

Ubuntu Root Server

- Einführung, Provider- und Distributionsauswahl
- Installation, LVM, RAID
- SSH-Absicherung
- Webserver (Apache)
- Mail-Server (Postfix, Dovecot)

Teil I

- **Einführung**
- Installation
- SSH-Absicherung
- Webserver (Apache)
- Mail-Server (Postfix, Dovecot)

Webhosting vs. Root-Server

- vordefiniertes Software-Angebot
- Administration per Web-Schnittstelle und durch den Provider
- mehrere User/Rechner
- kostengünstig
- Beliebige Programmauswahl
- root-Rechte
- Administration per ssh
- erfordert Know-how und Eigenverantwortung

Zwischending: virtueller Root-Server

Alternative: lokaler Server mit Internet-Standleitung

Welche Distribution?

- Red Hat Enterprise Linux oder SUSE Linux Enterprise (Novell)
- CentOS (kostenlose RHEL-Alternative)
- Debian oder Ubuntu

- **Ungeeignet**
 - openSUSE (24 Monate Updates)
 - Fedora (13 Monate Updates)

Dieser Vortrag gilt

- zu 100 % für Ubuntu
- zu 95 % für Debian
- zu 70 % für andere Distributionen

Warum Ubuntu?

- ausgezeichnete Release-Planbarkeit
 - 5-jähriger Update-Zeitraum für LTS-Versionen
 - gute Erreichbarkeit der Paketquellen
 - persönliche Präferenz
-
- unbedingt LTS-Version, z.B. 8.04 oder 10.04
(LTS = Long Time Support, d.h. 5 Jahre Updates)

Welcher Provider?

- Vertrauensfrage
- Administrationshilfen
 - Reboot/Reset-Funktion
 - Rescue-System (Live-System mit SSH-Zugang)
 - Neuinstallation
- Distributionsangebot
 - 64-Bit
 - LVM/RAID-Installation
 - Installation per VNC (leider nicht für Debian/Ubuntu)

Eckdaten

- CPU
- RAM
- Festplatten (zwei für RAID-1)
- inkludierter Traffic
- IP-Adressen
- Backup-Speicher
- Monitoring-Funktionen
- Preis

Teil II

- Einführung
- **Installation**
- SSH-Absicherung
- Webserver (Apache)
- Mail-Server (Postfix, Dovecot)

RAID

- RAID = Redundant Array of Independent Disks
- mehr Datensicherheit und/oder höhere Geschwindigkeit
- verschiedene RAID-Level:
hier nur **RAID-1** = Mirroring
- verschiedene Verfahren
(Hardware-/-BIOS/Software-RAID):
hier nur **Software-RAID**
- RAID ersetzt keine Backups!

