sdocs

Release 0.1.alpha

Jeffrey Scherling

September 18, 2016

Contents

1 contents 3

2 resources 27

sdocs is an akronym for small/simple dokumentation

CHAPTER 1

contents

1.1 overview

Warning:
A collection of Howto's, Guides, Snippets collected from the web
Only for private use and there ist no warranty for correct information
You use it at your own risk, and all information is copyrighted by the owner
Most of this Source is written and collected by Jeffrey Scherling [a]

[a] Have Fun!

Note: This document was generated on 2016-09-18 at 20:23.

1.2 sdocs

this is the index of sdocs

1.2.1 Slackware

Howto Setup Slackware

I Prerequisites

- current source tree

II Installation for Slackware

- use slackpkg

- use sbopkg
- use SlackBuild scripts

Basic Steps

- use the slackware installer

III Configuration

Basic Steps

- use the slackware installer

Setup Slackware [1]

- http://www.slackware.com

footnotes

1.2.2 Sphinx

a quick start guide

I Prerequisites

1. python
2. sphinx
3. webbrowser or pdfviewer

II Build the documentation

1. enter sphinx-quickstart # create the root directory of documentation
2. edit conf.py # set the output to your needs
3. create your docu name.rst
4. add name.rst to index.rst
5. make html, latexpdf or linkcheck

III Look at the Documentation

1. open index.html with your webbrowser
2. open projectname.pdf with your pdfviewer

[1] see too: http://alien.slackbook.org/blog/

IV Markup

- markup http://sphinx-doc.org/rest.html
- links http://sphinx-doc.org/markup/inline.html#ref-role
- markup code http://sphinx-doc.org/markup/code.html
- guide http://docs.python-guide.org/en/latest/writing/documentation/

V Themes

- nice themes http://docs.writethedocs.org/tools/sphinx-themes/

1.2.3 Webserver

webserver and their configuration

Nginx

- **nginx configuration**
  - config files
  - websites configs

this is the main nginx configuration file *nginx.conf*

```
user        amorsql amorsql; # user group of processes
worker_processes  2;

events {
        worker_connections  1024;
}

http {
        include                         mime.types;
        default_type                  application/octet-stream;
        gzip                          on;
        gzip_min_length                  5000;
        gzip_buffers                  4 8k;
        gzip_types                  text/plain text/css application/x-javascript text/xml application/x
        gzip_proxied                  any;
        gzip_comp_level                  2;
        ignore_invalid_headers          on;
        include                         sites-enabled/*;

        # test it
        sendfile                on;
}
```

another nice things + squid (with caching) + zope (python server)

