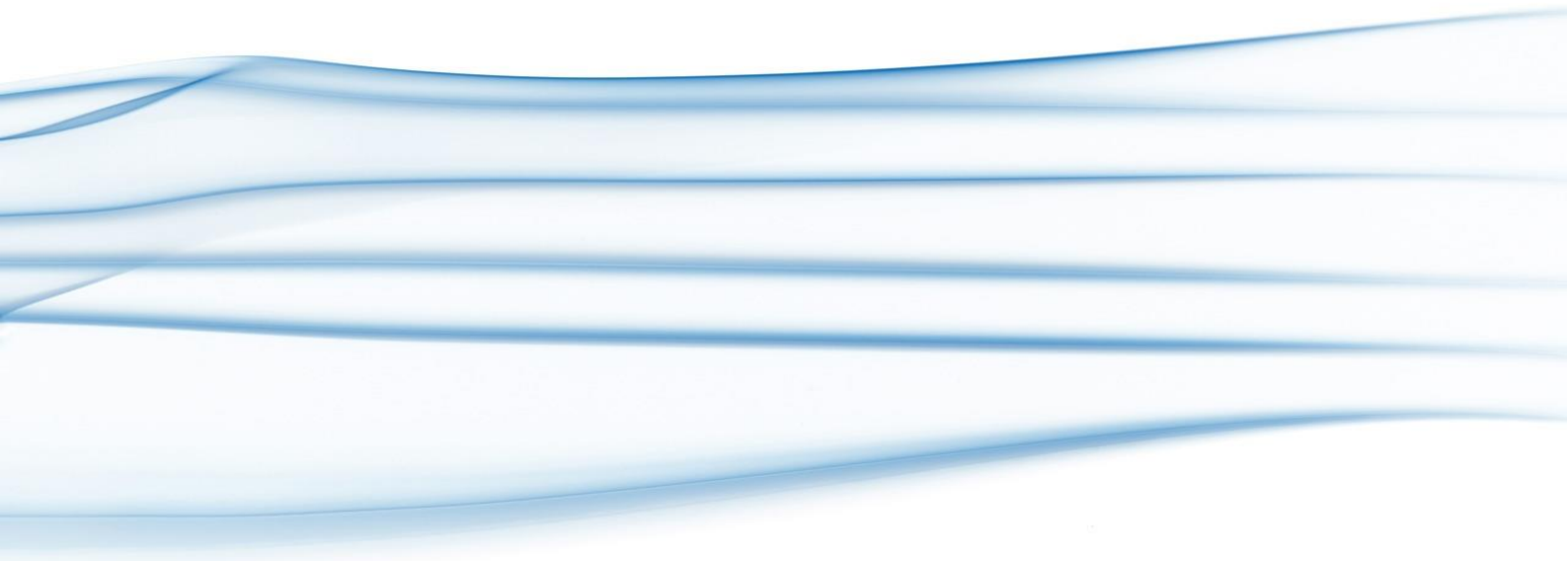
Installation eines Root-Servers

auf Basis von LEMP und Ajenti-V

Version 03.11.2014



Inhaltsverzeichnis

Information zu dieser Anleitung

Autor

J.F.
eMail: j-f@gmx-topmail.de

Nutzungsbedingungen

Haftungsausschluss

Warenzeichen

Hinweise zur Verwendung

Die einzelnen Abschnitte dieser Anleitung bauen aufeinander auf und deshalb sollte die Reihenfolge bei der Umsetzung eingehalten werden.

Auf Screenshots und Grafiken wurde weitestgehend verzichtet, stattdessen wurden folgende Formatierungen für eine bessere Lesbarkeit verwendet.

```
Eingabe in der Konsole
```

```
Eingaben in einer Eingabemaske (z.B. Installationsroutine oder Webseite)
```

```
Auszug aus einer Textdatei
```

```
Vollständiger Inhalt einer Textdatei
```

```
Meldungen und Hinweise
```

```
********      Kennzeichnung für Kennwörter die selbst zu vergeben sind.
```

1. Zielsetzung

Diese Anleitung beschreibt die Installation eines Servers mit Debian Wheezy auf einem Root-Server von netcup und richtet sich dabei besonders an Einsteiger, die aufgrund der Komplexität dieser Thematik schnell die Übersicht verlieren.

Für die webbasierte Verwaltung des Systems kommt Ajenti zum Einsatz. Das in Python geschriebene Admin-Panel ist ein recht junges Projekt und daher noch nicht so verbreitet. Es ist modular aufgebaut und bietet eine sehr moderne Oberfläche, auf der sich auch Einsteiger sehr schnell zurechtfinden. Es sei aber noch darauf hingewiesen, dass Ajenti in der aktuellen Version nicht als Reseller- und Kunden-Panel ausgelegt ist und somit nur für die Verwaltung des eigenen Servers geeignet ist.

Neben der eigentlichen Funktionalität liegt ein weiterer Schwerpunkt dieser Anleitung bei einer möglichst geringen Hardwareanforderung und der Absicherung vor verschiedenen Angriffsszenarien, so dass dieses Tutorial gerade für kleine Root-Server gut geeignet ist.

System-Informationen

Die im Folgenden beschriebene Installation wurde mehrfach auf einem Root-Server von netcup auf Basis von KVM getestet. Eine Übersicht der verwendeten Hard- und Softwareumgebung und der Funktionen, die nach Abschluss der Installation zur Verfügung stehen, sind im Folgenden aufgeführt. Der verwendete Hauptspeicher ist natürlich das absolute Minimum und daher auch nur für sehr kleine Umgebungen geeignet.

Hardware

```
Hoster:      netcup
Produkt:     Root-Server Weltmeister (KVM)
Hostname:    v1234567890.yourvserver.net
CPU:         1 vCore 2,6 GHz (Intel® Xeon® Westmere E56xx)
RAM:         512 MB DDR 3
HDD:         40 GB
```

Software

```
System:      Debian 7.6.0 Wheezy
Webserver:   Nginx 1.6.2-1
PHP:         5.6.2-1 (PHP5-FPM)
Datenbank:   MySQL 5.6.19
SMTP:        Exim 4.80-7
IMAP/POP3:   Courier 4.10.0 / 0.68.2-1
FTP:         Pure-FTPd 1.0.36-1.1
Spamschutz:  SPF, DKIM, DMARC, greylistd 0.8.8
IPS:         iptables, fail2ban 0.8.6-3
Monitoring:  Munin 2.0.6-4, Logwatch 7.4.0
Admin-Panel: Ajenti + Ajenti-V 1.2.22.16, phpMyAdmin 3.4.11.1-2
Backup:      Rsync 3.0.9-4, bzip2 1.0.6-4
```

2. Installation Debian 7.6.0

Basis für den Root-Server ist eine minimale Installation von Debian 7.6.0 (Wheezy). Für die Installation wird das kleine Debian 7.6.0 Installations-Image benötigt. Dieses kann über die Debian-Seite oder über die hier aufgeführten URLs heruntergeladen werden.

