (format t "~A ~,5F~%" (nth 0 row) (nth 1 row))
```

CL-DBI

- (ql:quickload "dbi")

```
(require "dbi")

(setq *conn*
    (dbi:connect :mysql
                 :host "127.0.0.1"
                 :database-name "mydb"
                 :username "user"
                 :password "secret"))

(setq query (dbi:prepare *conn* "select timestamp, value from data LIMIT 100;"))
(setq result (dbi:execute query))

(loop for row = (dbi:fetch result)
  while row
  do (format t "~A ~,5F~%" (getf row :|timestamp|) (getf row :|value|)))
```

Mongo

- (ql:quickload "cl-mongo")

```
(use-package :cl-mongo)
(db.use "test")
(defvar *DOC* (make-document))
(add-element "key" "value" *DOC*)`
```

```
(db.insert "mycollection" *DOC*)
(db.find "mycollection" (kv "key" "value"))
```

ORM

- http://common-lisp.net/project/elephant/index.html
- http://common-lisp.net/project/cl-prevalence/

Date

- (ql:quickload "local-time")

Timestamp for now

```
(local-time:now)
```

- Get current month

```
(local-time:timestamp-month (local-time:now))
```

Universal time

```
(get-universal-time)
```

Unix time

```
(local-time:timestamp-to-unix (local-time:now))
```

Elisp

Basics

- Go get a REPL with M-x ielm
- (message "hello world")

Performance

- Use `(goto-char (point-min))` instead of `(beginning-of-buffer)`

Handling Buffers

- Get object of current and most recent buffer

```
(current-buffer)
(other-buffer)
```

- Exec commands on another buffer without changing to it

```
(set-buffer "name")
```

- Iterate over buffer list

```
(dolist (buffer buffer-list)
  (message (concat "BLAH " (buffer-name buffer))))
```

Switch to an existing buffer or create a new one

- To create a new one just enter a not existing name as string

```
(switch-to-buffer (other-buffer))
(switch-to-buffer-other-window "some-new-buffer")
```

Working with directories

```
(directory-files "~/")
```

Read file as list of lines

```
(defun read-lines (filePath)
  "Return a list of lines of a file at filePath."
  (with-temp-buffer
    (insert-file-contents filePath)
    (split-string (buffer-string) "\n" t)))
```

Cursor

- point-min ; beginning of buffer
- point ; current position
- point-max ; end of buffer

Shell

- Execute a shell command with `call-process-shell-command`

Function

- ask user for parameter in (interactive) function
- b existing buffer
- B buffer name but doesnt need to exist

- d position of point
- D directory
- f file
- r region
- s text

```
(interactive "fFilename:")
```

Run as script

```
emacs --script myscript.el
```

Profiling

```
profiler-start
profiler-report
profiler-stop
```

- You can expand lines with a + by pressing RET

Debugging

- `trace-function`
- `edebug-all-defs`
- `edebug-defun` behind function definition
- <SPC> - execute next expression
- n - next debuggable statement
- c - continue
- i - step into
- b - set breakpoint
- x - set conditional breakpoint
- u - unset breakpoint
- g - goto next breakpoint
- h - goto here
- d - backtrace
- e - eval expression e.g. (symbol-value ‘some-var)

Detect mode

```
(when (derived-mode-p 'emacs-lisp-mode) (message "MUH"))
```

Misc

- restore point and mark after executing do-something

```
(save-excursion do-something)
```

- Run a command if user is idle

```
(defun balle()
(message "MUH"))

(run-with-idle-timer 10 t 'balle)
```

- Use common lisp (Emacs 24.3 and later)

```
(require 'cl-lib)
(cl-defun print-name (&key first (last "?"))
```

- Earlier Emacs versions

```
(require 'cl)
(defun* print-name (&key first (last "?"))
```

- Common Lisp interpreter written in Emacs Lisp https://github.com/larsbrinkhoff/emacs-cl
- Namespaces https://github.com/Bruce-Connor/names
- http://www.emacswiki.org/emacs/ElispCookbook
- get integer value of char with ?

Filesystem

- (ql:quickload "cl-fad")
- http://weitz.de/cl-fad/

Get current directory

```
(truename ".")
```

List a directory

```
(directory (make-pathname :directory '(:absolute "var" "log") :name :wild :type␣
↪:wild))
```

Parsing

- Print filename and type

```
(pathname-name (pathname "/some/file.txt"))
(pathname-type (pathname "/some/file.txt"))
```

Test if file exists

```
(probe-file "/some/file")
```

Input / Output

- Write file

```
(let ((stream (open "/some/file/name.txt" :direction :output)))
  (format stream "hello world~%")
    (close stream))
```

- or using a macro
- :if-exists :supersede will override an existing file

```
(with-open-file (out filename
                 :direction :output
                 :if-exists :supersede)
   (with-standard-io-syntax
     (print *content* out))))
```

- Read file

```
(let ((in (open "/some/file/name.txt" :if-does-not-exist nil)))
  (when in
    (loop for line = (read-line in nil)
              while line do (format t "~a~%" line))
                (close in)))
```

- or using a macro

```
(with-open-file (in filename)
  (with-standard-io-syntax
    (setf *content* (read in)))))
```

- Ask user for Input

```
(defun prompt-read (prompt)
  (format *query-io* "~a: " prompt)
  (force-output *query-io*)
  (read-line *query-io*))
```

Macros

Overview

- Macros generate code

Example

```
(defmacro thread (&rest body)
  `(sb-thread:make-thread (lambda () (progn ,@body)) :name ,(symbol-name (gensym))))
```

- ‘ defines partial evaluation (only code prefixed by , will get executed)
- the @ before body handles the variable as a list
- gensym generates a random symbol name
- this macro can be used like

```
(thread (print "FOO") (print "BAR"))
```

Debugging

- macroexpand resolves the real code of a macro
- the macro must be prefixed by ‘ otherwise it gets executed

```
(macroexpand '(mymacro "param"))
```

Mail

- (ql:quickload “cl-smtp”)
- Send HTML mail with attachment

```
(cl-smtp:send-email
 +mail-server+
 "from-email@example.com"
 "to-email@example.com"
  "Subject"
   "<html><body>
   <p>
   Shiny <strong>h</strong><em>t</em><small>m</small>l.
   </p>
   <p>
   </body></html>"
    :extra-headers '(("Content-type" "text/html; charset=\"iso-8859-1\""))
    :attachments '("/path/to/file"))
```

Networking

Get hostname

- Local

```
(machine-instance)
```

- Remote

```
(require 'sb-bsd-sockets)
(in-package 'sb-bsd-sockets)
(host-ent-name (get-host-by-name "www.lisp.org"))
```

Example socket client

- (ql:quickload "usocket")