1.2.4 Ftpserver

Ftpserver and their configuration

Vsftpd

configuration of vsftpd

http://www.basicconfig.com/linuxnetwork/ftp_server#check-vsftpd http://wiki.ubuntuusers.de/vsftpd

```
# Example config file /etc/vsftpd.conf
#
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
# default
local_umask=022
#
#
#
# Uncomment this to allow the anonymous FTP user to upload files. This only has an effect if the abov
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by a different user. Note! Us
# recommended!
#chown_uploads=YES
#chown_username=whoever
```

```
#
# You may override where the log file goes if you like. The default is shown
# below.
xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format. Note that the default log
xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the ftp server can use as a to
nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not recommended for security
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore the request. Turn on the
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner="_____________________________________Welcome to my ftp Site!_____"
#
# customize your login
banner_file=/etc/vsftpd.banner_file
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
# dangerous don't use it
chroot_local_user=NO
chroot_list_enable=YES
passwd_chroot_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
```

```
# the presence of the "-R" option, so there is a strong case for enabling it.
ls_recurse_enable=YES
#
# When "listen" directive is enabled, vsftpd runs in standalone mode (rather
# than from inetd) and listens on IPv4 sockets. To use vsftpd in standalone
# mode rather than with inetd, change the line below to 'listen=YES'
# This directive cannot be used in conjunction with the listen_ipv6 directive.
listen=NO
#
# This directive enables listening on IPv6 sockets. To listen on IPv4 and IPv6
# sockets, you must run two copies of vsftpd with two configuration files.
# Make sure, that one of the listen options is commented !!
#listen_ipv6=YES
#
# adds by jeff
#
# allow write with chroot
allow_writeable_chroot=YES
#
# access to only this users
userlist_deny=NO
userlist_enable=YES
userlist_file=/etc/vsftpd.user_list
#
# ssl
#ssl_enable=NO
#ssl_sslv2=YES
#create ssl - Zertifikat for ssl using
#openssl req -x509 -nodes -days 365 -newkey rsa:1024  -keyout /etc/ssl/private/vsftpd.pem -out /etc/s
#
# guests remapping all non annonymus to this login
#guest_enable=YES
#guest_username=ftpuser
#
max_clients=3
max_per_ip=2
#
tilde_user_enable=YES
#
# hide and deny files and directories
hide_file={/,/media,/gamma_sftp}
deny_file={/,/media,/gamma_sftp}
#
# ftp commands to deny
# deny change to the parent of the current working directory.
#cmds_denied=XCUP
#
# set the default mmask
#file_open_mode=0777
#
#
#
#
```

restart the process

root@gamma:~# /etc/rc.d/rc.inetd restart Starting Internet super-server daemon: /usr/sbin/inetd

customize the login with vsftpd.banner_file

```
      _____________________________________________Welcome to my ftp Site!_____
              ooo                  ooo
              oo                   oo
     ooooo    oo    oooo     oooo    oo  ooo ooo         ooo oooo    ooo ooo    oooo
    oo       oo       oo  oo       oo oo     oo    oo    oo      oo    ooo  oo oo  oo
    ooo      oo    ooooo  oo       oooo       oo oooo oo    ooooo    oo       oooooo
     oo    oo  oo  oo  oo       oo oo        ooo  ooo    oo  oo    oo       oo
ooooo  oooo  oooooo  oooo  ooo  ooo     oo    oo     oooooo  oo        ooooo
_____________________________________________________________________User 1_______
```

allow user who are permitted to login with vsftpd.user_list

```
User 1
User 2
User 3
```

allow user who are login with chroot in a jail with vsftpd.chroot_list

```
User 1
User 2
User 3
```

1.2.5 Webdav

Howto Setup Webdav (file transfer over http, like ftp)

I Prerequisites

- http server (nginx)
- webdav client (owncloud)

II Installation for Slackware

- use slackpkg
- use sbopkg
- use SlackBuild scripts

III Configuration

quick setup

- setup nginx
- setup owncloud

setup nginx [1]

- http://wiki.nginx.org/HttpDavModule
- http://docs.jelastic.com/nginx-webdav-module

global webdav, owncloud configuration

[1] see too: owncloud

- http://wiki.ubuntuusers.de/WebDAV
- https://kuther.net/blog/running-owncloud-webdav-nginx

footnotes

1.2.6 Databases

configuration of different databeses

MariaDB

Howto Setup MariaDB

I Prerequisites

1. mariadb-5.5.40-x86_64-2.txz
2. privileged user only for sql-data, mostly mysql