An dieser Stelle muss man sich entscheiden, ob man ein 32Bit- oder 64Bit-System installieren will. Bei weniger als 1GB Hauptspeicher ist ein 32Bit-System die bessere Wahl, denn 64Bit-Systeme haben bei gleicher Konfiguration einen etwas höheren Hauptspeicherbedarf. Bei der hier beschriebenen Installation macht das am Ende eine Differenz von ca. 100 bis 150 MB aus.

```
64Bit -> http://cdimage.debian.org/debian-cd/7.6.0/amd64/iso-cd/debian-7.6.0-amd64-netinst.iso
32Bit -> http://cdimage.debian.org/debian-cd/7.6.0/i386/iso-cd/debian-7.6.0-i386-netinst.iso
```

Um die Installation zu starten muss das ISO-Image als CD/DVD auf dem Root-Server eingebunden werden. Hierzu ist es bei netcup erforderlich, dass ISO-Image mittels FTP auf einen Upload-Server zu laden. Die Zugangsdaten für den FTP-Server können über das VCP abgerufen werden.

```
netcup vcp -> v1234567890 -> CDROM -> Eigene CDRoms
```

Nachdem das ISO-Image auf dem Server bereitgestellt und als CD/DVD eingebunden wurde, kann der Server gestartet werden. Für eine Minimalinstallation von Debian sind folgende Schritte notwendig.

```
Advanced options -> [Enter]
Expert install -> [Enter]
Choose language -> [Enter]
Language: German – Deutsch -> [Enter]
Land oder Gebiet: Deutschland -> [Enter]
Land, das zur Bestimmung...: Deutschland -> de_DE.UTF-8 -> [Enter]
Zusätzliche Gebietsschemata: [x] de_DE -> <Weiter> [Enter]
System-Gebietsschema: de_DE.UTF-8 -> [Enter]
Tastatur konfigurieren -> [Enter]
Deutsch -> [Enter]
CD-ROM erkennen und einbinden -> [Enter]
Diese Module laden: [ ] usb-storage (USB storage) -> <Weiter> [Enter]
CD-ROM gefunden -> <Weiter> [Enter]
Installer-Komponenten von CD laden -> [Enter]
Installer-Komponenten -> <Weiter> [Enter]
Netzwerk-Hardware erkennen -> [Enter]
Diese Module laden: [ ] usb-storage (USB storage) -> <Weiter> [Enter]
Netzwerk einrichten -> [Enter]
```

Die IP-Adresse eines Root-Servers wird in aller Regel fest vorgegeben, aber die Zuweisung kann dennoch automatisch erfolgen. Wird dies vom Hoster nicht unterstützt, muss die IP-Konfiguration manuell vorgenommen werden.

```
Netzwerk automatisch einrichten? -> <Ja> [Enter]
Wartezeit (in Sekunden) für Erkennung einer Verbindung: 3 -> <Weiter> [Enter]
Rechnername: v1234567890 -> <Weiter> [Enter]
Domain-Name: yourvserver.net -> <Weiter> [Enter]
Benutzer und Passwörter einrichten -> [Enter]
Shadow-Passwörter benutzen? -> <Ja> [Enter]
Root das Anmelden erlauben? -> <Ja> [Enter]
Root-Passwort: ******** -> <Weiter> [Enter]
Bitte geben sie das Passwort zur Bestätigung nochmals ein: ******** -> <Weiter> [Enter]
Soll jetzt ein normales Benutzerkonto erstellt werden? -> <Nein> [Enter]
Uhr einstellen -> [Enter]
```

Handelt es sich bei dem Server um eine virtuelle Maschine, wird die Zeit vom Hostsystem übernommen. In diesem Fall muss NTP deaktiviert werden.

```
Die Uhr mittels NTP einstellen? -> <Nein> [Enter]
Wählen Sie Ihre Zeitzone: Europe/Berlin -> [Enter]
Festplatten erkennen -> [Enter]
Diese Module laden: [ ] usb-storage (USB storage) -> <Weiter> [Enter]
Festplatten partitionieren -> [Enter]
```

Zur Partitionierung von Festplatten gibt es recht unterschiedliche Meinungen. Oft wird empfohlen, Bereiche, die möglicherweise schnell volllaufen, auf extra Partitionen zu verlagern. So soll verhindert werden, dass das gesamte System in Mitleidenschaft gezogen wird, denn es gibt einige Angriffsszenarien, die genau dies erreichen wollen. Auch wird mit einer Aufteilung auf mehrere Partitionen eine übergreifende Fragmentierung unterbunden.

Die Partitionierung für den hier verwendeten Root-Server könnte daher wie folgt aussehen.

```
/          3 GB
/boot    100 MB
/tmp       1 GB
/var      15 GB
/srv      20 GB
swap       1 GB
```

Verwendet man eine solche Aufteilung, muss man sich sehr genau überlegen, wie viel Speicherplatz den Partitionen zugeordnet werden soll, denn sonst kommt schnell zu einer ungünstigen Verteilung von freiem und belegtem Speicherplatz. Bei kleineren Festplatten macht es daher durchaus Sinn auf diese Aufteilung zu verzichten und den gesamten Speicherplatz damit besser auszunutzen. In diesem Beispiel wird daher auch das einfache Partitionierungsschema verwendet.

```
Partitionierungsmethode: Geführt – vollständige Festplatte verwenden -> [Enter]
Wählen Sie die zu partitionierende Festplatte: Virtuelle Festplatte 1 (vda) – 42,9 GB Virtio
lock Device -> [Enter]
Partitionierungsschema: Alle Dateien auf eine Partition, für Anfänger empfohlen -> [Enter]
Partitionierung beenden und Änderungen übernehmen -> [Enter]
Änderungen auf die Festplatten schreiben? -> <Ja> [Enter]
Grundsystem installieren -> [Enter]
Zu installierender Kernel: linux-image-686-pae bzw. linux-image-amd64 -> [Enter]
In die initrd aufzunehmende Treiber: angepasst: nur für das System benötigte Treiber einbinden
-> [Enter]
Paketmanager konfigurieren -> [Enter]
Einen Netzwerkspiegel verwenden? -> <Ja> [Enter]
Protokoll für Datei-Downloads: http -> [Enter]
Land des Debian-Archiv-Spiegelservers: Deutschland -> [Enter]
Debian-Archiv-Spiegelserver: ftp.de.debian.org -> [Enter]
http-Proxy-Daten: -> <Weiter> [Enter]
>>Non-free<<-Software verwenden? -> <Nein> [Enter]
>>Contrib<<-Software verwenden? -> <Nein> [Enter]
Zu verwendende Dienste: [x] Sicherheitsaktualisierungen [x] Release-Updates -> <Weiter>
[Enter]
Software auswählen und installieren -> [Enter]
An der Paketverwendungserfassung teilnehmen? -> <Nein> [Enter]
Möchten Sie man und mandb >>setuid man<< installieren? -> <Nein> [Enter]
Welche Software soll installiert werden? [ ] alle Dienste abwählen -> [Enter]
GRUB-Bootloader auf einer Festplatte installieren -> [Enter]
Den GRUB-Bootloader in den Master Boot Record installieren? -> <Ja> [Enter]
Installation abschließen - [Enter]
Ist die Systemzeit auf UTC gesetzt? -> <Ja> [Enter]
Installation abgeschlossen -> <Weiter> [Enter]
```

Die Grundinstallation von Debian ist abgeschlossen. Wird der Root-Server auf einer virtuellen Maschine eingerichtet, ist nun ein guter Zeitpunkt für einen ersten Snapshot.

3. Vorbereitung des Servers

Für die weitere Installation und Konfiguration des Servers sind einige Vorbereitungen erforderlich.

Installation OpenSSH

Ein Großteil der Installation und Konfiguration und ein Teil der Administration werden über die Konsole erfolgen, daher ist ein SSH-Zugriff auf dem Server essentiell. Dieser wird mittels OpenSSH installiert.

```
apt-get install openssh-server
```

Eine Vielzahl von zumeist automatisierten Angriffen richtet sich auf den Standard-Port für SSH (TCP/22). Aus diesem Grund macht es durchaus Sinn den Port für SSH zu ändern.

```
nano /etc/ssh/sshd_config
```

```
Port 8002
```

Nachdem die Konfigurationsdatei gespeichert und der SSH-Dienst neu gestartet wurde, kann man (z.B. mit PuTTY) auf die SSH-Konsole zugreifen.

```
service ssh restart
```

IPv6 deaktivieren

Bei dieser Anleitung wird von einer reinen IPv4-Adressierung ausgegangen, weshalb IPv6 nicht benötigt wird und daher deaktiviert werden kann bzw. sollte. So wird unter anderem verhindert, dass man über ein mögliches Router Advertisement eine IPv6-Adresse erhält, denn die hier verwendete iptables-Konfiguration ist nicht für IPv6 gültig ist. Dies könnte sonst dazu führen, dass Dienste per IPv6 zugänglich sind, obwohl sie mit einer IPv4-Regel geblockt werden.