```
(require :usocket)
(setq sock (usocket:socket-connect "www.codekid.net" 80))
(format (usocket:socket-stream sock) "~A~C~C~A~C~C~A~C~C~C~C"
                                     "GET / HTTP/1.1"
                                     #\Return #\Newline
                                     "Host: www.codekid.net"
                                     #\Return #\Newline
                                     "Connection: close"
                                     #\Return #\Newline #\Return #\Newline)
(force-output (usocket:socket-stream sock))

(loop for line = (read-line (usocket:socket-stream sock))
   while line do
   (format t "~A~%" line))
```

SNMP

- (ql:quickload "snmp")

```
(snmp:with-open-session (s "192.168.1.1"
                           :version :v1
                           :community "public")
   (snmp:snmp-walk s "system"))
```

- For more see http://cl-net-snmp.svn.sourceforge.net/viewvc/cl-net-snmp/snmp/branches/6/doc/papers/ILC09-SNMP.pdf

Parallel

Start thread in SBCL

- Run a function in a thread

```
(sb-thread:make-thread #'(lambda () (a-function)) :name "a good name")
```

- List all running threads

```
(sb-thread:list-all-threads)
```

- Check if a thread is running

```
sb-thread:thread-alive-p
```

- Kill a thread

```
sb-thread:terminate-thread
```

Start thread in CLISP

- Run a function in a thread

```
(MT:MAKE-THREAD #'a-function :name "a good name")
```

- List all running threads

```
(mt:list-threads)
```

- Check if a thread is running

```
(mt:thread-active-p "thread name")
```

Bordeaux Threads

- Portable thread library
- http://trac.common-lisp.net/bordeaux-threads/wiki/ApiDocumentation

```
(ql:quickload "bordeaux-threads")
(bordeaux-threads:make-thread (lambda () (print "MUH")) :name "balle")
```

Clustering

- lfarm is a Common Lisp library for distributing work across machines using the lparallel API.
- https://github.com/lmj/lfarm

Profiling

- In Emacs

```
elp-instrument-function
elp-instrument-package
```

- In SBCL

```
(require :sb-sprof)

(defvar l '())
(dotimes (x 10000) (push (random 10) l))
(sb-sprof:with-profiling (:report :graph) (loop for x in l collect (* x 2)))
```

Refactoring

- https://github.com/emacsmirror/redshank
- http://www.foldr.org/~michaelw/emacs/redshank/

Regexp

- http://weitz.de/cl-ppcre/
- (ql:quickload :cl-ppcre)

Match

```
(if (cl-ppcre:scan "^[1-9]+$" "23-")
  (print "muh"))
```

Split

```
(cl-ppcre:split "\\s" "a cow says mooh")
```

Replace

```
(cl-ppcre:regexp-replace-all "\\d" "h4llo" "a\\1")
```

Shell

- Execute a command on shell

```
(require 'asdf)
(asdf:run-shell-command "touch foo")
```

- Reading output of command

```
(with-output-to-string (stream) (asdf:run-shell-command "ps ax" :output stream))
```

- or better

```
(ql:quickload "trivial-shell")
(setq output (trivial-shell:shell-command "df -k"))
```

Strings

Concatenate

```
(concatenate 'string "foo " bar)
```

Split

- (ql:quickload “cl-utilities”)

```
(cl-utilities:split-sequence #\newline some-string)
```

SLIME

Getting help

- , help - repl commands

- ctrl+c ctrl+d d - describe symbol
- ctrl+c ctrl+d h - hyperspec manual
- ctrl+c ctrl+d ctrl+z - apropos
- ctrl+c ctrl+d ctrl+p - list symbols of package

Evaluation

- ctrl+c ctrl+r - eval region
- ctrl+x ctrl+e - eval expression
- ctrl+c ctrl+z - jump to repl

Jumping around

- alt+x slime-selector - jump around in lisp / slime buffer
- alt+. - jump to source
- alt+x slime-list-callers
- alt+p / n - previous / next warning / error

Autocompletion

- ctrl+c TAB - autocomplete symbol
- ctrl+c ctrl+s - autocomplete form

Helpers

- ctrl+alt+q - format lisp expression
- ctrl+c ctrl+k - compile whole file
- ctrl+alt+t transpose expressions
- ctrl+alt+k kill expression
- ctrl+c return - macro expansion

Debugger

- ctrl+c alt+i - inspect value
- ctrl+c ctrl+t trace expression
- (untrace)
- (on frame in stacktrace) alt+x sldb-show-source
- i - inspect frame
- t - toggle details
- v - view source

- q - quit
- c - continue

Using Swank for remote lisp repl

- Starting swank server

```
(ql:quickload "swank")

(setf swank:*use-dedicated-output-stream* nil)
(setf swank:*communication-style* :fd-handler)
(swank:create-server :dont-close t)
```

- Setup SSH tunnel on local Emacs machine

```
ssh -L 4005:localhost:4005 me@remote-machine
```

- Reconnect slime with M-x slime-connect

Swank as a daemon

- https://github.com/vy/swank-daemon

Templating

- http://www.common-lisp.net/project/cl-emb/

Testing

There is a unit testing framework build into Emacs

Tools

- Emacs Lisp Lint (check code quality)

```
elint-current-buffer
elint-file
elint-directory
```

Web

Full featured http client

- (ql:quickload “drakma”)
- GET with auth

```
(require 'drakma)
(setq resp (drakma:http-request "http://www.codekid.net"
                                :basic-authoization '("user" "password")))
(print resp)
```

- POST

```
(drakma:http-request "http://some.secure.website/with-login"
                     :method :post
                     :parameters '(("username" "hans")
                                   ("password" "wurst")))
```

- If you want to read the respone from a stream set `:want-stream t`
- More examples on http://weitz.de/drakma/#examples

Parsing HTML5

```
(html5-parser:parse-html5 (drakma:http-request "http://www.google.de"))
```

Web frameworks

- mod_lisp
- http://wookie.beeets.com
- https://github.com/fukamachi/caveman/
- http://weblocks.viridian-project.de

Web server

- http://weitz.de/hunchentoot/

Perl

Debugging

Pdb basics

- s - single step
- n - next instruction
- <enter> - repeat previous command
- c (<line>) - continue (to line)
- p $var - print
- b - set breakpoint
- b <condition e.g. $var eq “foo”> - conditional breakpoint
- L - list all breakpoints

- B <linel*> - delete breakpoint

Run pdb in emacs

- esc+x perldb

Enable debugger at runtime

- cpan Enbugger

```
require Enbugger;
Enbugger->stop;
```

REPL on fatal errors

- cpan Carp::REPL

Useful Perl one liners

Search and replace

- Search in file and replace a with b