RAID-Spezialfälle

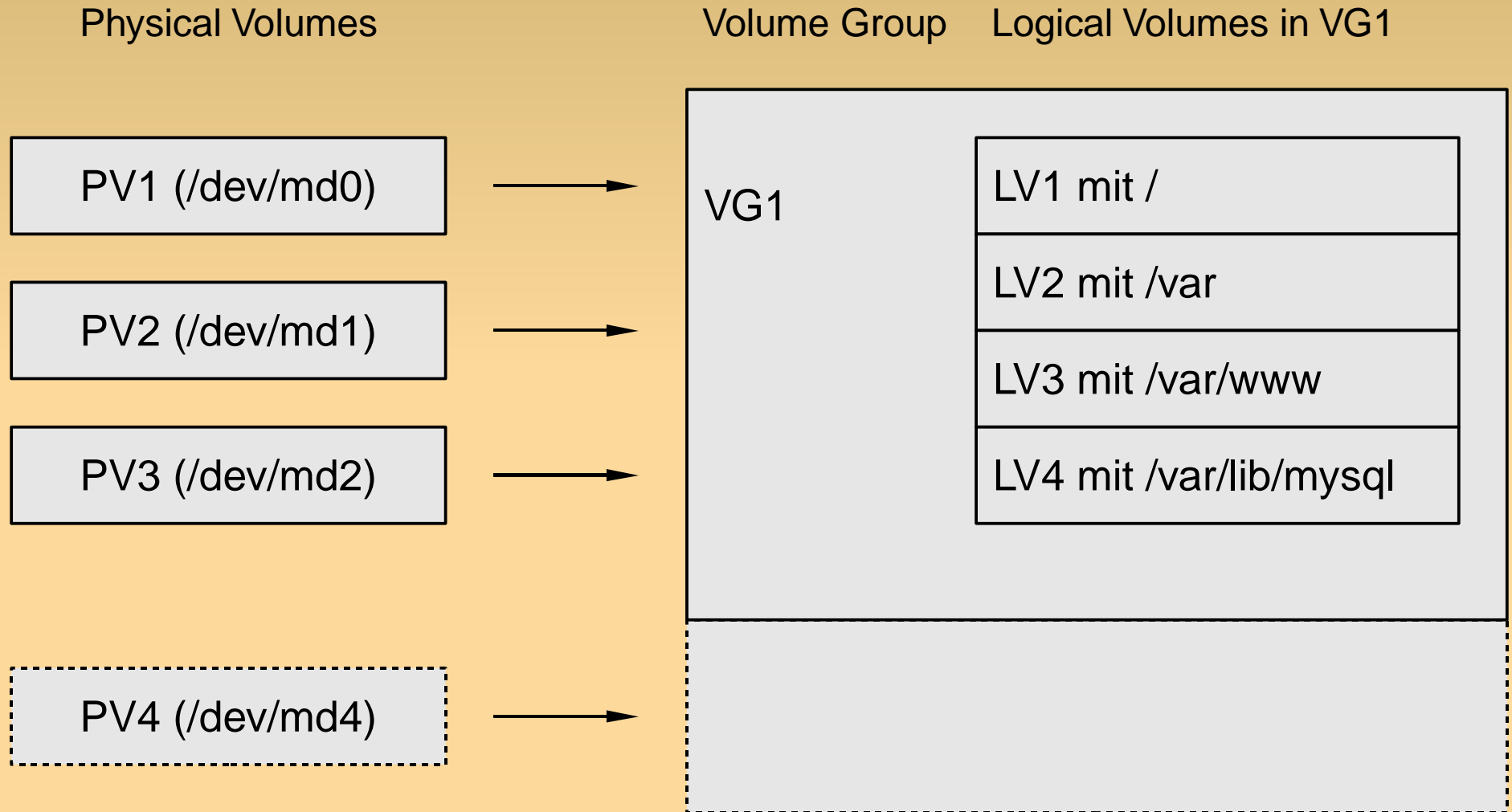
- GRUB + RAID
- Swap + RAID
- Neustart mit defekter Festplatte
- RAID-Monitoring erfordert lokalen E-Mail-Server (MTA)

Tipp: LVM- und RAID-Administration üben
(z.B. auf lokalem Testrechner oder
in einer virtuellen Maschine)