Zum Deaktivieren von IPv6 ist es am einfachsten, einen sysctl-Parameter zu setzen. Hierzu muss eine Datei mit dem entsprechenden Parameter erstellt werden.

```
echo "net.ipv6.conf.all.disable_ipv6=1" >/etc/sysctl.d/disableipv6.conf
sysctl -p /etc/sysctl.d/disableipv6.conf
```

Die IPv6-Auflösung sollte ebenfalls deaktiviert werden. Dazu muss man die IPv6-Adressen in der *„/etc/hosts“* auskommentieren.

```
nano /etc/hosts
```

Die Datei sollte dann so aussehen.

```
127.0.0.1       localhost
127.0.1.1       v1234567890.yourvserver.net      v1234567890

# The following lines are desirable for IPv6 capable hosts
#::1     localhost ip6-localhost ip6-loopback
#ff02::1 ip6-allnodes
#ff02::2 ip6-allrouters
```

MX-Eintrag und Reverse DNS

Da später auch ein Mailserver installiert werden soll, ist im DNS ein MX- (Mail Exchange) und RDNS-Eintrag erforderlich. Bei netcup können diese Einträge wie folgt eingerichtet werden.

```
netcup ccp -> Domains -> mydomain.de -> DNS
Host: @
Type: MX
MX: 10
Destination: v1234567890.yourvserver.net

[Speichern]
```

```
netcup ccp -> Produkte -> v1234567890 - Root-Server Weltmeister -> RDNS
Hostname: v1234567890.yourvserver.net

[ändern]
```

Dotdeb-Paketquellen

Die Standard-Quellen für Debian-Pakete beinhalten zwar stabile und umfangreich getestete Programm-Versionen, doch dafür sind sie oft nicht besonders aktuell. Da Nginx, PHP und MySQL aber in möglichst aktueller Version installiert werden sollen, müssen zusätzliche Paketquellen hinzugefügt werden.

```
echo "deb http://packages.dotdeb.org wheezy all" >> /etc/apt/sources.list
echo "deb-src http://packages.dotdeb.org wheezy all" >> /etc/apt/sources.list
echo "deb http://packages.dotdeb.org wheezy-php56 all" >> /etc/apt/sources.list
echo "deb-src http://packages.dotdeb.org wheezy-php56 all" >> /etc/apt/sources.list
wget http://www.dotdeb.org/dotdeb.gpg -O- |apt-key add -
apt-get update
```

Zertifikate erstellen

Für SSL- / TLS-Verbindungen (https, imaps, smtps etc.) werden Zertifikate benötigt. Diese können von offiziellen Zertifizierungsstellen erworben oder wie in diesem Fall als selbstsignierte Zertifikate erstellt werden. Da selbstsignierte Zertifikate aber nicht von einer offiziellen Zertifizierungsstelle stammen, haben sie den Nachteil, dass sie von den meisten Browsern nur mit einer Warnmeldung akzeptiert werden. Die auf diesen Zertifikaten basierenden SSL-Verbindungen sind deswegen aber nicht weniger sicher, dem System ist eben nur der Herausgeber nicht bekannt. Im Gegensatz zu offiziellen Zertifikaten, die meist nur eine Gültigkeit von 1 Jahr haben, kann man bei einem selbstsignierten Zertifikate die Gültigkeitsdauer frei definieren. In diesem Beispiel sind es 10 Jahre (-days 3650).

Die Zertifikate werden mit Hilfe OpenSSL erstellt, was daher zuerst installiert werden muss.

```
apt-get install openssl
```

Für die Dienste, die unabhängig von den später gehosteten Domains auf dem Root-Server laufen, wird ein Zertifikat mit dem FQDN des Hostsystems benötigt.

```
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout
/etc/ssl/private/v1234567890.yourvserver.net.key -out
/etc/ssl/certs/v1234567890.yourvserver.net.crt
```

```
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:v1234567890.yourvserver.net
Email Address []:
```

```
cat /etc/ssl/private/v1234567890.yourvserver.net.key
/etc/ssl/certs/v1234567890.yourvserver.net.crt >
/etc/ssl/certs/v1234567890.yourvserver.net.pem
chmod 0755 /etc/ssl/private/
```

```
chmod 0755 /etc/ssl/certs/v1234567890.yourvserver.net.crt
chmod 0755 /etc/ssl/certs/v1234567890.yourvserver.net.pem
chmod 0755 /etc/ssl/private/v1234567890.yourvserver.net.key
```

Die erste Domain erhält ein Wildcard-Zertifikat, dieses ist für alle Subdomains gültig.

```
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/ssl/private/mydomain.de.key
-out /etc/ssl/certs/mydomain.de.crt
```

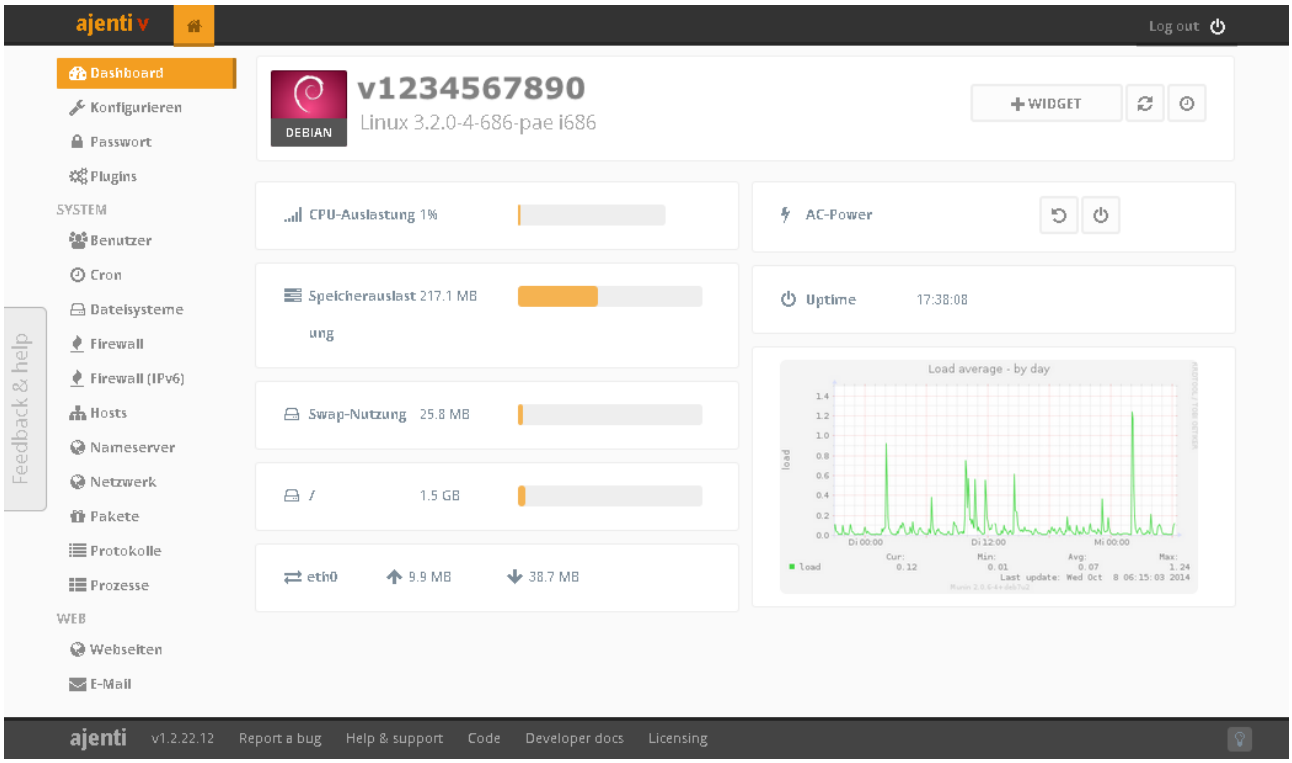
```
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:*.mydomain.de
Email Address []:
```

```
cat /etc/ssl/private/mydomain.de.key /etc/ssl/certs/mydomain.de.crt >
/etc/ssl/certs/mydomain.de.pem
chmod 0755 /etc/ssl/private/
chmod 0755 /etc/ssl/certs/mydomain.de.crt
chmod 0755 /etc/ssl/certs/mydomain.de.pem
chmod 0755 /etc/ssl/private/mydomain.de.key
```

4. Ajenti und Ajenti-V

Version: 1.2.22.16 / 0.2.52
Webseite: http://ajenti.org/

Für die webbasierte Administration des Root-Severs kommt das Open-Source-Panel Ajenti mit dem Webhosting Add-on Ajenti-V zum Einsatz.