```
perl -p -e 's/a/b/g' datei
```

- Replace and make backup before

```
perl -p -i.old -e 's/a/b/g' datei
```

- Recursivly replace files

```
find . -type f | xargs perl -p -e 's/a/b/g'
```

ASCII, Hex and Binary conversion

- ASCII2Hex (urlencoded)

```
perl -e 'map { print "%"; printf("%x",ord($_)); } split(//,$ARGV[0]); print"\n";'
```

- Hex2ASCII

```
perl -e 'map { print chr(hex($_)); } split(/%/,$ARGV[0]); print "\n";'
```

- ASCII2Binary

```
perl -e 'map { print "$_ "; printf("%08b",ord($_)); print "\n"; } split(//,$ARGV[0]);'
```

- Binary2ASCII

```
perl -e 'print chr(ord(pack("B*",$ARGV[0]))."\n";'
```

Base64

- Encoding

```
perl -MMIME::Base64 -e 'print MIME::Base64::encode_base64($ARGV[0]) . "\n"' "BLA BLA␣
↪BLA"
```

- Decoding

```
perl -MMIME::Base64 -e 'print MIME::Base64::decode_base64($ARGV[0]) . "\n"'
↪"QkxBIEJMQSBCTEE="
```

Make /etc/hosts to domain names

```
perl -n -e 'my @a=split(/\s*\s/,$_); print "$a[1]\tIN\tA\t$a[0]\n";' /etc/hosts >>␣
↪domain.name.fwd
```

Perlbrew

List perl versions

```
perlbrew available
```

Install a new interpreter

```
perlbrew install perl-5.17.10
```

Use new interpreter

```
perlbrew list
perlbrew use perl-5.17.10
```

Switch default interpreter

```
perlbrew switch perl-5.17.10
```

Upgrade to latest perl version

```
perlbrew upgrade-perl
```

Create virtualenv

```
perlbrew lib create <name>
perlbrew use @<name>
```

Install modules in virtualenv

```
perlbrew use @<virtualenv>
perlbrew install-cpanm
cpanm <module>
```

Web stuff

Manipulate cookies

```
perl -MLWP::UserAgent -e '$a = LWP::UserAgent->new(useragent => "yoo 1.3"); \
$r = HTTP::Request->new(GET => $ARGV[0]); \
$r->header("Cookie" => "$ARGV[1]=$ARGV[2]"); \
print $a->request($r)->content;' \
http://some.site key value
```

Read HTTP headers

```
perl -MLWP::UserAgent -e '$a = LWP::UserAgent->new(useragent => "fooblah 1.0"); \
$r = $a->get($ARGV[0]); \
map { print "$_:" . $r->header("$_") . "\n"; } $r->header_field_names;' \
http://www.chaostal.de
```

Python

Date tricks

- Which date is in a week?

```
from datetime import datetime, timedelta
in_one_week = datetime.today() + timedelta(days=7)
print in_one_week.strftime("%d.%m.%Y")
```

- Create datetime object from date

```
mydate = datetime(1981, 12, 31, 23, 59)
```

- Get unix time of a date

```
time.mktime(datetime(1982, 12, 31, 16, 32).timetuple())
```

- Get date from unix time

```
datetime.datetime.fromtimestamp(1004260000)
```

- Format date

```
print mydate.strftime("%d.%m.%Y %H:%M")
```

- Combine date and time

```
from datetime import datetime, time
datetime.combine(datetime.today(), time())
```

- Check if date is over

```
end_date = datetime(2011, 12, 24, 0, 0)

if datetime.now() > end_date:
```

Debugging

Pdb

- pdb is the integrated (but quite unhandy) debugger
- To start it at a specific code point

```
import pdb; pdb.set_trace()
```

- `l` to list the current code, `p` to print a variable, `n` for next statement, `s` to step-into and `b` for breakpoint
- use the power of `dir()` to get a list of attributes of a variable / class

Pudb

- Nice ncurses based debugger
- Same keybindings as pdb
- Install

```
pip install pudb
```

- Afterwards edit /usr/lib/python2.6/site-packages/urwid/raw_display.py, go to `signal_init` and `signal_restore`, comment out all code and insert `pass`
- To invoke pudb

```
import pudb; pudb.set_trace()
```

- To start a script in pudb

```
python -m pudb.run yourfile.py
```

Winpdb

- A graphical debugger with remote support and conditional breakpoints
- To invoke it

```
import rpdb2; rpdb2.start_embedded_debugger("password")
```

- Now launch `winpdb` File –> Password, File –> Attach
- To tunnel winpdb through SSH

```
ssh -C -N -f -L 51000:localhost:51000 user@$SERVER_HOST
```

Inject a pudb into a running process

- Install pyrasite
- Create a inject-pudb.py file

```
import pudb;pudb.set_trace()
```

- Find pid of desired python process
- Inject the code

```
pyrasite <pid> inject-pudb.py
```

Connect an ipython shell to a running process

- Install pyrasite
- Create a inject-ipython.py file

```
from IPython.frontend.terminal.embed import InteractiveShellEmbed
ipshell = InteractiveShellEmbed()
ipshell()
```

- Find pid of desired python process
- Inject the code

```
pyrasite <pid> inject-ipython.py
```

Python Diffing and Fuzzy matching

- difflib.SequenceMatcher()

Distutils

Simple example

```
from distutils.core import setup

CLASSIFIERS = [
      'Development Status :: 5 - Production/Stable',
              'Environment :: Web Environment',
              'Framework :: Django',
              'Intended Audience :: Developers',
              'Operating System :: OS Independent',
              'Programming Language :: Python',
              'Topic :: Internet :: WWW/HTTP :: Dynamic Content',
              'Topic :: Software Development',
              'Topic :: Software Development :: Libraries :: Application Frameworks',
```

```
  ]

setup(name="myproject",
    version="1.0",
    description="some description",
    author="me",
    packages=["mypackage"],
    scripts=["myscript"],
    install_requires=["some_module>=version"],
)
```

Complex example

```
import os
import sys
from distutils.command.build_py import build_py as _build_py
from distutils.core import setup
from distutils.dir_util import copy_tree


class build_py(_build_py):
  setup(name="myproject",
    version="1.0",
    description="some description",
    author="me",
    packages=["mypackage"],
)

  copy_tree("dir_of_dirs", os.path.join(sys.prefix, "share", "foo", "bar"), update=1,␣
↪verbose=1)
```

Encoding hell

- encode() liefert hex z.b. xb6 für ö
- decode() liefert char von hex
- UTF8 CSV Dateien richtig einlesen

```
csv.reader(codecs.EncodedFile(file, "rb"), "utf-8")
```

Fabfile

```
from fabric.api import hosts, run

@hosts('host1', 'host2')
def task():
    run('uname -a')
```

File stuff

- Parse dirname from path