II Installation for Slackware

1. use slackpkg
2. use sbopkg
3. use SlackBuild scripts

III Configuration

Howto: http://docs.slackware.com/howtos:databases:install_mariadb_on_slackware

1. mysql_install_db –user=mysql
2. chown -R mysql.mysql /var/lib/mysql
3. chown 755 /etc/rc.d/rc.mysqld
4. /etc/rc.d/rc.mysqld start
5. mysqladmin -u root password 'NEW_PASSWORD'
6. use phpmyadmin to manage databases

```
root@gamma:~# mysql_install_db --user=mysql
Installing MariaDB/MySQL system tables in '/var/lib/mysql' ...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MariaDB root USER !
To do so, start the server, then issue the following commands:

'/usr/bin/mysqladmin' -u root password 'new-password'
'/usr/bin/mysqladmin' -u root -h gamma password 'new-password'

Alternatively you can run:
'/usr/bin/mysql_secure_installation'
```

```
which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the MariaDB Knowledgebase at http://mariadb.com/kb or the
MySQL manual for more instructions.

You can start the MariaDB daemon with:
cd '/usr' ; /usr/bin/mysqld_safe --datadir='/var/lib/mysql'

You can test the MariaDB daemon with mysql-test-run.pl
cd '/usr/mysql-test' ; perl mysql-test-run.pl

Please report any problems at http://mariadb.org/jira

The latest information about MariaDB is available at http://mariadb.org/.
You can find additional information about the MySQL part at:
http://dev.mysql.com
Support MariaDB development by buying support/new features from
SkySQL Ab. You can contact us about this at sales@skysql.com.
Alternatively consider joining our community based development effort:
http://mariadb.com/kb/en/contributing-to-the-mariadb-project/

root@gamma:~#
```

IV Backup

http://wiki.ubuntuusers.de/MySQL/Backup http://www.mysqldumper.de/tutorials/

PHPmyadmin

Howto Setup PHPmyadmin

I Prerequisites

- databank mariadb or mysql
- webserver with php enabled
- phpmyadmin-4.2.11-noarch-1js.txz

II Installation

Basic Steps

- check mariadb
- check webserver configuration with php (fastcgi, info.php)
- set permissions of php sessions save path to webserver's user:group and 770 (grep session.save.path /etc/httpd/php.ini ; e.g. /var/lib/php)
- install phpmyadmin
- set permissions of phpmyadmin to webserver user:group (see nginx.conf, fastcgi.conf)
- mkdir config && chmod o+rw && set permission according to the webserver

- check setup url http://phpmyadmin/setup

III Configuration

basic steps

- go to and follow the steps http://phpmyadmin/setup

manual steps

1. cd phpMyAdmin
2. mkdir config # create directory for saving
3. chmod o+rw config # give it world writable permissions
4. cp config.inc.php config/ # copy current configuration for editing
5. chmod o+w config/config.inc.php # give it world writable permissions
6. open http://phpmyadmin/setup/ # see nginx config
7. mv config/config.inc.php . # move file to current directory
8. chmod o-rw config.inc.php # remove world read and write permissions
9. open http://phpmyadmin/
10. login with root login and password
11. start adminstration

- quick-setup: http://wiki.phpmyadmin.net/pma/Setup
- quick-install: http://wiki.phpmyadmin.net/pma/Quick_Install
- docs-config: http://docs.phpmyadmin.net/en/latest/config.html

MySQLDumper

Howto Setup MySQLDumper

I Prerequisites

1. MySQLDumper1.24.4.zip

II Installation for Slackware

1. unzip

III Configuration

http://www.mysqldumper.de/tutorials/

IV Backup

use cron

PostgreSQL

Howto Setup Postgresql

I Prerequisites

1. postgresql-5.1.tar.gz
2. privileged user only for sql-data
3. phpPgAdmin (manage tool)
4. barman (optional backup tool)