Installation von Ajenti und Ajenti-V

Ajenti wird über die Paketquellen der Entwicklerseite installiert. Da Ajenti modular aufgebaut ist, können die benötigten Module als einzelne Pakete hinzugefügt werden. Aufgrund der Paket-Abhängigkeiten werden bei der Installation auch Nginx, PHP, MySQL, Exim 4, Courier IMAP und Pure-FTPd mitinstalliert und die zusätzlich erforderlichen PHP-Module können hier gleicht mit angegeben werden.

```
echo "deb http://repo.ajenti.org/debian main main debian" >> /etc/apt/sources.list
wget http://repo.ajenti.org/debian/key -O- | apt-key add -
apt-get update
apt-get install ajenti ajenti-v ajenti-v-nginx ajenti-v-mysql ajenti-v-php-fpm ajenti-v-mail
ajenti-v-ftp-pureftpd php5-mysqlnd php5-sqlite php5-gd php5-mcrypt php5-cli php5-curl
```

```
Neues Passwort für den MySQL->>root<<-Benutzer: ******** -> [Enter]
Wiederholen Sie das Passwort für den MySQL->>root<<-Benutzer: ******** -> [Enter]
Verzeichnisse für WWW-Administration anlegen? -> <Nein> [Enter]
SSL-Zertifikat erforderlich -> [Enter]
```

Nachdem die Grundinstallation Installation von damit Ajenti abgeschlossen wurde, sollte die Funktion getestet und ein paar wichtige Einstellungen vorgenommen werden.

```
URL: https://v1234567890.yourvserver.net:8000
Username: root
Password: admin
```

Initial-Kennwort ändern

Das initiale Kennwort ist für Bots ein gefundenes Fressen, daher sollte es so schnell wie möglich geändert werden.

```
Ajenti -> Passwort

PASSWORT ÄNDERN
Altes Passwort: ********
Neues Passwort: ********
Passwort-Bestätigung: ********

[SPEICHERN]
```

Sprache ändern und selbstsigniertes Zertifikat für Ajenti-Panel einbinden

Erfahrungsgemäß funktioniert die automatische Erkennung der Systemsprache bei Ajenti nicht korrekt, daher wird diese manuell festgelegt. An dieser Stelle wird Ajenti auch gleich angewiesen, das selbstsignierte Zertifikat für die HTTPS-Verbindung zu nutzen.

```
Ajenti -> Konfigurieren -> Allgemein

ALLGEMEIN
Sprache: de_DE

SSL
Pfad zum Zertifikat: /etc/ssl/certs/v1234567890.yourvserver.net.pem

[SPEICHERN] -> [NEUSTART]
```

Verzeichnisse für den Webserver erstellen

Bei Ajenti-V werden Webseiten unterhalb von *„/srv/“* abgelegt. Um den Webserver vom eigentlichen System besser zu isolieren, bietet es sich an, diesen Pfad auch für die Speicherung der temporären Dateien des Webservers zu verwenden. Hierzu müssen ein paar Verzeichnisse erstellt und dem User *„www-data“* zugeordnet werden.

```
mkdir /srv/tmp
mkdir /srv/sessions
mkdir /srv/data
mkdir /srv/mydomain.de/
mkdir /srv/mydomain.de/www
chown -R www-data:www-data /srv/tmp
chown -R www-data:www-data /srv/sessions
chown -R www-data:www-data /srv/www
chown -R www-data:www-data /srv/data
chown -R www-data:www-data /srv/mydomain.de/www
```

5. Nginx konfigurieren

Die folgenden Einstellungen sind optional, denn sie dienen lediglich der Optimierung von Nginx für Systeme mit nur einem Prozessorkern und wenig Hauptspeicher.

```
nano /etc/nginx/nginx.conf
```

Für die Bearbeitung von Anfragen sind unter Nginx Worker-Prozesse zuständig. Die Anzahl dieser Worker-Prozesse sollte dabei der Anzahl der CPU-Kerne entsprechen. Höhere Werte erzeugen unnötige Last.

```
worker_processes 1;
```

Ein Worker-Prozess kann mehrere Verbindungen zum Webserver bedienen. Standard-Wert ist 768, kleiner Projekte sollten aber auch mit 256 auskommen. Ob dieser Wert für das eigene System ausreicht oder ober er möglicherweise auch noch verkleinert werden kann, muss man natürlich testen.

```
worker_connections 256;
```

Standardmäßig ist gzip bereits aktiviert. Bei vielen Webseiten werden Stylesheets, Feeds und zahlreiche JavaScript-Files genutzt. Diese können mit Hilfe von gzip ebenfalls komprimiert ausgegeben werden.

```
gzip on;
gzip_disable "msie6";

gzip_vary on;
gzip_proxied any;
gzip_comp_level 6;
gzip_buffers 16 8k;
gzip_http_version 1.1;
gzip_types text/plain text/css application/json application/x-javascript text/xml
application/xml application/xml+rss text/javascript;
```

Konfigurationsdatei speichern anschließend den Nginx-Dienst neu starten.

```
service nginx restart
```

6. PHP konfigurieren

Auch wenn die Sicherheit von PHP in der letzten Zeit immer mehr in den Fokus gerückt ist und man an verschiedenen Stellen nachgebessert hat, ist es dennoch empfehlenswert, die ein oder anderen Vorgabeeinstellungen anzupassen.

FastCGI Process Manager

```
nano /etc/php5/fpm/php-fpm.conf
```

Sobald 10 PHP-FPM Kind-Prozesse innerhalb einer Minute abstürzen, wird der Dienst neugestartet. Die Kindprozesse warten dabei maximal 10 Sekunden auf den Hauptprozess.

```
emergency_restart_threshold = 10
emergency_restart_interval = 1m
process_control_timeout = 10s
```

PHP

Die in der *„php.ini“* festgelegten Einstellungen gelten global, also für alle PHP-Scripte auf dem Server. Da mit den folgenden Einstellungen aber gegebenenfalls nicht alle PHP-Anwendungen laufen, können für einzelnen Webseiten in Ajenti-V hiervon abweichende Einstellungen festgelegt werden.

```
nano /etc/php5/fpm/php.ini
```

Mit dieser Einstellung wird PHP nur den exakten Dateipfad ausführen. Ist der Wert 1 und ist die gesuchte Datei nicht vorhanden wird PHP versuchen eine Datei zu finden, die dem gesuchten Pfad am ähnlichsten ist. Dies kann bei Fehlern im Code zu einer massiven Sicherheitslücke führen.

```
cgi.fix_pathinfo=0
```

Einige Funktionen sollten definitiv nicht verfügbar sein und werden in der Regel auch nicht benötigt. Diese können daher deaktiviert werden.

```
disable_functions =
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsign
aled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_
get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exe
c,pcntl_getpriority,pcntl_setpriority,escapeshellcmd,exec,ini_restore,passthru,popen,proc_nice
,proc_open,shell_exec,system
```

Standardmäßig gibt PHP bei jeder Anfrage seine Identität (Namen und Versionsnummer) im HTTP-Header preis ("X-Powered-By"). Dies ist nicht erforderlich und sollte deaktiviert werden.

```
expose_php = Off
```

PHP-Fehlermeldungen werden im Browser angezeigt und können Angreifern die Suche nach Sicherheitslücken erleichtern. Diese sollten daher im produktiven Betrieb deaktiviert werden.

```
display_errors = Off
```

Wird der Zugriff auf Dateien außerhalb des Servers zugelassen, kann fremder Code eingeschleust werden. In den meisten Fällen ist dies nicht erforderlich und somit kann diese Option auch deaktiviert werden.

```
allow_url_fopen = Off
```

Diese Einstellung erlaubt es sogar Dateien von anderen Servern zu inkludieren. Es gibt sicher nur wenig Szenarien die des erfordern, daher ist diese Option unbedingt zu deaktivieren.

```
allow_url_include = Off
```