```
os.path.dirname(path)
```

- Parse Filename from path

```
os.path.basename(path)
```

- Test if a file is readable

```
os.access(out_file, os.R_OK)
```

Functional programming

Basics

- lambda - Create an anonymous function
- filter(func, list) - Call func on every list item, return a new list of elements where func returned True
- map(func, list) - Call func on every list item, return new list a returned results
- reduce(func, list) - Call func with first to items of list, than call the result with the third, that result with the forth and so on

```
reduce(lambda x,y: x+y, [1,2,3])
-> (1 + 2) + 3
```

- zip(list1, list2, listN) - creates list of tuples containing the n-th item of the input lists
- A complete example

```
even = lambda x: x % 2
square = lambda x: x * x
map(square, filter(even, range(1,100)))
[1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441, 529, 625, 729, 841, 961, 1089, 1225,␣
↪1369, 1521, 1681, 1849, 2025, 2209, 2401, 2601, 2809, 3025, 3249, 3481, 3721, 3969,␣
↪4225, 4489, 4761, 5041, 5329, 5625, 5929, 6241, 6561, 6889, 7225, 7569, 7921, 8281,␣
↪8649, 9025, 9409, 9801]
```

- The same with list comprehensions

```
[x*x for x in range(1,100) if x % 2]
```

- find items contained in both lists

```
[x for x in list1 if x in list2]
```

Generators

- Generator remember it's state and return (yield) the next item

```
def gen_even_numbers():
    i = 2

    while True:
        yield i
```

```
        i += 2

gen = gen_even_numbers()
gen.next()
```

- Generator expressions are like list comprehensions with round parentheses

```
gen = (x for x in filter(lambda x: x % 2 == 0, range(100)))
```

Coroutine

- A coroutine is like a generator but excepts chunk-wise input
- It executes till the next *yield* and wait for you to *send* data in
- Dont forget the pretending *next* otherwise the coroutine will never reach the first *yield*

```
def receiver():
  while True:
    print "Waiting for data..."
    data = (yield)
    print "Got " + str(data)

recv = receiver().next()
recv.send("abc")
recv.close()
```

Functools

- create new function with fixed parameter

```
def sum(a,b):
    return a + b

import functools
add_two = functools.partial(sum, b=2)
```

Closure

- A closure is a function pointer with saved parameters

```
from urllib import urlopen

def page(url):
  def get():
    return urlopen(url).read()
  return get

codekid = page("http://www.codekid.net")
codekid()
```

Decorators

- A Decorator is a function that wraps another function
- code by hand

```
def hello(func):
  def callf(*args, **kwargs):
    print "Hello"
    func(*args, **kwargs)
    print "Bye"
  return callf
```

- with functools

```
from functools import wraps
def my_decorator(func):
    @wraps(func)
    def wrapper(*args, **kwds):
        print 'Calling decorated function'
        return func(*args, **kwds)
    return wrapper
```

- http://rxwen.blogspot.com/2010/12/python-decorators.html

Conditional decorator with arguments

```
from nose.tools import timed
import time

def not_on_travis(decorator):
  def wrapper(func):
    if os.getenv("TRAVIS"):
      return func
    else:
      return decorator(func)
  return wrapper

@not_on_travis(timed(0.1))
def say(what):
  print "You said " + what
  time.sleep(1)
```

Memoize Decorator

- Caches function results for inputs

```
def memoize(f):
    cache = {}

    @wraps(f)
    def helper(x):
        if x not in cache:
            cache[x] = f(x)
        return cache[x]
    return helper
```

Introspection

- type() returns the class of an object
- hasattr() / getattr() / setattr to query / get / set a property
- issubclass()

```
for (n, v) in myobj.__dict__.items(): print "%s: %s" % (n, v)
```

Import-Hook

```
import __builtin__
old_import = __builtin__.__import__
known_imports = []

def import_hook(name, globals=None, locals=None, fromlist=None):
    if name.startswith("django_chuck") and name not in known_imports:
        if fromlist:
            for symbol in fromlist:
                print name + "." + symbol
                known_imports.append(name + "." + symbol)
        else:
            print name
            known_imports.append(name)

    return old_import(name, globals, locals, fromlist)

__builtin__.__import__ = import_hook
```

Inrospect imports with compiler

```
# code borrowed from zope
class ImportFinder:
    def __init__(self):
        self.imports = []

    def visitFrom(self, statement):
        stmt = statement.asList()
        if stmt[0] == '__future__':
            # we don't care what's imported from the future
            return
        names_dict = {}

        for orig_name, as_name in stmt[1]:
            # we don't care about from import *
            if orig_name == '*':
                continue

            self.imports.append(stmt[0] + "." + orig_name)

    def visitImport(self, stmt):
        for orig_name, as_name in stmt.names:
            self.imports.append(orig_name)
```

```
ast = compiler.parseFile("django_chuck/commands/create_project.py")
log = ImportFinder()
compiler.walk(ast, log)
print log.imports
```

Ipython

Load a file into external editor

```
edit file
```

- Use Emacs

```
EDITOR=emacsclient ipython
edit file
```

Run a file

```
run file
```

Save code from shell

```
save filename.py 1-10
```

Display available namespaces

```
who
```

- Including data types

```
whos
```

Get documentation

- Use ? after module, function, whatever
- pinfo some module or function or whatever

Show source code of a function

```
psource <function>
```

- Get whole source file

```
pfile <function>
```

Edit a function

- -x will not execute the new code

```
edit <function>
```

Save shell command in variable

```
foo = !ps ax | grep something
```

Define aliases

```
%alias pids ps ax | grep app | cut -d " " -f 1
%store pids
```

Debugging

- Switch on pdb on execptions

```
pdb
```

- Run script in pdb

```
run -d file
```

- Run script in pdb with breakpoint in line 23

```
run -d -b 23 file
```

- Automatically start Ipython debugger on exception

```
try:
  ret = 1 / 0
except Exception, e:
  import sys, IPython
  IPython.Shell.IPShell(argv=[])
  IPython.Debugger.Pdb(IPython.ipapi.get().options.colors).set_trace(sys._getframe())
```

Profile

- http://pynash.org/2013/03/06/timing-and-profiling.html

```
%time some_function
```

- Run cProfile

```
%prun file or function
```

- Filter output

```
%prun -l some_filter_string file or function
```

History

- Show history

```
hist
```

- Execute command nr x

```
_ix
```

- Ranges (1-5)