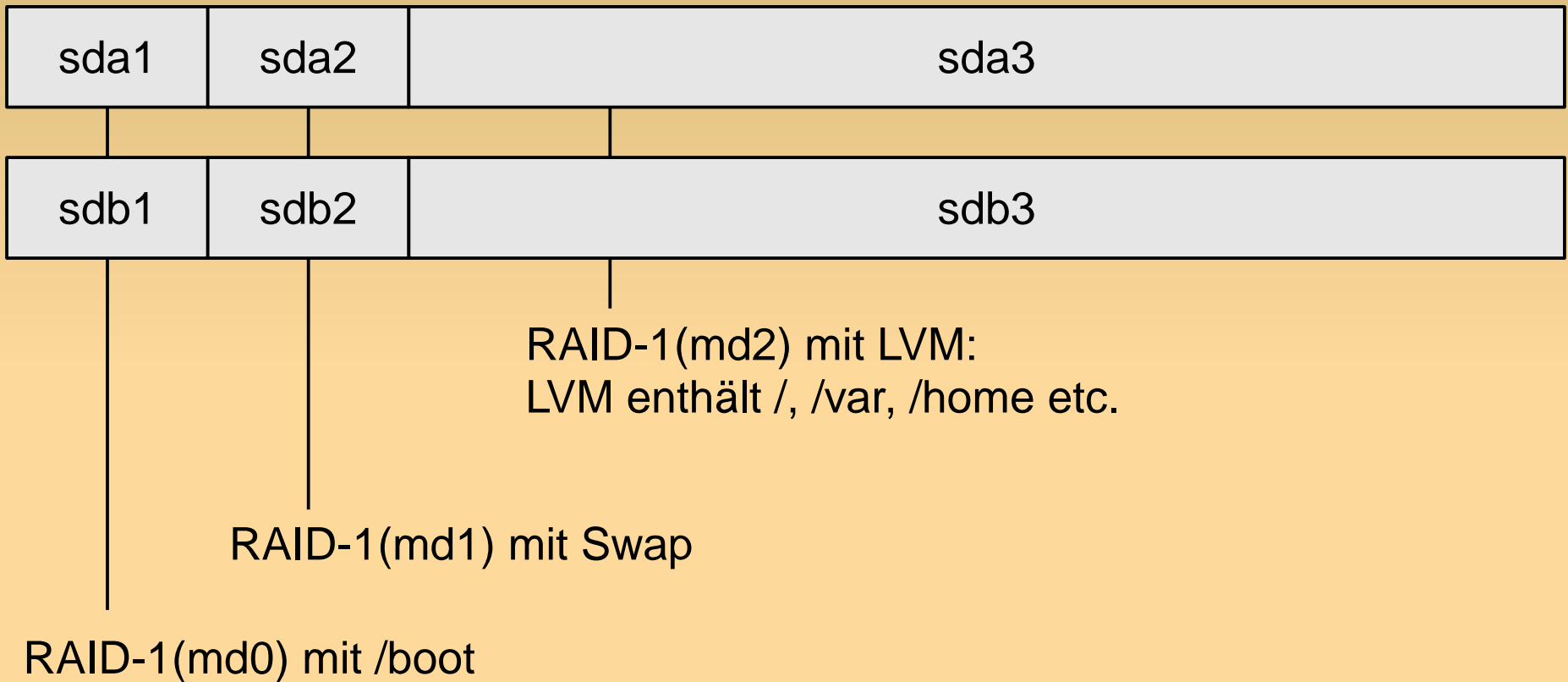
LVM

- LVM = Logical Volume Manager
- logische Schicht zwischen Festplatten und Partitionen
- erlaubt Veränderung der Partitionen im laufenden Betrieb
- Snapshot-Funktion, ideal für Backups (MySQL!)
- nur minimaler Geschwindigkeitsnachteil
- LVM + GRUB, LVM + Swap

LVM-Glossar



Installationsempfehlung



Installation

Debian/
Ubuntu-
spezifisch

- in der Regel vorgegebenes Minimal-System
- Umbau in RAID- und LVM-System ist möglich, aber aufwändig
- falls Sie die Installation selbst durchführen können (Ubuntu-Server-CD oder Debian)
 - zuerst alle physikalische Partitionen einrichten
 - dann RAID
 - dann LVM
 - zuletzt Nutzung der Partitionen bzw. LVs definieren

RAID-1-Tipps

Ubuntu-
spezifisch

- Ubuntu \geq 8.04.02 verwenden oder Update durchführen
- **grub-install /dev/md0** (mit /boot-Partition)
- # Datei /etc/initramfs-tools/conf.d/mdadm
BOOT_DEGRADED=true
update-initramfs -u
- # Datei /boot/grub/menu.lst
kopt=root=/dev/mapper/vgl-root ro **rootdelay=10**
update-grub

Teil III

- Einführung
- Installation
- **SSH-Absicherung**
- Webserver (Apache)
- Mail-Server (Postfix, Dovecot)

SSH

- SSH ist unverzichtbar
- Problem: automatisierte Angriff-Tools, die beliebte Loginnamen (root!) + Passwörter testen
- auf öffentlichen Servern sind Tausende Login-Versuche pro Tag normal!

Elementare Sicherheitsregeln

- keine Accounts ohne Passwort!
(siehe `/etc/shadow`)
- keine trivialen Passwörter
(eventuell durch `pam_cracklib` erzwingen)
- Ubuntu: generell kein direkter root-Login möglich

SSH – kein root-Login

- setzt voraus, dass User-Login und anschließender Wechsel in root-Login möglich ist (su/sudo); vorher testen!
- # in /etc/ssh/sshd_config
PermitRootLogin **no**
- **/etc/init.d/ssh reload**

SSH-Port ändern

- # in `/etc/ssh/sshd_config`
Port ***nnn*** (*standardmäßig 22*)
- **`/etc/init.d/ssh reload`**
- **`ssh -p nnn`**, aber **`scp -P nnn`**
- kein Schutz gegen Port-Scans
- kann zu Firewall-Problem führen

SSH-Authentifizierung mit Schlüsseln

- **client\$ ssh-keygen**
(Schlüsseldatei durch Passphrase verschlüsseln)
client\$ scp .ssh/id_rsa.pub user@server:key
- **server\$ mkdir -p .ssh**
server\$ cat key >> .ssh/authorized_keys
- **Login testen!**
- **# in /etc/ssh/sshd_config am Server**
PasswordAuthentication no
- **/etc/init.d/ssh reload**

Vorsicht: jetzt ist Login von einem beliebigen Rechner aus nicht mehr möglich!