PHP lässt nur noch den Zugriff auf Dateien zu, die in oder unterhalb der angegebenen Pfade gespeichert sind. Damit wird unterbunden, dass auf die Systemdateien des Servers zugegriffen werden kann. Für PHP-Anwendungen die außerhalb dieses Verzeichnisses liegen (z.B. phpMyAdmin) können die entsprechenden Pfade hier aufgenommen oder später über die Webseiten-Konfiguration ergänzt werden.

```
open_basedir = /srv
```

Pfad in dem PHP seine Session-Informationen speichert. Aus Sicherheitsgründen sollte nicht das Standard-Temp-Verzeichnis verwendet werden.

```
session.save_path = /srv/sessions
```

Pfad in dem PHP hochgeladene Dateien speichert. Auch hier ist es besser nicht das Standard-Temp-Verzeichnis zu verwenden.

```
upload_tmp_dir = /srv/tmp
```

Pfad in dem PHP temporäre Dateien speichert. Und auch hier sollte das Standard-Temp-Verzeichnis nicht verwendet werden.

```
sys_temp_dir = /srv/tmp
```

Ist dieser Wert nicht definiert, werden PHP-Scripte die die Funktion *„date()“* verwenden schnell das error.log füllen, daher empfiehlt es sich Einstellung vornehmen.

```
date.timezone = Europe/Berlin
```

Die Ausführung von PHP-Scripten wird standardmäßig nach 30 Sekunden vom Parser gestoppt. Erfahrungsgemäß reicht diese Zeit bei einigen Anwendungen aber nicht aus, um ein Script ordnungsgemäß zu beenden. Daher sollte dieser Wert gegebenenfalls angepasst werden.

```
max_execution_time = 60
```

Der seit PHP 5.5.x integrierte PHP Beschleuniger Zend OPcache verwendet standardmäßig 64MB Hauptspeicher. Für diese Konfiguration sollen 32MB ausreichen.

```
opcache.memory_consumption=32
```

Konfigurationsdateien speichern anschließend PHP FPM und Nginx neu starten.

```
service php5-fpm restart
service nginx restart
```

7. MySQL konfigurieren

Da MySQL wesentlich zur Auslastung des Hauptspeichers beiträgt, sollte die Konfiguration an das System angepasst werden. Aufgrund der Vielfältigkeit von Anwendungen gibt es hierfür keine Standardvorgaben, daher muss dies in Abstimmung mit den gehosteten Anwendungen erfolgen. Folgend daher nur eine mögliche Konfiguration für Systeme mit wenig Hauptspeicher.

```
cp /etc/mysql/my.cnf /etc/mysql/my.cnf.dist
nano /etc/mysql/my.cnf
```

```
[client]
port = 3306
socket = /var/run/mysqld/mysqld.sock

[mysqld_safe]
socket = /var/run/mysqld/mysqld.sock
nice = 0

[mysqld]
# Basic Settings
user = mysql
pid-file = /var/run/mysqld/mysqld.pid
socket = /var/run/mysqld/mysqld.sock
port = 3306
basedir = /usr
datadir = /var/lib/mysql
tmpdir = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
explicit_defaults_for_timestamp
bind-address = 127.0.0.1

# Fine Tuning
key_buffer = 16K
sort_buffer_size = 64K
read_buffer_size = 256K
thread_cache_size = 8
table_open_cache = 4
max_allowed_packet = 1M
thread_stack = 64K
myisam-recover-options = BACKUP

# Query Cache Configuration
query_cache_limit = 256K
query_cache_size = 4M

# Logging and Replication
expire_logs_days = 10
max_binlog_size = 100M
log-error = /var/log/mysql/error.log

[mysqldump]
quick
quote-names
max_allowed_packet = 16M

[mysql]
no-auto-rehash

[isamchk]
key_buffer = 8M
sort_buffer_size = 8M

[myisamchk]
key_buffer = 8M
sort_buffer_size = 8M
```

```
[mysqlhotcopy]
interactive-timeout

!includedir /etc/mysql/conf.d/
```

Konfigurationsdatei speichern anschließend den MySQL-Dienst neu starten.

```
service mysql restart
```

Kommt es nach dem Neustart von MySQL zu der Fehlermeldung "*Kein Verzeichnis Anmeldung mit HOME=/*", muss das Home-Verzeichnis für den User mysql angepasst werden.

```
service mysql stop
usermod -d /var/lib/mysql mysql
service mysql start
```

Ajenti-Plugin konfigurieren

An dieser Stelle sollte auch gleich das MySQL-Plugin in Ajenti konfiguriert werden. Hier muss nur das Kennwort des MySQL-Benutzers „*root*“ angegeben werden.

```
Ajenti -> Konfigurieren -> Plugins

MySQL -> [KONFIGURIEREN]

Host: localhost
Benutzername: root
Kennwort: ********

[OK] -> [NEUSTART]
```

8. MySQL-Datenbanken verwalten

Da Ajenti bisher nur sehr eingeschränkte Möglichkeiten zur Verwaltung von MySQL-Datenbanken bietet, wird die Installation von phpMyAdmin empfohlen.

phpMyAdmin installieren

Über die offiziellen Debian-Paketquellen wird leider nur eine ältere Version installiert, da für diese aber immer noch entsprechende Security-Bugfixes verteilt werden und man in der Regel auf die neuen Features verzichten kann, soll das Distributionspaket in dieser Konfiguration ausreichen. Es bietet auch den Vorteil, dass ein Update wesentlich einfacher ist und bei Bedarf auch mittels cron-apt automatisiert werden kann.

```
apt-get install phpmyadmin
```

```
Webserver, die automatisch konfiguriert werden sollen: [ ] apache2 [ ] lighttpd -> <Ok>
[Enter]
Konfigurieren der Datenbank für phpmyadmin mit dbconfig-common? -> <Ja> [Enter]
Passwort des administrativen Datenbank-Benutzers: ******** -> [Enter]
MySQL-Anwendungspasswort für phpmyadmin: ******** -> [Enter]
Passwort-Bestätigung: ******** -> [Enter]
```

```
nano /etc/phpmyadmin/config.inc.php
```

Authentifizierungstyp auf die sicherere HTTP-Authentifizierung ändern.

```
$cfg['Servers'][$i]['auth_type'] = 'http';
```

Bei Bedarf kann man die Systemdatenbanken in phpMyAdmin ausblenden, hierzu einfach noch folgende Zeile hinzufügen.

```
$cfg['Servers'][$i]['hide_db'] = '(information_schema|mysql|performance_schema|phpmyadmin)';
```

Um phpMyAdmin nun verfügbar zu machen, muss noch eine Webseite in Ajenti erstellt werden.

```
Ajenti -> WEB -> Webseiten
Ajenti V ist noch nicht aktiviert -> [AKTIVIEREN]

NEUE WEBSEITE -> Name: phpMyAdmin -> [ERSTELLEN]
phpMyAdmin -> [VERWALTEN]

Allgemein
ALLGEMEIN -> Wartungsmodus: [ ]
WEBSEITE-DATEIEN -> Pfad: /usr/share/phpmyadmin

Domains
[HINZUFÜGEN] -> Domain: v1234567890.yourvserver.net

Ports
Host: * | Port: 8001 | SSL: [x]

SSL
SSL-Zertifikat-Pfad: /etc/ssl/certs/v1234567890.yourvserver.net.crt
SSL-Schlüssel-Pfad: /etc/ssl/private/v1234567890.yourvserver.net.key

Inhalt
PHP FastCGI - > [ERSTELLEN]
PHP -> PHP.ini-Werte:
open_basedir = /srv:/usr/share/phpmyadmin:/etc/phpmyadmin:/var/lib/phpmyadmin;

Erweitert
Benutzerdefinierte Konfiguration:
```

```
location ~ /.ht         { deny all; }
location ^~ /libraries/ { deny all; return 404; }
location ^~ /frames/    { deny all; return 404; }
location ^~ /setup/     { deny all; return 404; }
location ^~ /libs/      { deny all; return 404; }

[ÄNDERUNGEN ÜBERNEHMEN]
```

Die Installation von phpMyAdmin ist damit abgeschlossen und kann nun getestet werden.

```
URL: https://v1234567890.yourvserver.net:8001
Username: root
Password: ********
```