```
In[1:6]
```

- Print output of command 42

```
Out[42]
```

Bookmarks

- Create a dir bookmark

```
bookmark name
```

- Save bookmark

```
store name
```

- List bookmarks

```
bookmark -l
```

- Delete bookmark

```
bookmark -d name
```

Macros

- Save history commands 1-5 in a macro muh

```
macro muh 1-5
```

- Save the macro

```
store muh
```

- Execute it by calling its name
- Show the source

```
print muh
```

Background jobs

- Start a statement in the background

```
bg some_func()
```

- Show status of job

```
job[0].status
```

- Get the result

```
job[0].result
```

Connect to an existing console

- Start ipython with console or qtconsole

```
ipython kernel
[IPKernelApp] To connect another client to this kernel, use:
[IPKernelApp] --existing kernel-26492.json
```

- Now you can connect another console by executing

```
ipython console --existing kernel-26492.json
```

- If the kernel is on another system you either have to setup ssh tunnel for all port specified in the json file

```
ssh user@remote -f -N -L 61947:127.0.0.1:61947
```

- or append *–ssh user@remote* to the ipython console command
- The kernel.json file can be found in `~/.ipython/profile_default/security`
- To manually create all ssh tunnels

```
for port in $(cat kernel-1234.json | grep '_port' | grep -o '[0-9]\+'); do ssh␣
↪myremotehost -f -N -L $port:127.0.0.1:$port; done
```

Parallel computing

- Its possible to do parallel computing in different ways like SSH, MPI, PBS and even Windows HPC
- The controller opens a socket and allow engines to connect to it
- First of all lets start an Ipython console locally for every core

```
ipcluster start
ipython
```

- To run a simple hello world on all cores exec

```
from IPython.parallel import Client
c = Client()
c.ids
c[:].apply_sync(lambda : "Hello, World")
```

- Manually start a controller on your head node and start engines on all computing nodes (must have access to the file ~/.ipython/profile_default/security/ipcontroller-engine.json generated by the controller)

```
ipcontroller --enginessh=user@host
ipengine
```

- Now you can import the module `os` on every engine and print uname information

```
with c[:].sync_imports():
  import os
%px print os.uname()
```

- If all nodes share a home directory for ipcontroller-engine.json file you can automatically start the controller on localhost and engines on remote
- Create a new profile with

```
ipython profile create --parallel --profile=ssh
```

- Edit `ipcluster_config.py` and add

```
c.IPClusterEngines.engine_launcher_class = 'SSHEngineSetLauncher'
c.SSHEngineSetLauncher.engines = { 'host1.example.com' : 2,
          'host2.example.com' : 5,
          'host3.example.com' : (1, ['--profile-dir=/home/different/location']),
          'host4.example.com' : 8 }
```

- The key of the engines dict is the host and value number of engines to start
- Now you can setup the cluster by executing

```
ipcluster start --profile=ssh
```

- Use map to call a function on a dataset on all nodes

```
v = c[:]
result = v.map_sync(lambda x: x*x, data)
```

Save state of a console

- You can start / stop logging of a sessions state with `%logstart <file>` / `%logstop`
- To load the session state exec

```
ipython -lp <file>
```

- To automatically save state edit `~/.ipython/profile_default/ipython_config.py` and set

```
c.TerminalInteractiveShell.logstart = True
c.TerminalInteractiveShell.logfile = '~/ipython_session.log'
```

Extensions

- To install an extension execute

```
%install_ext https://www...
```

- https://pypi.python.org/pypi/ipython-sql
- Cython
- http://blog.cloudera.com/blog/2014/08/how-to-use-ipython-notebook-with-apache-spark/

JSON

- Validate JSON

```
echo '{ 1.2:3.4}' | python -mjson.tool
```

Looping

- Loop a list with index

```
for i, entry in enumerate(a_list):
    print i, entry
```

- Check empty list in for loop

```
for egg in spam:
  print egg
else:
  print "There is no spam"
```

Matplotlib

A simple line graph

```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[100,25,70,80,100]
plt.plot(a,b)
plt.show()
```

A bar graph

```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[100,25,70,80,100]
fig = plt.figure()
ax = fig.add_subplot(111)
ax.bar(a,b)
plt.savefig("graph.png")
```

Two bar diagrams beside

- bar(left, height, width)
- Can be single values or arrays

```
import numpy as np
import matplotlib.pyplot as plt

a=[1,2,3,4,5]
b=[100,25,70,80,100]
c=[90,35,80,80,90]

width=0.25
fig = plt.figure()
ax = fig.add_subplot(111)

plt.bar(np.array(a), b, width, color="r")
plt.bar(np.array(a) + width, c, width)
plt.show()
```

Metaprogramming

Metaclasses

- Metaclasses demystified

Get all attributes of a datatype

code-block:: python

    dir(var)

Dynamically create a new class

```
classobj = type("MyClass", (object,), {})
```

- Show all attributes and their values of an Python object

Respond to every method call

..code-block:: python

    import functools

    class Bla:

        def __getattr__(self, name):

            def handleattr(attrname, args): print “Hi I am attribute ”, attrname, ” with arguments ”, args

            return functools.partial(handleattr, name)

Misc

- Inside Python Objects

Examing the byte-code

```
python -m gis <python_file>
```

Multiprocessing

- For real concurrency on mutli-core cpus use the package multiprocessing due to the global-interpreter-lock in the python interpreter
- Celery is an asynchronous task queue/job queue

Parallel map

```
import urllib2
import multiprocessing

def fetch(url):
  print "GET " + url
  body = urllib2.urlopen(url).read()
  print "FIN " + url; return (url, body)

urls = ['http://www.google.de', 'http://www.ccc.de', 'http://www.heise.de', 'http://
↪www.codekid.net']
pool = multiprocessing.Pool()
result = pool.map_async(fetch, urls):
timeout = 3

try:
  for url, body in result.get(timeout):
    print "GOT body for " + url
except multiprocessing.TimeoutError:
  print "Got timeout :("
  pool.terminate()
```

Gevent

```
import gevent
import urllib2

def fetch(url):
  print "GET " + url
  body = urllib2.urlopen(url).read()
  print "FIN " + url
  return (url, body)

urls = ['http://www.google.de', 'http://www.ccc.de', 'http://www.heise.de', 'http://
↪www.codekid.net']
jobs = [gevent.spawn(fetch, url) for url in urls]
```

```
gevent.joinall(jobs, timeout=2)
[job.get() for job in jobs]
```

Multiprocessing with Queues