II Installation for Slackware

- use sbopkg

III Configuration

1. create user with username, give him a passwd and make sure to log in homedir /var/lib/pgsql/
2. create group with username or other
3. change permissions of /var/lib/pgsql/ to chown -R username:username
4. create database: su username -c “initdb -D /var/lib/pgsql/9.3/data –locale=de_DE.UTF-8 -A md5 -W”

```
root@host:~# su username -c "initdb -D /var/lib/pgsql/9.3/data --locale=de_DE.UTF-8 -A md5 -W"
could not change directory to "/root": Permission denied
The files belonging to this database system will be owned by user "username".
This user must also own the server process.

The database cluster will be initialized with locale "de_DE.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "german".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/pgsql/9.3/data ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
creating configuration files ... ok
creating template1 database in /var/lib/pgsql/9.3/data/base/1 ... ok
initializing pg_authid ... ok
Enter new superuser password:
Enter it again:
setting password ... ok
initializing dependencies ... ok
creating system views ... ok
loading system objects' descriptions ... ok
creating collations ... ok
creating conversions ... ok
creating dictionaries ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
```

```
loading PL/pgSQL server-side language ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
copying template1 to postgres ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

    postgres -D /var/lib/pgsql/9.3/data
or
    pg_ctl -D /var/lib/pgsql/9.3/data -l logfile start
```

5. change username and group in rc.postgresql
6. start the database: /etc/rc.d/rc.postgresql start

```
root@host:~# /etc/rc.d/rc.postgresql start
Starting PostgreSQL
waiting for server to start.... done
server started
```

IV Manage Database

- phpPgAdmin

login with the privileged user above in phpPgAdmin

V Backuptools

- pg_dump
- barman

Postgresql Backup

see: http://www.postgresql.org/docs/9.4/static/backup-dump.html

dump db pg_dump dbname > outfile

reload db psql dbname < infile

dump large db pg_dump dbname | gzip > filename.gz

reload gunzip -c filename.gz | psql dbname

phpPgAdmin

Howto Setup phpPgAdmin

I Prerequisites

1. phpPgAdmin-5.1-noarch-1js

II Installation for Slackware

1. use slackpkg

2. use sbopkg
3. use SlackBuild scripts

III Configuration

Setup: http://phppgadmin.sourceforge.net/doku.php

1.2.7 Drupal

Howto install Drupal

I Prerequisites

- werbserver nginx and check php with info.php
- databank mariadb or postgresql
- phpmyadmin or phpPgAdmin
- databank for drupal

II Installation

Basic Steps

- download drupal
- move to target dir
- extract drupal
- configure drupal and set permissions
- install drupal

Advanced Steps

- check webserver with info.php:

```
<?php
phpinfo();
?>
```

- configure php (php.ini):

```
max_execution_time = 600
max_input_time = 600
mysql.connect_timeout = 600
```

- configure mariadb (my-large.cnf) for translation import very important:

```
innodb_flush_log_at_trx_commit = 2
```

- extract drupal:

```
tar xvzf drupal-7.34-DE.tar.gz
```

- move drupal to target dir:

```
mv drupal-x.x/* drupal-x.x/.htaccess ./
For Drupal 7, also add:
mv drupal-x.x/.gitignore ./
```

- configure drupal:

```
cp sites/default/default.settings.php sites/default/settings.php
chmod 664 sites/default/settings.php
chmod a+w sites/default
cp drupal-7.34.de.po profiles/standard/translations

# after installation
chmod 644 sites/default/settings.php
chmod 755 sites/default
```

- configure cron:

```
/usr/bin/curl -s $URL_A
```

III Guides

- setup drupal: https://www.drupal.org/start
- drupal installation: https://www.drupal.org/documentation/install
- drupal with nginx: http://wiki.nginx.org/Drupal
- translation-import: http://blog-das-oertchen.de/archive/201111/drupal-drupal-7-innodb-und-die-oberflaechenuebersetzung

IV Modules

https://www.drupal.org/project/ctools https://www.drupal.org/project/views https://www.drupal.org/project/webform https://www.drupal.org/project/panels

1.2.8 Django

Howto Setup a Nginx, Django, Postgresql, Gunicorn deployment

I Prerequisites

1. server: nginx
2. framework: django
3. databank: postgresql (mariadb)
4. python-translator: gunicorn (uwsgi)

Addons a. virtual environments: virtualenv b. documentation: sphinx c. maybe pip for easy installation