9. Mailserver konfigurieren

SSL/TLS-Verschlüsselung aktivieren

Um später verschlüsselt auf die Postfächer zugreifen zu können, muss SSL / TLS für den Mailserver aktiviert und das bereits erstellte Host-Zertifikat zugewiesen werden.

```
Ajenti -> WEB -> E-Mail
Ajenti-V-Mail ist noch nicht aktiviert -> [AKTIVIEREN]

Erweitert -> TLS
Aktivieren: [x]
Zertifikatspfad: /etc/ssl/certs/v1234567890.yourvserver.net.crt
Privater Schlüssel-Pfad: /etc/ssl/private/v1234567890.yourvserver.net.key

[ÄNDERUNGEN ÜBERNEHMEN]
```

Weiterleitungen für root und nobody und www-data einrichten

Viele Programme verwenden standardmäßig die Accounts root, nobody oder www-data zur eMail-Adressierung von Systemnachrichten. Für diese Accounts existierten jedoch keine Postfächer und damit entsprechende eMails auch zugestellt werden können, ist eine Weiterleitung einzurichten. Der unter Linux übliche Weg über die Datei „*/etc/aliases*“ funktioniert in Verbindung mit Ajenti nicht, daher müssen die Weiterleitungen über die eMail-Konfiguration in Ajenti eingerichtet werden.

```
Ajenti -> WEB -> E-Mail -> Allgemein

NEUES POSTFACH -> Name: root@v1234567890.yourvserver.net -> [WEITERLEITUNG]

root@v1234567890.yourvserver.net -> Zieladresse -> [HINZUFÜGEN] -> postmaster@mydomain.de

NEUES POSTFACH -> Name: nobody@v1234567890.yourvserver.net -> [WEITERLEITUNG]

nobody@v1234567890.yourvserver.net -> Zieladresse -> [HINZUFÜGEN] -> postmaster@mydomain.de

NEUES POSTFACH -> Name: www-data@v1234567890.yourvserver.net -> [WEITERLEITUNG]

www-data@v1234567890.yourvserver.net -> Zieladresse -> [HINZUFÜGEN] -> postmaster@mydomain.de

[ÄNDERUNGEN ÜBERNEHMEN]
```

Hinweis: Die Zieladresse „postmaster@mydomain.de“ wird erst im späteren Verlauf dieser Anleitung erstellt.

POP3 installieren

Da es von Ajenti nicht direkt unterstützt wird, muss POP3 / POP3S (insofern benötigt) manuell installiert werden.

```
apt-get install courier-pop courier-pop-ssl
```

Für einen SSL-verschlüsselten Zugriff auf POP3 muss ein Zertifikat in der Konfigurationsdatei für POP3S angegeben werden. Auch hier wird wieder das bereits erstellte Host-Zertifikat verwendet.

```
nano /etc/courier/pop3d-ssl
```

```
TLS_CERTFILE=/etc/ssl/certs/v1234567890.yourvserver.net.pem
```

Wenn die Konfigurationsdatei gespeichert und der Dienst neu gestartet wurde, sollte nun auch der Mailabruf über POP3 und POP3S funktionieren.

```
service courier-pop-ssl restart
```

10. FTP-Server konfigurieren

SSL/TLS aktivieren

Auch wenn es immer noch weit verbreitet ist, wird dringend davon abgeraten unverschlüsselt auf einen nichtöffentlichen FTP-Server zuzugreifen, denn auf diesem Weg werden die Zugangsdaten im Klartext übertragen und können so schnell abgefangen werden. Da Ajenti bisher noch keine SSL/TLS Konfiguration für Pure-FTPd unterstützt, muss dies manuell vorgenommen werden.

Zuerst muss SSL/TLS in Pure-FTPd aktiviert werden.

```
echo 1 > /etc/pure-ftpd/conf/TLS
```

Da Pure-FTPd einen vorgegebenen Pfad für das Zertifikat verlangt, kann an dieser Stelle leider nicht das bereits vorhandene Host-Zertifikat verwendet werden. Daher wird hiervon eine Kopie erzeug und unter dem geforderten Pfad abgelegt.

```
cp /etc/ssl/certs/v1234567890.yourvserver.net.pem /etc/ssl/private/pure-ftpd.pem
```

Nach einem Neustart des Dienstes sollte nun auch der verschlüsselte Zugriff funktionieren.

```
service pure-ftpd restart
```

Ports für passiven FTP festlegen

Die meisten Clients, die auf einen FTP-Server zugreifen befinden sich heute hinter einer Firewall oder verwenden NAT für den externen Zugriff. Der FTP-Server ist also nicht in der Lage selbst eine Verbindung zum Client aufzubauen, daher wird vorwiegend der passive Modus verwendet. In diesem wird für jede gleichzeitige Client-Verbindung ein zusätzlicher Port oberhalb von 1023 erforderlich. Diese Ports sollten fest vorgegeben werden, um sie so später in Firewall freigeben zu verwenden zu können.

Hierzu muss eine entsprechende Konfigurationsdatei für Pure-FTPd erstellt werden.

```
nano /etc/pure-ftpd/conf/PassivePortRange
```

In diesem Beispiel stehen also Ports für 50 gleichzeitige Verbindungen zur Verfügung.

```
50001 50050
```

Nachdem die Datei erstellt wurde, ist ein Neustart des Dienstes erforderlich.

```
service pure-ftpd restart
```

11. Automatische Software- und Sicherheitsupdates

Das Programm cron-apt prüft jeden Tag um 04:00 Uhr die vorhandenen Paketinformationen, aktualisiert die Paketliste, lädt gegebenenfalls neue Pakete und versendet eine Info-Mail. Bei Bedarf können die neuen Pakete auch direkt mittels cron-apt installiert werden - dies ist natürlich mit Vorsicht zu genießen.

cron-apt installieren

Die Installation des cron-apt Paketes ist schnell erledigt. Der Empfänger der Mail-Benachrichtigung und der Benachrichtigungstyp (*upgrade, error, always*) kann in der Konfigurationsdatei geändert werden.

```
apt-get install cron-apt
nano /etc/cron-apt/config
```

```
# Configuration for cron-apt. For further information about the possible
# configuration settings see /usr/share/doc/cron-apt/README.gz.
APTCOMMAND=/usr/bin/apt-get
MAILTO="root"
MAILON="upgrade"
```

Mail-Benachrichtigung aktivieren

Um die eMail-Benachrichtigung bei neunen Updates zu aktivieren, muss lediglich eine Konfigurationsdatei kopiert werden.

```
cp /usr/share/doc/cron-apt/examples/9-notify /etc/cron-apt/action.d/
```

Abschließend kann die Funktion von cron-apt getestet werden.

```
cron-apt -s
```

Updates automatisch installieren

Sollen neue Pakete nicht nur heruntergeladen, sondern auch gleich automatisch installiert werden, muss der Parameter „*-d*“ in der folgenden Datei entfernt werden.

```
nano /etc/cron-apt/action.d/3-download
```

Der Inhalt der Datei sollte dann so aussehen.

```
autoclean -y
dist-upgrade -y -o APT::Get::Show-Upgraded=true
```

12. Fail2ban - Intrusion Prevention System Framework installieren

Fail2ban überwacht Logfiles, erkennt bösartige Zugriffe und sperrt IP-Adressen für einen bestimmten Zeitraum.