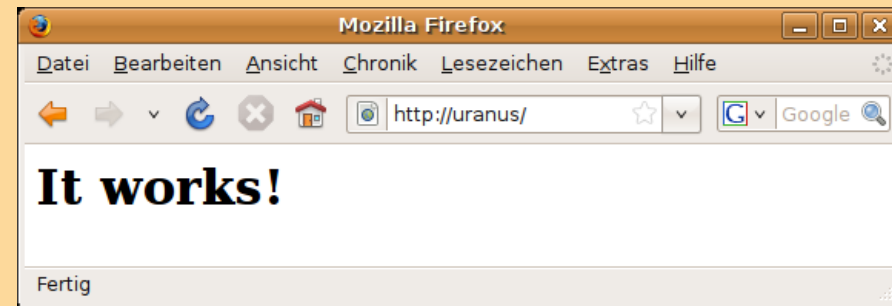
Teil IV

- Einführung
- Installation
- SSH-Absicherung
- **Webserver (Apache)**
- Mail-Server (Postfix, Dovecot)

Apache-Installation

Debian/
Ubuntu-
spezifisch

- **apt-get install apache2** (*worker-Threading*)
oder
apt-get install apache2 apache2-mpm-prefork
- mit Webbrowser **http://hostname**
- setzt DNS-Konfiguration für *hostname* voraus
- andernfalls **http://n.n.n.n** ansehen (IP-Adresse des Root-Servers)



Apache-Eckdaten

Debian/
Ubuntu-
spezifisch

Konfigurationsdateien	/etc/apache ² /*
DocumentRoot-Verzeichnis	/var/www/
Init-Script (Start/Stopp)	/etc/init.d/apache ²
Account für Apache	www-data
Logging-Dateien	/var/log/apache ² /*

Apache-Konfigurationsdateien

Debian/
Ubuntu-
spezifisch

<code>/etc/apache2/apache2.conf</code>	Startpunkt und Basiskonfiguration
<code>/etc/apache2/httpd.conf</code>	benutzerspezifische Konfiguration
<code>/etc/apache2/conf.d/*</code>	sonstige Konfigurationsdateien
<code>/etc/apache2/ports.conf</code>	überwachte Ports, normalerweise <code>^</code>
<code>webverzeichnis/.htaccess</code>	verzeichnisspezifische Konfiguration
<code>/etc/apache2/mods-available/*</code>	verfügbare Erweiterungsmodule
<code>/etc/apache2/mods-enabled/*</code>	Links auf aktive Erweiterungsmodule
<code>/etc/apache2/sites-available/*</code>	verfügbare Websites (virtuelle Hosts)
<code>/etc/apache2/sites-enabled/*</code>	Links auf aktive Websites
<code>/etc/apache2/envvars</code>	Umgebungsvariablen für <code>/etc/init.d/apache2</code>

Apache-Modulverwaltung

Debian/
Ubuntu-
spezifisch

- Links von `mods-enabled` nach `mods-available`
- Kommandos **`a2enmod`** / **`a2dismod`**
- danach: **`/etc/init.d/apache2 reload`**
- zusätzliche Module installieren:
Paketname `libapache2-mod-*`

Servername

falls beim Start die Warnung *could not reliably determine server's fully qualified domain name* erscheint:

in /etc/apache2/httpd.conf

Servername **meine-domain.de**

Zeichensatz

- Apache überträgt HTML-Dateien Byte für Byte
- header-Information über Zeichensatz
- standardmäßig gilt **AddDefaultCharset off**
Apache ermittelt den Zeichensatz aus HTML-Code
`<meta http-equiv="Content-Type"
content="text/html;charset=utf-8" />`
- alternativ kann der Zeichensatz durch **AddDefaultCharset xxx** fix eingestellt werden, wahlweise in `conf.d/charset` (global) oder in den Konfigurationsdateien für virtuelle Sites

Standard-Website

Debian/
Ubuntu-
spezifisch

Konfiguration durch
`/etc/apache2/sites-available/default:`

- DocumentRoot `/var/www`
- diverse Site-spezifische Einstellungen, Optionen für `/var/www`-Verzeichnis etc.
- enthält auch einige globale Grundeinstellungen (Logging, Aktivierung von namensbasierten virtuellen Hosts)

Virtuelle Hosts

Debian/
Ubuntu-
spezifisch

- ein Server – mehrere Websites
- drei Verfahren
 - namensbasiert (Hostname im HTTP-Header)
Einstellung **NameVirtualHost** * in sites-available/default
 - IP-basiert (IP-Adresse im Header)
 - Port-basiert
(eher unüblich, aber ideal für Tests)

Virtuellen Host einrichten

Debian/
Ubuntu-
spezifisch

- # Datei /etc/apache2/sites-available/firma-xy
<VirtualHost *>
 DocumentRoot /verzeichnisxy/
 ServerName www.firma-xy.de
 ServerAlias firma-xy.de
 ... (ErrorLog, CustomLog, ServerAdmin, ErrorDocument etc.)
</VirtualHost>
- DNS-Eintrag für firma-xy.de muss IP-Adresse des Root-Servers enthalten!
- **a2ensite firma-xy**
- **/etc/init.d/apache2 reload**