II Installation for Slackware

1. use slackpkg
2. use sbopkg
3. use SlackBuild scripts

III Configuration

Django-Setup

Howto Setup Django

I Prerequisites

1. webserver environment with php and database
2. framework: django
3. python-translator: gunicorn (uwsgi)

Addons a. virtual environments: virtualenv

II Installation for Slackware

1. use slackpkg
2. use sbopkg
3. use SlackBuild scripts

III Configuration

Gunicorn

howto setup and use gunicorn wsgi http server

http://gunicorn.org

I Prerequisites

- python
- nginx
- postgresql
- phpPgAdmin
- gunicorn
- django

II Quickstart

http://gunicorn.org/#quickstart

```
$ sudo pip install virtualenv
$ mkdir ~/environments/
$ virtualenv ~/environments/tutorial/
$ cd ~/environments/tutorial/
$ ls
bin  include  lib
$ source bin/activate
(tutorial) $ pip install gunicorn
(tutorial) $ mkdir myapp
(tutorial) $ cd myapp/
(tutorial) $ vi myapp.py
(tutorial) $ cat myapp.py
```

```
  def app(environ, start_response):
      data = "Hello, World!\n"
      start_response("200 OK", [
      ("Content-Type", "text/plain"),
      ("Content-Length", str(len(data)))
      ])
      return iter([data])

  (tutorial) $ ../bin/gunicorn -w 4 myapp:app

2010-06-05 23:27:07 [16800] [INFO] Arbiter booted
2010-06-05 23:27:07 [16800] [INFO] Listening at: http://127.0.0.1:8000
2010-06-05 23:27:07 [16801] [INFO] Worker spawned (pid: 16801)
2010-06-05 23:27:07 [16802] [INFO] Worker spawned (pid: 16802)
2010-06-05 23:27:07 [16803] [INFO] Worker spawned (pid: 16803)
2010-06-05 23:27:07 [16804] [INFO] Worker spawned (pid: 16804)
```

howto start apps (hello world)

```
# howto start a python app with gunicorn
# http://gunicorn.org/#quickstart

# 1. way gunicorn howto
gunicorn -w 4 hello:app

# 2. way
gunicorn hello:app -b localhost:8000

# 3. way
gunicorn --workers=2 hello:app

# 4. way
gunicorn hello:app

# look at
http://localhost:8000/

# or
http://127.0.0.1:8000
```

II Configuration

howto configure nginx for gunicorn https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-django-with-postgres-nginx-and-gunicorn

nginx global config

```
# user (should be a non privilleged)
user        amorsql amorsql; # user and group of processes

# worker or processes to run
worker_processes  2; # default 2

# events which can connect max. = 2x1024
events {
        worker_connections  1024; # default 1024
```

```
        accept_mutex off; # gunicorn
}

# http core functions
http {
        include                         mime.types;
        default_type                application/octet-stream;

        # gzip
        gzip                            on; # default on
        gzip_min_length                 5000;
        gzip_buffers                    4 8k;
        gzip_types                  text/plain text/css application/x-javascript text/xml application/x
        gzip_proxied                    any;
        gzip_comp_level                 2; # default 2

        # restrictions
        ignore_invalid_headers          on; # default on
        sendfile                  on; # default on
        client_max_body_size            3m; # default 3m

        # include sites
        include                         sites-enabled/*;
}
```