Fail2ban installieren

Die Implementierung von Fail2ban ist nicht ganz einfach, denn es sollte optimal an das System und die darauf laufenden Dienste angepasst werden. Die Standard-Installation von Fail2ban beinhaltet bereits Filter für die gängigsten Dienste, leider gehört Nginx noch nicht dazu, so dass hierfür und auch für Ajenti entsprechende Filter erstellt werden müssen. Die hier aufgeführte Fail2ban-Konfiguration sollte auf jeden Fall auf dem eigenen System getestet werden bevor dieses produktiv genutzt wird.

```
apt-get install fail2ban
```

Filter fail2ban.conf erstellen

Mit diesem Filter überwacht Fail2ban sein eigenes Logfile um Systeme zu identifizieren, die schon mehrfach gesperrt wurden. Dies ist erforderlich, um besonders penetrante Angreifer für einen länger Zeit zu sperren.

```
nano /etc/fail2ban/filter.d/fail2ban.conf
```

```
# Fail2Ban configuration file
#
# Block persistent attacks
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = fail2ban.actions: WARNING \[(.*)\] Ban <HOST>

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex = fail2ban.actions: WARNING \[fail2ban\] Ban <HOST>
```

Filter ajenti-login.conf erstellen

```
nano /etc/fail2ban/filter.d/ajenti-login.conf
```

```
# Fail2Ban configuration file
#
# Block failed Ajenti-Login (only password)
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = failed login .* from <HOST>

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
```

```
#
ignoreregex =
```

Filter nginx-http-auth.conf erstellen

```
nano /etc/fail2ban/filter.d/nginx-http-auth.conf
```

```
# Fail2Ban configuration file
#
# Block failed Nginx www-authentication
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = ^<HOST> - .* "(GET|POST|HEAD).*HTTP/.*" 401 [0-9]{1,}

# Option:  ignoreregex
# Notes :  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#

ignoreregex =
```

Filter nginx-scan.conf erstellen

```
nano /etc/fail2ban/filter.d/nginx-scan.conf
```

```
# Fail2Ban configuration file
#
# Block scan of no existing content
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = ^<HOST> - .* "(GET|POST|HEAD).*HTTP/.*" (400|403|404|405|406|444) [0-9]{1,}

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Filter nginx-noscript.conf erstellen

```
nano /etc/fail2ban/filter.d/nginx-noscript.conf
```

```
# Fail2Ban configuration file
#
# Block trying to execute scripts
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
```

```
#
failregex = Unable to open primary script: .*, client: <HOST>

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Filter nginx-proxy.conf erstellen

```
nano /etc/fail2ban/filter.d/nginx-proxy.conf
```

```
# Fail2Ban configuration file
#
# Block trying to use server as proxy.
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = ^<HOST> -.*GET http.*

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Filter nginx-login.conf erstellen

```
nano /etc/fail2ban/filter.d/nginx-login.conf
```

```
# Fail2Ban configuration file
#
# Blocks that fail to authenticate using web application's log in page
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = ^<HOST> -.*POST /sessions HTTP/1\.." 200

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Filter nginx-dos.conf erstellen

```
nano /etc/fail2ban/filter.d/nginx-dos.conf
```

```
# Fail2Ban configuration file
#
# Block DDOS
#
```

```
[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = ^<HOST> -.*"(GET|POST).*HTTP.*"$

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Filter exim-auth.conf erstellen

```
nano /etc/fail2ban/filter.d/exim-login.conf
```

```
# Fail2Ban configuration file
#
# Block failed Exim-Login
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = login authenticator failed for (\S+ )?\(\S+\) \[<HOST>\]: 435 Unable to
authenticate at present

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Modul-Optionen festlegen

Nach dem die zusätzlichen Filter erstellt wurden, müssen diese und auch verschiedene Standardfilter in der Konfigurationsdatei von Fail2ban aktiviert und konfiguriert werden.

```
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
nano /etc/fail2ban/jail.local
```

```
[DEFAULT]

ignoreip = 127.0.0.1/8
bantime  = 3600
maxretry = 3

backend = auto
destemail = root

banaction = iptables-multiport
mta = sendmail
protocol = tcp
chain = INPUT

action_ = %(banaction)s[name=%(__name__)s, port="%(port)s", protocol="%(protocol)s",
chain="%(chain)s"]

action_mw = %(banaction)s[name=%(__name__)s, port="%(port)s", protocol="%(protocol)s",
chain="%(chain)s"]
```

```
             %(mta)s-whois[name=%(__name__)s, dest="%(destemail)s", protocol="%(protocol)s",
chain="%(chain)s"]

action_mwl = %(banaction)s[name=%(__name__)s, port="%(port)s", protocol="%(protocol)s",
chain="%(chain)s"]
              %(mta)s-whois-lines[name=%(__name__)s, dest="%(destemail)s",
logpath=%(logpath)s, chain="%(chain)s"]

action = %(action_mwl)s

[fail2ban]
enabled   = true
filter    = fail2ban
action    = iptables-allports[name=fail2ban]
            sendmail-whois[name=fail2ban]
logpath   = /var/log/fail2ban.log
findtime  = 604800
maxretry  = 3
bantime   = 604800

[ajenti-login]
enabled   = true
port      = 8000
filter    = ajenti-login
logpath   = /var/log/ajenti/ajenti.log*
maxretry  = 3

[nginx-http-auth]
enabled   = true
port      = http,https,8001
filter    = nginx-http-auth
logpath   = /var/log/nginx*/*access*.log
maxretry  = 3

[nginx-scan]
enabled   = true
port      = http,https
filter    = nginx-scan
logpath   = /var/log/nginx*/*access*.log
findtime  = 60
maxretry  = 6

[nginx-noscript]
enabled   = true
port      = http,https
filter    = nginx-noscript
logpath   = /var/log/nginx*/*error*.log
maxretry  = 3

[nginx-proxy]
enabled   = true
port      = http,https
filter    = nginx-proxy
logpath   = /var/log/nginx*/*access*.log
maxretry  = 0

[nginx-login]
enabled   = true
port      = http,https
filter    = nginx-login
logpath   = /var/log/nginx*/*access*.log
maxretry  = 3

[nginx-dos]
enabled   = true
port      = http,https
filter    = nginx-dos
logpath   = /var/log/nginx*/*access*.log
```

```
findtime  = 60
maxretry  = 300

[xinetd-fail]
enabled   = false
filter    = xinetd-fail
port      = all
banaction = iptables-multiport-log
logpath   = /var/log/daemon.log
maxretry  = 2

[exim-login]
enabled   = true
port      = smtp,ssmtp
filter    = exim-login
logpath   = /var/log/exim*/rejectlog
maxretry  = 1

[ssh]
enabled   = true
port      = ssh,8002
filter    = sshd
logpath   = /var/log/auth.log
maxretry  = 3

[ssh-ddos]
enabled   = true
port      = ssh,8002
filter    = sshd-ddos
logpath   = /var/log/auth.log
maxretry  = 3

[pure-ftpd]
enabled   = true
port      = ftp,ftp-data,ftps,ftps-data
filter    = pure-ftpd
logpath   = /var/log/syslog
maxretry  = 3

[exim]
enabled   = true
port      = smtp,ssmtp
filter    = exim
logpath   = /var/log/exim*/rejectlog
maxretry  = 1

[courierauth]
enabled   = true
port      = smtp,ssmtp,imap2,imap3,imaps,pop3,pop3s
filter    = courierlogin
logpath   = /var/log/mail.log
maxretry  = 3
```

Konfigurationsdateien speichern anschließend Fail2ban neu starten.

```
service fail2ban restart
```

Mit folgender Befehlszeile kann die Funktion der einzelnen Filter (z.B. für ssh) getestet werden.

```
fail2ban-regex /var/log/auth.log /etc/fail2ban/filter.d/sshd.conf
```

Start-/Stop-Benachrichtigungen deaktivieren

Standardmäßig versendet Fail2ban auch eMails, wenn Jails gestartet oder gestoppt wurden. Bei einem Neustart des Servers kommen so schnell mehr als 20 eMails zusammen. Wen das stört, der kann diese Meldungen deaktivieren, dazu müssen die Parameter „*actionstart*“ und „*actionstop*“ in den entsprechenden Konfigurationsdateien geleert werden. Die ist mit folgender Befehlszeile schnell erledigt.

```
sed -i s/"actionunban ="/"actionunban =\nactionstart =\nactionstop ="/g
/etc/fail2ban/action.d/*mail*.conf
```

13. Absicherung des Servers mit iptables

Zum Schutz vor ungewollten Netzwerkzugriffen bietet Ajenti unter dem Punkt „*Firewall*“ eine komfortable Konfigurationsoberfläche für iptables. Da die Erstellung eines komplett neuen Regelwerks über diese Oberfläche jedoch sehr mühselig werden kann und sich zudem so schnell Fehler einschleichen, kann das initiale Regelwerk auch als Script hinterlegt und anschließend geladen werden.