Verschlüsselte Verbindungen (HTTPS)

- erfordert Zertifikat und eigene IP-Adresse
- Zertifikat enthält öffentlichen Schlüssel und eine Unterschrift der Zertifizierungseinrichtung
- Zertifikat kann selbst erstellt werden (Kommando **openssl**)
- eigene Zertifikate
 - sind genau so abhörsicher wie kommerzielle Zertifikate
 - aber nicht vertrauenswürdig, weil die Identität des Zertifikaterstellers nicht überprüft wurde

Apache-HTTPS-Konfiguration

Debian/
Ubuntu-
spezifisch

- **a2enmod ssl**
- **cp server.pem server.crt /etc/apache2**
(unverschl. privater Schlüssel + Zertifikat)
- Datei sites-available/secure.firma-abc.de
- **a2ensite secure.firma-abc.de**
- **/etc/init.d/apache2 restart**

Apache-HTTPS-Konfiguration

```
# Datei sites-available/secure.firma-abc.de
<VirtualHost 1.2.3.5:443>
  ServerName secure.firma-abc.de
  DocumentRoot /verzeichnis-abc-secure
  SSLEngine on
  SSLCertificateFile /etc/apache2/server.crt
  SSLCertificateKeyFile /etc/apache2/server.pem
  ...
  <Directory /verzeichnis-abc-secure>
    ...
  </Directory>
</VirtualHost>
```

PHP

Debian/
Ubuntu-
spezifisch

- **apt-get install php5**
- Konfiguration:
`/etc/php5/apache2/php.ini`

MySQL

Debian/
Ubuntu-
spezifisch

- **apt-get install mysql-server**
- Passwort-Absicherung für root während der Installation
- Konfiguration: `/etc/mysql/my.cnf`
- Datenbankverzeichnis: `/var/lib/mysql`

MySQL-Grundkonfiguration

Debian/
Ubuntu-
spezifisch

- Error-Log in `/var/log/syslog`
- sonstiges Logging ist deaktiviert
- `bind-address=127.0.0.1`,
d.h. Netzwerkzugriff nur für localhost
- Start-Script `/etc/mysql/debian-start`,
MySQL-Benutzer `debian-sys-maint`,
dessen Passwort in `/etc/mysql/debian.cnf`

Logrotate

- **apt-get install logrotate**
- verarbeitet `/var/log/apache2/*.log`
einmal wöchentlich und
archiviert für 52 Wochen
- ignoriert individuelle Logging-Dateien
virtueller Hosts
- Anpassung in `/etc/logrotate.d/apache2`

Webalizer

- **apt-get install webalizer**
- Konfiguration: `/etc/webalizer/webalizer.conf`
- zusätzliche Webalizer-Konfigurationsdateien für virtuelle Hosts mit eigenen Logging-Dateien erforderlich

Webalizer + Logrotate

Debian/
Ubuntu-
spezifisch

- # in /etc/webalizer/*.conf
Incremental yes
...
- # in /etc/logrotate.d/apache2
prerotate
/etc/cron.daily/webalizer
postrotate
...

Teil V

- Einführung
- Installation
- SSH-Absicherung
- Webserver (Apache)
- **Mail-Server (Postfix, Dovecot)**