```
from multiprocessing import Process, Queue
import commands

nr_of_threads = 4

def do_work(work_queue, result_queue):
    while work_queue.qsize():
      job = work_queue.get()
      result_queue.put(["what", "ever"])

def parallel_work(jobs, nr_of_threads):
  work_queue = Queue()
  result_queue = Queue()
  result = {}

  for job in jobs:
      work_queue.put(job)

  if nr_of_threads > len(jobs):
      nr_of_threads = len(jobs)

  for i in range(nr_of_threads):
      worker = Process(target=do_work, args=(work_queue,result_queue))
      worker.start()

  while len(result.keys()) < len(jobs):
      data = result_queue.get()
      print data
      result[data[0]] = data[1]

  return result
```

Fork Decorator

```
def forked(func):
  def wrapped(*args, **kwargs):
    import os

    pid = os.fork()
    if pid > 0: func(*args, **kwargs)

  return wrapped
```

Thread Decorator

```
def threaded(name):
  def callf(func):
    def wrapped(*args, **kwargs):
      import thread

      def newfunc():
        func(*args, **kwargs)
      thread.start_new_thread(newfunc, ())
    return wrapped
  return callf
```

MapReduce

- Disco MapReduce Framework with Python API
- Local example for multi-core cpu

```
import sys
from multiprocessing import Pool

def split_words(line):
    return [x.rstrip("\n") for x in line.split(" ")]


def myreduce(mylist):
    """
    gets [['word1'], ['word1', 'word2', 'word1']]
    returns {'word1': 3 'word2': 1}
    """
    result = {}

    for sublist in mylist:
        for word in sublist:
            try:
                result[word] += 1
            except KeyError:
                result[word] = 1

    return result


if len(sys.argv) < 2:
    print sys.argv[0] + ": <file>"
    sys.exit(1)

pool = Pool(processes=10)
lines = file(sys.argv[1]).xreadlines()

words = pool.map(split_words, lines)
word_count = myreduce(words)

for (word, count) in word_count.items():
    print word + ": " + str(count)
```

Pandas

Tutorials

- http://www.gregreda.com/2013/10/26/intro-to-pandas-data-structures/
- http://pyvideo.org/video/2330/diving-into-open-data-with-ipython-notebook-pan

Parse Excel or CSV into DataFrame

```
import pandas as pd
# data = pd.read_excel('example.xlsx', 'sheet1')
data = pd.read_csv("example.csv", sep=';', header=None, names=['host', 'speed', 'time
↪'])
print data[data.speed <= 950]
```

Get data from SQL

```
from pandas.io import sql
import sqlite3

conn = sqlite3.connect('example.sqlite3')
query = "SELECT * FROM mytable WHERE mycol = 'MUH';"

data = sql.read_frame(query, con=conn)
print data.describe()
```

Read data from the web

```
from urllib2 import urlopen
from StringIO import StringIO

url = urlopen('http://www.codekid.net/data_file.txt').read()
data = pd.read_table(StringIO(url), sep='\t')
```

Converter

- replace all $ in salary column with nothing

```
data = pd.read_csv('example.csv',
                   converters={'salary': lambda x: float(x.replace('$', ''))})
```

Selecting stuff

- where

```
print data[(data.speed <= 950) | (data.time > 0.2)]
print data[(data.speed > 950) & (data.time < 0.2)]
```

- join

```
pd.merge(left_frame, right_frame, on='key', how='inner')
pd.merge(left_frame, right_frame, on='key', how='left')
pd.merge(left_frame, right_frame, on='key', how='right')
```

Merging

```
data_full = pd.concat([data, data_set2])
data_full = pd.merge(data, data_set2)
```

Aggregation

- Group data by column title
- Run np.size and np.mean functions on column rating

```
data.groupby('title').agg({'rating': [np.size, np.mean]})
```

Iteration

```
for entry in data[['date', 'value']].itertuples():
  print entry
```

Graphing

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_excel('example.xlsx', 'sheet1')
data.set_index("name", inplace=True)
plt.figure()
plt.title("Just a simple test")
data.plot(kind="barh")
plt.show()
```

Performace tricks

- use generators instead of lists were possible
- run interpreter with `-O`
- For big ranges use `xrange()` cause it creates a generator
- Use `xreadlines()` for big files cause it also creates a generator
- Setting environment variable `PYTHONUNBUFFERED` will force stdout / stderr to be unbuffered
- `os.fdopen("file", "r", 0)` to open file without buffering
- Use `flush()` on filehandle to flush I/O buffer
- Use a list and `join` instead of string concatenation

- Prefer `while 1` instead of ‘‘ while True‘‘
- Write “x < y < z” instead of “x < y and y < z”
- `"test " + var` is slower than `"test %s" % (var,)`
- `map()`, generator and list comprehension are faster than a loop
- put your global variables into a `main` function and call it by:

```
if __name__ == '__main__':
  main()
```

- profile your code

```
import profile
profile.run("any_python_expression")
```

Pip

- Install module from git repository

```
pip install --upgrade -e "git://server/repo.git#egg=some_name"
```

- Save all installed modules into file

```
pip freeze > requirements.txt
```

- Install all modules from file

```
pip install -r requirements.txt
```

- Config

```
[global]
timeout = 60

[freeze]
timeout = 10

[install]
download-cache=~/.pip/cache/
use-mirrors = true
```

Pygraphviz

Simple example

```
import pygraphviz as pgv

G=pgv.AGraph(strict=False,directed=True, rankdir = "LR")

test1 = "Test 1"
test2 = "Test 2"
test3 = "Test 3"
test4 = "Test 4"
```

```
G.add_node(test1)
G.add_node(test2)
G.add_node(test3)
G.add_node(test4)

G.add_edge(test1, test2)
G.add_edge(test2, test3)
G.add_edge(test2, test4)

e = G.get_edge(test2, test3)
e.attr["label"] = "no"

e = G.get_edge(test2, test4)
e.attr["label"] = "yes"

G.node_attr['shape']='box'
G.node_attr['style']='filled'
G.node_attr['fillcolor']='grey'

G.layout("dot")
G.draw('graph.png')
```

DOT code example

- DOT code

```
digraph testdriven {

test [label="Write test", shape="box"] // some comment
run [label="Run test", shape="oval"]
code [label="Write code", shape="box"]
green [label="Green", shape="oval", style="filled", color="green"]

test->run
run->code
code->green
green->test [label="Next iteration", arrowhead="normal", style="solid"]
}
```

- Python code to generate image

```
import tempfile
import pygraphviz as pgv

tmpfile = tempfile.NamedTemporaryFile(delete=False)
tmpfile.write(dot_code)
tmpfile.flush()

graph = pgv.AGraph(tmpfile.name)
graph.layout(prog="fdp")
graph.draw(out_file)
tmpfile.close()
```

How to test DOT code

```
dot -Tpng test.dot > output.png
```

Pylint

- Ignore one error on one line

```
# pylint: disable-msg=E1103
```

Python Shell

- Attach python console to a running process with [http://code.google.com/p/rfoo/|rconsole]
- Stop a running process by signal and let it drop you to a python console

```
import code
import signal
signal.signal(signal.SIGUSR2, lambda sig, frame: code.interact())

user@shell> kill -SIGUSR2 <PID>
```