nginx gunicorn config

```
# gunicorn localhost config file for nginx
# http://docs.gunicorn.org/en/latest/deploy.html

# upstream server where gunicorn is listening see /etc/rc.d/rc.gunicorn
upstream app_server {
        server                  unix:/tmp/gunicorn.sock         fail_timeout=0;
        #server                  127.0.0.1:8000 fail_timeout=0;
        }

server {
        listen                  127.0.0.1:80;
        #listen                  127.0.0.1:443;
        client_max_body_size        4G; # default 4G
        server_name         gunicornapps; # add 127.0.0.1 gunicornapps to /etc/hosts
        #server_name         _;
        keepalive_timeout           5; # default 5
        root                 /var/www/nginx/gunicornapps;
        #charset         utf-8;

        location / {
                # checks for static file, if not found proxy to app
                try_files           $uri @proxy_to_app;
                allow          127.0.0.1; # localhost
                deny           all;
        }

        location @proxy_to_app {
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header Host $http_host;
                proxy_redirect off;
```

```
                proxy_pass   http://app_server; # send to upstream server
        }

        #error_page 500 502 503 504 /500.html;
        #location = /500.html {
        #root                /var/www/nginx/gunicornapps;
        #}
}
```

here you can find startscripts at bottom if needed (see next point) http://docs.gunicorn.org/en/latest/deploy.html

IV Running Djangoprojects

1. **create a django project::** django-admin.py startproject mysite
2. create a database in postgresql (more stable as other databases) use phpPgAdmin
3. change database setting in settings.py
4. **sync database and create the superuser for the djangoproject::** python manage.py syncdb
5. **the most important ;-) start and test your application with gunicorn::** gunicorn mysite.wsgi:application

see: https://docs.djangoproject.com/en/1.4/howto/deployment/wsgi/gunicorn/ 6. create an app:

```
python manage.py startapp [appname]
```

7. etc.

1.2.9 Github

Howto Setup Github

I Prerequisites

- git
- ssh
- login on github

II Installation

- not necessary

III Configuration

quick setup

- create ssh key
- add ssh key
- create a repo in github
- push your changes to your repo

setup git [1]

- https://help.github.com/articles/set-up-git/

global git configuration

```
git config --global user.name "YOUR NAME"
git config --global user.email "YOUR EMAIL ADDRESS"
```

initialisize repo and push it to remote repo with same name

```
cd repo
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:githubuser/repo.git
git push -u origin master
```

IV How to use it

if every thing works and content changed you do mostly these steps

- git status
- git add –all
- git commit -m “commit”
- git push

simple howtos

- http://githowto.com/setup
- http://gitreal.codeschool.com/levels/1

V Other Dokumentation

create a repo

- https://help.github.com/articles/create-a-repo/
- https://help.github.com/articles/fork-a-repo/
- https://help.github.com/articles/be-social/

sync a fork

- https://help.github.com/articles/syncing-a-fork/
- https://help.github.com/articles/pushing-to-a-remote/

git docs

- http://git-scm.com/

[1] see too: man git or man gittutorial

VI Use the Repo

e.g. fork sphinx docs on readthedocs

- https://readthedocs.org/

footnotes

1.2.10 Backup

cool backup software

http://www.cyberciti.biz/open-source/awesome-backup-software-for-linux-unix-osx-windows-systems/

Rsnapshot

http://www.rsnapshot.org/

1.2.11 Readthedocs

a cool site for documentations

https://readthedocs.org/

1.2.12 Raspberry Pi

Slackware Arm Installation on Raspberry Pi

I Prerequisites

- a raspberry pi board
- slackwarearm

II Installation for Slackware

install slackwarearm [1]

- http://docs.slackware.com/howtos:hardware:arm:raspberrypi

III Configuration

quick setup

- ???

[1] see too: docs.slackware.com

footnotes

1.2.13 Sound

Howto Setup Sound

I Prerequisites

- alsa

II Installation for Slackware

- use slackpkg
- use sbopkg
- use SlackBuild scripts

III Configuration

quick setup

- lsmod | grep snd (check if snd_hda_intel is loaded)
- aplay -l (check sound devices)
- cat /proc/asound/card*/codec* | grep Codec
- alsamixer
- alsactl store
- kde system setup multimedia

setup sound [1]