Ziel

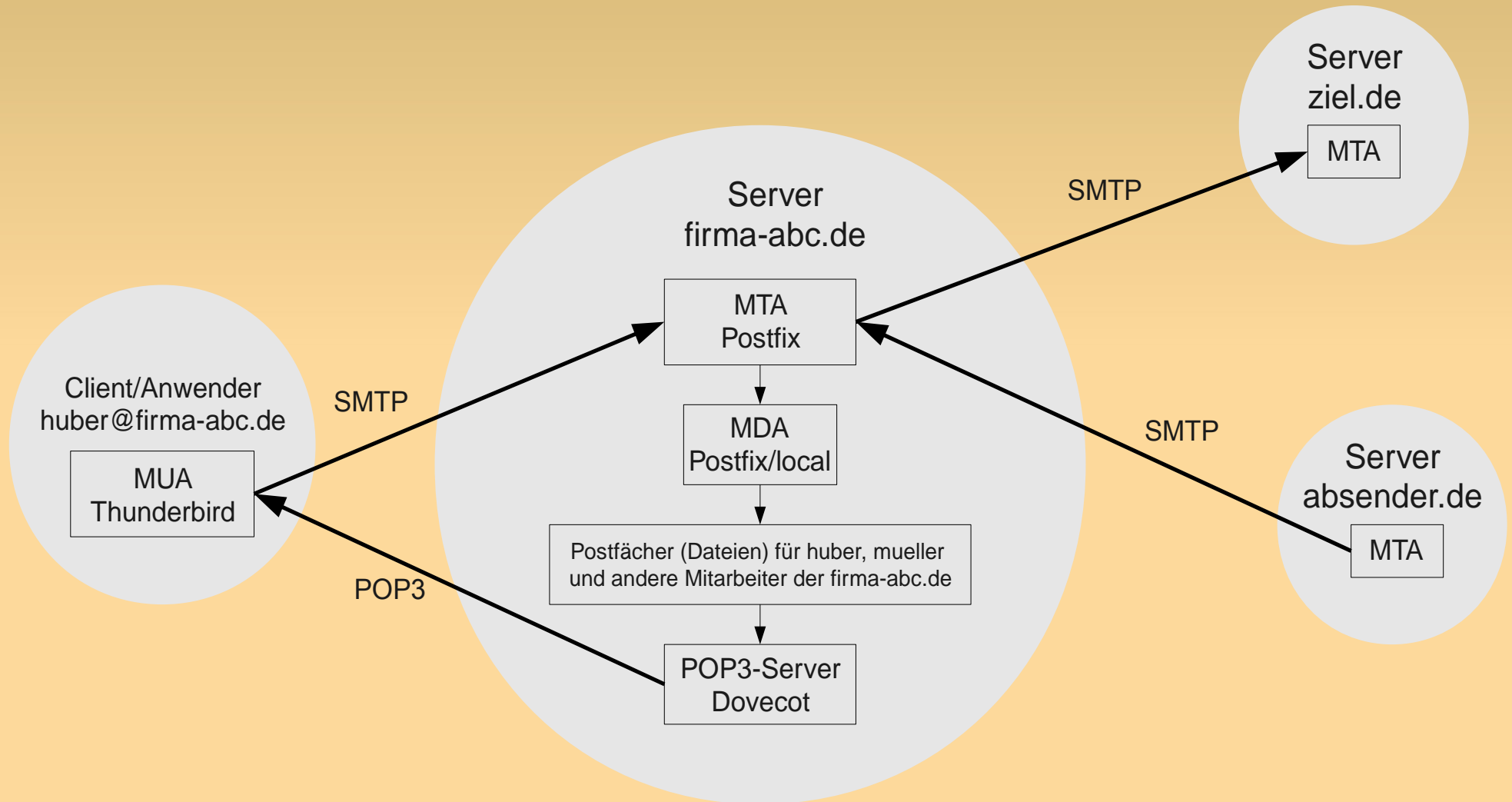
- E-Mail-Server für die firma-abc.de
- Server läuft auf einem Root-Server
- jeder Mitarbeiter bekommt eine Adresse (name@firma-abc.de)
- jeder Mitarbeiter kann per SMTP E-Mails versenden und per POP3 seine E-Mails herunterladen
- optional SPAM- und Virenschutz, IMAP, Web-Mail

Glossar

- MTA = Mail Transfer Agent ('Mail Server', z.B. Postfix)
- MDA = Mail Delivery Agent
- MUA = Mail User Agent (z.B. Thunderbird)

- SMTP = Simple Mail Transfer Protocol
- POP = Post Office Protocol
- IMAP = Internet Message Access Protocol

Überblick



DNS-Konfiguration

- A-Einträge ordnen Namen und IP-Adressen zu
- der MX-Eintrag gibt den Hostnamen an, der für die E-Mail einer Domain zuständig ist
- einfache Konfiguration für Web- und Mail-Server:

Typ	Name	Wert	Priorität
A	firma-abc.de	213.214.215.216	
A	www.firma-abc.de	213.214.215.216	
A	mail.firma-abc.de	213.214.215.216	
MX		mail.firma-abc.de	10

DNS-Konfiguration, Teil 2

- Reverse DNS-Eintrag:
Zuordnung IP-Adresse → Hostname
- dafür ist der Provider des Root-Servers verantwortlich; zumeist Web-Konfiguration möglich

DNS-Test

```
user$ host -t MX firma-abc.de
```

```
firma-abc.de mail is handled by 10 mail.firma-abc.de
```

```
user$ host mail.firma-abc.de
```

```
mail.firma-abc.de has address 213.214.215.216
```

```
user$ host firma-abc.de
```

```
firma-abc.de has address 213.214.215.216
```

```
firma-abc.de mail is handled by 10 mail.firma-abc.de.
```

```
user$ host 213.214.215.216
```

```
216.215.214.213.in-addr.arpa domain name pointer
```

```
firma-abc.de.
```