SOAP

- http://code.google.com/p/soapbox/
- https://fedorahosted.org/suds/
- https://github.com/fvieira/wsdl2soaplib

String Conversion

- Convert char to int

```
ord("A")
```

- Convert char to hex

```
hex(ord("A"))
```

- Convert int to char

```
chr(42)
```

- Convert string to hex

```
import binascii
binascii.hexlify("test")
```

- Convert hex to string

```
binascii.unhexlify("74657374")
```

- String to unicode html

```
"".join(["&#" + str(hex(ord(x))) + ";" for x in "script"])
python2 -c 'import sys; print "".join(["&#" + str(hex(ord(x))) + ";" for x in sys.
↪argv[1]])' script
```

- String to url encoding

```
python2 -c 'import sys; print "".join(["%" + str(hex(ord(x))).replace("0x","") for x␣
↪in sys.argv[1]])' "javascipt:alert('BUH');"
```

System Administration

- How to use find in python

```
import commands
print commands.getoutput("find /tmp")
```

- shutil has some useful shell commands as functions
- Run a command on some hosts via SSH (fab task)

```
from fabric.api import hosts, run

@hosts('host1', 'host2')
def task():
    run('uname -a')
```

- Get all files in a directory recursively

```
def get_files(dir):
    files = []

    for (path, subdirs, new_files) in os.walk(dir):
        for new_file in new_files:
            files.append(os.path.join(path, new_file))

    return files
```

Misc

- http://jessenoller.com/blog/2009/02/05/ssh-programming-with-paramiko-completely-different
- http://plumbum.readthedocs.org

Templating

```
from string import Template
s = Template('$who likes $what')
s.substitute(who='tim', what='kung pao')
```

- http://www.makotemplates.org/

Testing

Testing with unittest

```
class MyTests(unittest.TestCase):
  def setUp(self):
      self.some_value = 23

  def test_addition(self):
      self.assertEqual(2 + self.some_value, 25)

 def test_exception(self):
   with self.assertRaises(ValueError):
       raise ValueError()
```

Testing with nose

- Nose can use unittest tests but you want do more with nose tests

```
my Tests():
  def setup(self):
    self.some_value = 23

  # assumes function runs without failure
  def test_addition(self):
    assert self.some_value + 2 == 25

  def test_addition_with_nose(self):
    import nose
    nose.tools.assert_equals(self.some_value + 2, 25)

  @raises(ValueError)
  def test_exception(self):
    raise ValueError()
```

- use nosy to automatically run all tests if code changes

```
pip install nosy
nosy
```

- check code coverage of your unittests

```
pip install coverage
nosetests --with-coverage
```

- testing performance

```
# test fails if it runs longer than 0.2 seconds
@timed(0.2)
def test_something():
  pass
```

- run profiler on every tested unit

```
nosetests --with-profile
```

- get a pdb on test failures

```
nosetests --pdb-failures
```

- get output in xUnit format (usefull for e.g. Jenkins)

```
nosetests --with-xunit --xunit-file=testresults.xml
```

- Integrate nose into setuptools

```
setup (
  # ...
  test_suite = 'nose.collector'
)
```

Doctests

- Use nose to run Doctests

```
def some_function(a, b):
    """
    >>> some_function(2, 3)
    5
    """
    return a + b
```

```
nosetests --with-doctest
```

Mocking

- Using nose and mock

```
@mock.patch('some_module.nasty_method', lambda x: True)
def test_nasty_method():
  assert some_module.nasty_method()
```

- Using mock without nose
- Using flexmock

Web stuff

Share dir with simple webserver

```
python -m SimpleHTTPServer 8080
```

Grep in url

```
python -c "import sys,urllib2; print filter(lambda line: sys.argv[2] in line, urllib2.
↪urlopen(sys.argv[1]).readlines())" http://www.codekid.net "Network Hacks"
```

Web Crawling

- Scapy

Hello world in WSGI

```
from wsgiref.simple_server import make_server

def hello_world(environ, start_response):
  start_response('200 OK', [('Content-Type', 'text/plain')])
  return ['Hello World!']

make_server('localhost', 8000, hello_world).serve_forever()
```

RabbitMQ

Overview

- Producer sends messages to an exchange
- An exchange decides what to do with a message and maybe forwords it to one or more queues
- A binding binds an exchange to a queue
- A queue store the message until the consumer receives (and acknowledges) it
- Get status

```
rabbitmqctl status
rabbitmqctl list_connections
rabbitmqctl list_exchanges
rabbitmqctl list_queues
rabbitmqctl list_users
```

User management

- Add a new user

```
rabbitmqctl add_user <username> <password>
```

- Change password

```
rabbitmqctl change_password <username> <newpassword>
```

- Delete user

```
rabbitmqctl delete_user <username>
```

- Testing access

```
import pika
parameters = pika.ConnectionParameters('localhost', 5672, '/', pika.PlainCredentials(
↪'guest', 'guest'))
```

```
connection = pika.BlockingConnection(parameters)
connection.disconnect()
```

Enable management plugin

```
/usr/lib/rabbitmq/bin/rabbitmq-plugins enable rabbitmq_management
```

- Restart rabbitmq-server
- Point your browser to *http://localhost:55672*

Troubleshooting

- RabbitMQ server needs *disk_free_limit* space (default 1 Gb) otherwise it wont accept messages

Python example

- pip install pika
- Sending

```
import pika

connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
channel = connection.channel()
channel.queue_declare(queue='hello', duarable=True)
channel.basic_publish(exchange='',
                      routing_key='hello',
                      body='Hello World!',
                      properties=pika.BasicProperties(delivery_mode=2))
print " [x] Sent 'Hello World!'"
```

- *duarable=True* save queue before restarting / stopping server
- *delivery_mode=2* save messages of this queue
- Receiving

```
import pika

connection = pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()
channel.queue_declare(queue='hello')

print ' [*] Waiting for messages. To exit press CTRL+C'

def callback(ch, method, properties, body):
  print " [x] Received %r" % (body,)

  channel.basic_qos(prefetch_count=1)
  channel.basic_consume(callback,
                        queue='hello',
                        no_ack=True)

  channel.start_consuming()
```

- Set *no_ack* to *False* to send acks after task was processes otherwise messages could get lost if worker dies

Shell Scripting

Sequence

```
for i in {1..32}; do echo $i; done
```

Calculation

```
x=3
y=$(($x+4))
```

Quote Spaces in Filenames

```
echo $FILE | sed 's/ /\\ /g'
```

Remove lines beginning with XXX

```
sed -i -e '/^XXX_/d' /my/file"
```

Get yesterdays date

```
date +%Y:%m:%d -d "1 day ago"
```

Now in unix time

```
date +%s -d "now"
```

Find files modified after date

```
find . -type f -newermt 2012-12-24
```

Get lines where the nth element is bigger than x

```
perl -n -e '@a=split(/\s/,$_); print $_ if $a[3] > 2;'
```