- http://docs.slackware.com/howtos:hardware:audio_and_snd-hda-intel

global sound configuration

```
alsamixer
alsactl store
```

special sound configuration

```
echo "options snd-hda-intel model=auto" > /etc/modprobe.d/snd-hda-intel.conf
```

output of aplay -l

```
**** List of PLAYBACK Hardware Devices ****
card 0: PCH [HDA Intel PCH], device 0: CONEXANT Analog [CONEXANT Analog]
Subdevices: 1/1
Subdevice #0: subdevice #0
card 0: PCH [HDA Intel PCH], device 1: Conexant Digital [Conexant Digital]
Subdevices: 1/1
Subdevice #0: subdevice #0
card 0: PCH [HDA Intel PCH], device 3: HDMI 0 [HDMI 0]
```

[1] see too: /etc/modprobe.d/sound.conf

```
Subdevices: 1/1
Subdevice #0: subdevice #0
```

output of cat /proc/asound/card*/codec* | grep Codec

```
Codec: Conexant CX20588
Codec: Intel CougarPoint HDMI
```

footnotes

1.2.14 timezone

how to setup the timezone

change timezone:

```
timeconfig
```

sync time with server:

```
ntpdate pool.ntp.org
```

write to hardware clock:

```
hwclock -w
```

1.2.15 Mount

a quick start guide

I Prerequisites

1. mount
2. udisk

II Automount

see: https://help.ubuntu.com/community/Mount/USB see: https://help.ubuntu.com/community/RenameUSBDrive

first umount all partions!

- **use blkid to dump partitions::** blkid

NTFS

- **check labeling of ntfs partions::** ntfslabel /dev/sdc2
- **change label of ntfs partions::** ntfslabel /dev/sdc? label

XFS

- **check labeling of xfs partions::** xfs_admin -l /dev/sdc1
- **change label of xfs partions::** xfs_admin -L label /dev/sdc1

1.2.16 Colors

Howto find cool Colors

I Prerequisites

go to

- https://color.adobe.com/de/create/color-wheel/

1.3 Aufgabenliste

1.3.1 Unerledigt

- setup mini server for backup
- write translate python module howto "for polish translate -t pl -f de nein" see: https://github.com/terryyin/google-translate-python
- write sdiff short howto sdiff -o name.merge file1 file 2 and export EDITOR=/usr/bin/vim
- look for knowlege: http://docs.slackware.com/slackware:localization
- setup translate toolkit 1.12.0 for lokalize see: https://translate-toolkit.readthedocs.org/en/stable-1.12.0/index.html see-converters: https://translate-toolkit.readthedocs.org/en/stable-1.12.0/commands/index.html#tools
- setup webdav, owncloud
- setup drupal 6 website relaunch
- setup raspberry pi for mini ftp-, media- and webserver
- setup proxy
- setup sendmail
- setup firewall
- setup a mail server with postfix and dovecot see: https://thomas-leister.de/allgemein/sicherer-mailserver-dovecot-postfix-virtuellen-benutzern-mysql-ubuntu-server-xenial/ see: http://www.admin-magazin.de/Online-Artikel/Postfix-einrichten-und-absichern
- gpg einrichten see: gpg for beginners http://zacharyvoase.com/2009/08/20/openpgp/ see : GPG-Howto

https://help.ubuntu.com/community/GnuPrivacyGuardHowto

1.3.2 Interessantes

- sieve mailfilter einrichten
- kolab groupware
- owncloud
- pass (passwortmananger)
- Software: https://prism-break.org/en/categories/gnu-linux/

1.3.3 Erledigt

- read root mail with mutt
- setup django
- write rc.gunicorn
- setup gunicorn
- setup wireshark a network analyze tool
- setup kde lokalize
- setup translate a python google translate module for the shell
- setup phpPgAmdin
- setup postgresql
- setup drupal 7 website relaunch
- setup playonlinux
- setup minidlna
- setup mysqldumper
- setup rsnapshot
- setup git
- setup ftpserver
- setup readthedocs
- setup ftp for public use

CHAPTER 2

resources

http://aikidokyritz.de