Postfix-Installation

Debian/
Ubuntu-
spezifisch

- **apt-get install postfix**
- Auswahl der Grundkonfiguration: **Internet Site**
- fertig!
 - Postfix empfängt E-Mails an *name@firma-abc.de*, sofern es den Linux-Account *name* gibt
 - Postfix speichert E-Mails in */var/mail/name* (mbox-Format)
 - lokale Benutzer können E-Mails versenden
 - Test dieser Funktionen mit **mutt**

Postfix-Konfiguration

- Konfigurationsdateien:
`/etc/postfix/main.cf`, `/etc/postfix/*` und `/etc/aliases`
- Sonderfall **einstellung = hash:/dateiname**
die Daten werden aus BDB-Datenbankdatei
`dateiname.db` gelesen;
nach Änderungen der zugrundeliegende Textdatei
muss die Datenbankdatei aktualisiert werden:
postmap dateiname
(bei `/etc/aliases`: Kommando **newaliases**)

main.cf

Hostname

myhostname = firma-abc.de

Domain für lokale E-Mails ohne explizite Domainangabe

/etc/mailname enthält in der Beispielkonfiguration firma-abc.de

myorigin = /etc/mailname

Ort der Alias-Datei

alias_maps = hash:/etc/aliases (gilt für Postfix; es kann weitere Alias-Quellen geben)

alias_database = hash:/etc/aliases (gilt für newaliases)

Versand neuer E-Mails nur vom lokalen Rechner zulassen

mydestination = firma-abc.de, localhost

mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128

nicht-lokale E-Mails selbst zustellen (kein Relaying)

relayhost =

E-Mail-Empfang über alle Netzwerkschnittstellen

inet_interfaces = all

keine Beschränkung der E-Mail- und Postfach-Größe

mailbox_size_limit = 0

Mail-Aliase

- ordnet E-Mail-Namen Login-Namen zu
- Beispiel:
Datei /etc/aliases
postmaster: root
webmaster: huber
bernhard.huber: huber
- nach Änderungen: **newaliases**

Explizite Empfängerliste

- standardmäßig empfängt jeder Linux-Account E-Mails (auch daemon, sys, man etc.); Abhilfe:
- # Datei /etc/postfix/local-recips
fischer x
huber x
root x
- **cd /etc/postfix**
postmap local-recips
- # in /etc/postfix/main.cf
local_recipient_maps = \
hash:/etc/postfix/local-recips \$alias_maps
- weiterhin Linux-Account erforderlich!

Dovecot

- POP3-Server
- IMAP-Server
- SMTP-Authentifizierung für Postfix
- Kommunikation SSL-verschlüsselt

Dovecot-Installation

Debian/
Ubuntu-
spezifisch

- **apt-get install dovecot-pop3**
- Konfigurationsdatei von 1100 auf 22 Zeilen kürzen:
cd /etc/dovecot
cp dovecot.conf dovecot.orig
grep -v '^[[[:space:]]*\#' dovecot.orig |
grep -v '^[[[:space:]]*\$' > dovecot.conf
- automatisch generiertes Zertifikat nur für 1 Monat!
- findet Postfächer in der Regel selbstständig
- wenn nicht:
in /etc/dovecot/dovecot.conf
mail_location = mbox:~/Mail:INBOX=/var/mail/%u

Zertifikat neu generieren

Debian/
Ubuntu-
spezifisch

```
cd /etc/ssl
```

```
cp private/ssl-cert-snakeoil.key private/ssl-cert-snakeoil.bak
```

```
cp certs/ssl-cert-snakeoil.pem certs/ssl-cert-snakeoil.bak
```

```
openssl genrsa -out server.key 1024
```

```
openssl req -new -x509 -key server.key -out server.pem -days 1826
```

...

```
Country Name (2 letter code) [AU]: de
```

```
State or Province Name (full name) [Some-State]: none
```

```
Locality Name (eg, city) []: Berlin
```

```
Organization Name (eg, company): firma-abc
```

```
Organizational Unit Name (eg, section) []:
```

```
Common Name (eg, YOUR name) []: firma-abc.de
```

```
Email Address []: postmaster@firma-abc.de
```

```
mv server.key private/ssl-cert-snakeoil.key
```

```
mv server.pem certs/ssl-cert-snakeoil.pem
```

```
/etc/init.d/dovecot restart
```