Generate a random number

```
echo $RANDOM
```

CHAPTER 26

Misc

Netlabels

Drum ‘n Bass

- http://www.plainaudio.com/
- http://zardonicrecs.netii.net/
- http://psymbionic.bandcamp.com/
- http://www.pantyraidmusic.com/music/
- http://bredemusic.com/

Triphop

- http://www.sofasound.de/
- http://www.pulsar-records.de
- http://www.fresh-poulp.net

House

- http://www.supafeed.net/

Electro

- http://www.netlabelism.net
- http://www.archive.org/details/enough_records

- http://www.plexrecords.com

Techno

- http://www.archive.org/details/phasetech
- http://www.archive.org/details/cicuta-netlabel

Mixes

- http://kor.gazaxian.com/
- http://www.cybermusique.de/
- http://pulsar.cc/?mixes
- http://www.mixotic.net/
- http://www.sonicwalker.com/
- http://remixta.net/

Suchtechniken

Allgemeine Suchtipps

- Formuliere Deine Anfrage so exakt wie möglich, verwende hierfür u.a. Such Operatoren
- Überlege was Du als Ergebnis haben willst
- Wenn Du was spezielles suchst nutze spezialisierte Suchmaschinen und Webkataloge
- Filter, Filter, Filter z.B. wenn Du nichts kommerzielles suchst -“site:com”
- Es gibt viele Möglichkeiten zu suchen, nutze sie
- Speichere erfolgreiche Search Strings (z.B. in einer Textdatei oder einem Wiki)
- Ggf lohnt es sich nur in einer bestimmten Regio zu suchen z.B. wegen der Gesetzeslage oder Kultur

Profi Suchtipps

- Nutze die Möglichkeit mancher Suchmaschinen automatisch neue Ergebnisse von Suchanfragen zu erhalten
- alerts.google.com
- pipes.yahoo.com
- Programmiere deinen eigenen Suchagenten mit Hilfe der offenen APIs von Google, Yahoo etc

Google Suchtechniken - Basics

- Boolsche Operatoren: and / or (|)
- and ist Standard
- Klammerung von Suchbegriffen z.B. (ccc and freiburg) or cccfr

- Begriff zusammenfassen mit "" z.B. "chaos computer club"
- Mit - kann man Wörter ausschliessen
- Ein Punkt steht für einen beliebigen Buchstaben
- Der Stern steht für ein Wort z.B. chaos * club

Google Suchtechniken - Advanced

- Suche in bestimmten Bereichen: intext, intitle, inurl
- filetype
- site - sucht nur auf einer Seite / Domain / TLD z.B. site:.de
- link - sucht nach Links z.b. link:www.cccfr.de
- cache
- numrange:1-10 (das selbe wie 1..10)
- daterange (sucht nach Indizierungsdatum in Julian Date Format, d.h. Anzahl der Tage seit 1. Januar 4713 B.C.)
- http://www.onlineconversion.com/julian_date.htm
- kann man jetzt auch in den erweiterten Optionen links in normalem Datumsformat angeben
- http://www.googleguide.com/advanced_operators_reference.html

Google Hacking Database

- http://www.hackersforcharity.org/ghdb/

Webkatalog

- Dmoz
- web.de

Volltextsuche

- Bing
- Yahoo (sucht mittlerweile über Bing)
- altavista (sucht über yahoo)
- lycos
- fireball (sucht über lycos)
- web.de
- ask.com
- T-Online (sucht über Google)

Meta-Searchengines

- Metager
- Metacrawler
- Yooci.com

Begriffe

- www.wikipedia.org

Nachrichten

- wikinews.org
- paperball.de
- news.google.com

Blogs

- technorati.com
- blogsearch.google.com
- bloglines.com
- blogsearchengine.com

RSS Feeds

- www.rss-verzeichnis.de
- www.rss-nachrichten.de

Podcasts

- podcast.de
- podster.de

Open Source Software

- freshmeat.net
- sourceforge.net

Linux Howtos und Tutorials

- howtoforge.net
- tldp.org

Orte, Geschäfte, Restaurants

- maps.google.com
- de.local.yahoo.com
- map24.de
- gelbeseiten.de

Rezepte

- suche-rezepte.de
- rezeptewiki.org
- www.chefkoch.de

Telefonnummern

- www.telefonbuch.de
- www.das-oertliche.de

Personen

- Suche auf Sozialen Netzen wie Facebook und StudiVZ
- yasni.com
- 123people.com
- email-verzeichnis.de

Produkte und Schnäppchen

- ebay.de
- geizkragen.de
- yoodo.eu
- www.guenstiger.de

Bilder und Fotos

- flickr.com
- images.google.com
- picsearch.de
- www.piqs.de (alle Fotos unter CC-BY)
- www.deviantart.com

Bücher

- books.google.com
- theeuropeanlibrary.org
- http://digital.library.upenn.edu/books/search.html

Bittorrent

- torrent.to
- bitreactor.to

IRC / XDCC

- irc.netsplit.de
- searchirc.com
- xdccing.com
- ircdig.com

Usenet

- groups.google.com
- groups.yahoo.com
- binsearch.info

Dateien

- filestube.com (sucht über Rapidshare u.a.)
- filesearch.ru

Videos

- youtube.com
- clipfish.de
- vimeo.com

Freie Musik

- jamendo.com
- starfrosh.ch
- archive.org

Dj Mixes

- mixdepot.net
- dj-mixes.com

Radiosender

- shoutcast.com
- freie-radios.net

Börseninformationen

- finance.google.com
- finance.yahoo.com

Bildung und Forschung

- bildungsserver.de
- forschungsportal.net (Forschungsergebnisse)

Exploits und Sicherheitslücken

- securityfocus.com
- packetstormsecurity.net
- osvdb.org

Source Code

- code.google.com
- koders.com
- krugle.com

Computer

- shodanhq.com (sucht nach Computern mit bestimmten offenen Ports / Software oder von einem bestimmten Typ wie z.B. Router oder nach SNMP Informationen)

Deutsche Gesetze

- juris.de

Statistiken

- destatis.de (Statistisches Bundesamt Deutschland)
- statista.com

Sterne / Galaxien und Planeten

- nasaimages.org
- www.google.com/sky
- spacetelescope.org

Andere Suchtechniken

- Archie (Suche über FTP Server)
- P2P Netzwerke wie eMule
- YaCy
- Maltego - Datamining Tool

Weiterführende Links

- fravia.com
- suchfibel.de
- suchlexikon.de
- googlelabs.com

CHAPTER 27

Indices and tables

- genindex
- modindex
- search