Betrieb als POP-Server

- funktioniert auf Anhieb
- Client-Konfiguration für POP3
 - Verschlüsselungsverfahren SSL oder TLS
 - Benutzername + Passwort wie am Linux-Account
 - nur Klartextpasswörter!
(das ist *kein* Sicherheitsrisiko, weil ja die ganze Kommunikation verschlüsselt erfolgt;
erforderlich, damit Dovecot zu PAM kompatibel ist)
 - Thunderbird: Option 'Sichere Authentifizierung' *nicht* verwenden
 - beim ersten Verbindungsaufbau muss das Dovecot-Zertifikat akzeptiert werden

SMTP-Authentifizierung für Postfix

- Postfix unterstützt SASL (Simple Authentication and Security Layer), kann die Authentifizierung aber nicht selbst durchführen
- Dovecot hilft

Postfix-Authentifizierung, Teil 2

Debian/
Ubuntu-
spezifisch

```
# Ergänzungen in /etc/dovecot/dovecot.conf
...
auth default {
    mechanisms = plain login
    ...
socket listen {
    client {
        path = /var/spool/postfix/private/auth
        mode = 0660
        user = postfix
        group = postfix
    }
}
}
```

Postfix-Authentifizierung, Teil 3

Debian/
Ubuntu-
spezifisch

Ergänzung in /etc/postfix/main.cf

```
...  
smtpd_sasl_auth_enable = yes  
smtpd_sasl_type = dovecot  
smtpd_sasl_path = private/auth  
smtpd_recipient_restrictions = permit_mynetworks,  
                                  permit_sasl_authenticated,  
                                  reject_unauth_destination
```

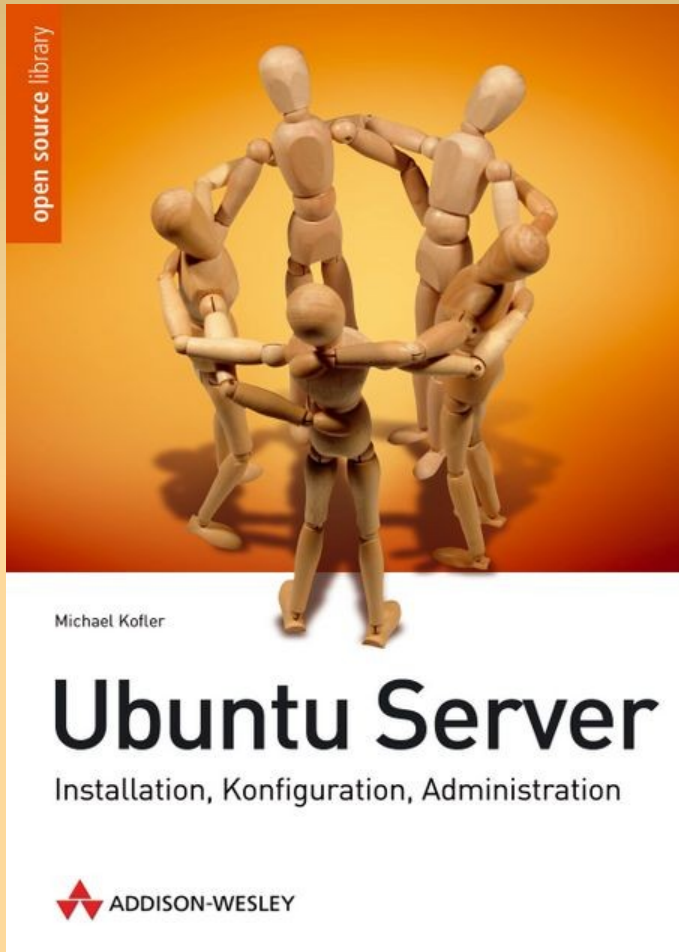
danach:

```
/etc/init.d/dovecot restart  
/etc/init.d/postfix reload
```

Spam- und Virenschutz

- Vorgehensweise
 - E-Mails aufgrund formaler Fehler sofort abweisen (z.B. wegen fehlendes Reverse-DNS-Eintrags, offensichtlich falschen DNS-Angaben etc.)
 - E-Mails nach einer Inhaltsanalyse abweisen oder als Spam/Virus kennzeichnen
- Programme
 - policyd-weight (wird nicht mehr gewartet)
 - SpamAssassin
 - ClamAV

Vielen Dank



- **Installation/Administration**
(LVM, RAID, Logging, Paketverwaltung)
- **LAN-Server/Home-Server**
(Samba, NFS, LDAP, Kerberos)
- **Root-Server**
(Apache, MySQL, Postfix, Dovecot)
- **Sicherheit**
(Firewall, VPN, AppArmor, Backups)

Weitere Infos: www.kofler.cc

Vortrag als PDF: www.kofler.cc/userver-vortrag.pdf