Grundkonfiguration

Die Konfiguration der Firewall muss natürlich immer auf das jeweilige System und dessen Umgebung abgestimmt werden. Daher soll folgendes Script nur als Grundlage für eine eigene Konfiguration dienen.

```
nano /etc/iptables.up.rules.ajenti
```

```
*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT

*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT

*filter
:INPUT DROP [0:0]
:OUTPUT DROP [0:0]
:FORWARD DROP [0:0]

#####################################################################
### Incoming Traffic
#####################################################################

-A INPUT -m state --state INVALID -j DROP # Invalid Packet
-A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP # Portscan - SYN + RST
-A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP # Portscan - SYN + FIN
-A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP # Portscan - FIN + URG + PSH
-A INPUT -p tcp --tcp-flags ALL ALL -j DROP # Portscan - ALL Flags
-A INPUT -p tcp --tcp-flags ALL NONE -j DROP # Portscan - nmap Null scan
-A INPUT -p tcp --tcp-flags ALL FIN -j DROP # Portscan - nmap FIN stealth scan
-A INPUT -p tcp --tcp-flags ALL URG,ACK,PSH,RST,SYN,FIN -j DROP # Portscan - XMAS

-A INPUT -i lo -j ACCEPT # Loopback (localhost)
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT # Established Sessions
-A INPUT -p icmp --icmp-type echo-request -j ACCEPT # ICMP - Echo-Request (8)
-A INPUT -i eth0 -p udp --dport 67:68 --sport 67:68 -j ACCEPT # DCHP - Client

-A INPUT -p tcp --dport 8002 -m state --state NEW -j ACCEPT # SSH - OpenSSH
-A INPUT -p tcp --dport 8000 -m state --state NEW -j ACCEPT # Ajenti - Webpanel
-A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT # HTTP - nginx
-A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT # HTTPS - nginx
-A INPUT -p tcp --dport 8001 -m state --state NEW -j ACCEPT # HTTPS - nginx (phpMyAdmin)
-A INPUT -p tcp -m multiport --dport 21,50001:50050 -m state --state NEW -j ACCEPT # FTP -
Pure-FTP
-A INPUT -p tcp --dport 25 -m state --state NEW -j ACCEPT # SMTP - Exim
-A INPUT -p tcp --dport 465 -m state --state NEW -j ACCEPT # SMTPS - Exim
-A INPUT -p tcp --dport 110 -m state --state NEW -j ACCEPT # POP3 - Courier
-A INPUT -p tcp --dport 995 -m state --state NEW -j ACCEPT # POP3S - Courier
-A INPUT -p tcp --dport 143 -m state --state NEW -j ACCEPT # IMAP - Courier
```

```
-A INPUT -p tcp --dport 993 -m state --state NEW -j ACCEPT # IMAPS - Courier

-A INPUT -j DROP # Any


######################################################################
### Outgoing Traffic
######################################################################

-A OUTPUT -m state --state INVALID -j DROP # Invalid Packet

-A OUTPUT -o lo -j ACCEPT # Loopback (localhost)
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT # Established Sessions
-A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT # ICMP - Echo-Request (8)
-A OUTPUT -o eth0 -p udp --dport 67:68 --sport 67:68 -j ACCEPT # DCHP - Client

-A OUTPUT -p tcp --dport 53 -m state --state NEW -j ACCEPT # DNS - System
-A OUTPUT -p udp --dport 53 -m state --state NEW -j ACCEPT # DNS - System
-A OUTPUT -p udp --dport 123 -m state --state NEW -j ACCEPT # NTP - System
-A OUTPUT -p tcp --dport 43 -m state --state NEW -j ACCEPT # WHOIS - System
-A OUTPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT # HTTP - nginx
-A OUTPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT # HTTPS - nginx
-A OUTPUT -p tcp --dport 21 -m state --state NEW -j ACCEPT # FTP - Pure-FTP
-A OUTPUT -p tcp --dport 25 -m state --state NEW -j ACCEPT # SMTP - Exim
-A OUTPUT -p tcp --dport 465 -m state --state NEW -j ACCEPT # SMTPS - Exim

-A OUTPUT -j DROP # Any


######################################################################
### Forward Traffic
######################################################################

-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT # Established Sessions

-A FORWARD -m state --state INVALID -j DROP # Invalid Packet

-A FORWARD -j DROP # Any

COMMIT
```

Nachdem das Script angepasst bzw. erstellt wurde, muss es in Ajenti noch aktiviert werden. Dabei sollte man den Autostart nicht vergessen, denn sonst werden die Regeln bei einem Neustart des Servers nicht automatisch geladen.

```
Ajenti -> SYSTEM -> Firewall -> filter

[ANWENDEN] -> [AKTIVIREN AUTOSTART]
```

14. Spam- und Virenschutz

Wie schon bei der Firewall muss auch das Thema Spam- und Virenschutz individuell betrachtet werden. Leider bietet Ajenti bisher noch keine vollwertige Implementierung von entsprechenden Schutzmaßnahmen, doch die Entwickler haben bereits Erweiterungen in diese Richtung angekündigt. Da aber noch nicht bekannt ist, um was sich dabei handeln wird, beschränkt sich der folgende Abschnitt nur auf die bereits verfügbaren Schutzfunktionen.

SPF - Sender Policy Framework

Mittels SPF soll das Fälschen von Absenderadressen bei eMails verhindert werden. Hierzu wird ein spezieller DNS-Eintrag mit dem zum Versand von eMails autorisierten Mailservern erstellt. Empfangende Mailserver können so schnell prüfen, ob eine eMail berechtigt ist oder ob die Absenderadresse gefälscht wurde.

Folgender Eintrag berechtigt alle Server, für die ein MX-Eintrag existiert.

```
@     IN TXT "v=spf1 mx -all"
```

Am Beispiel von netcup wird dieser wie folgt angelegt.

```
netcup ccp -> Domains -> mydomain.de -> DNS
Neue Zeile ->
Host: @
Type: TXT
Destination: v=spf1 mx –all

[Speichern]
```

ACHTUNG! Sollen eMails mit dem Server weitergeleitet werden oder kommen Anwendungen zum Einsatz, die eMails mit geänderter Absenderadresse (z.B. Absenderadresse = Empfängeradresse) versenden, kann dies zu Problemen führen. In diesem Fall wird daher folgender Eintrag empfohlen.

```
@     IN TXT "v=spf1 mx ?all"
```

DKIM - DomainKeys Identified Mail

Auf Basis eines privaten Schlüssels versieht DKIM eMails mit einer digitalen Signatur, die der empfangende Mailserver anhand des öffentlichen Schlüssels, welcher im DNS hinterlegt ist, verifizieren kann. So soll die Authentizität von eMail-Absendern sichergestellt werden.

Für die Konfiguration von DKIM bietet Ajenti-V einen Assistenten, mit dem die erforderlichen Schlüssel erstellt werden und Exim4 konfiguriert wird.

```
Ajenti -> WEB -> E-Mail

DKIM
Aktivieren: [x]
Selektor: mail

[NEUEN DKIM-SCHLÜSSEL ERZEUGEN] -> [ÄNDERUNGEN ÜBERNEHMEN]
```

Im Feld *„DNS Eintrag Vorlage“* sollte nun die notwendigen DNS-Einträge angezeigt werden. Für DKIM werden aber nur die Einträge aus Zeile 2 und 3 benötigt. Hier ein Beispiel.

```
_domainkey          10800 IN TXT "o=~; r=postmaster@<domain>"
```

```
mail._domainkey     10800 IN TXT "v=DKIM1; k=rsa;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0q4X0SYzHnfc06bYcJfHRwmwngGv3Pz48KIw06tnfUPihqQK
ZjH5fcPRdXci87+0D5Ngxliuqe/R7sSshUYQAxk015BHXcMAOuBAbVlCBBaitOi3y6XWjiEIz9DxT+X1Xwo+cBo+AJMQZh
JJaQEJzDfq+6tEUVZSsNvXIhpCpuc6UN15/7l3pUrilgo19+3KrBelPPn8G84y26Fo9F2y/f1xtdp9meZpIWyAHshbRO2l
s5u7UuHqbZOuVZCMKa5dP1lyGivtByS6/CSVnmRb6M8B84T1s1BXrbsVFhI4XRohzzZueBMX9N1WyU/3qa+n9ZzGTLK+7z
7ZhaqDE5IBgwIDAQAB"
```

Wie auch schon für SPF, müssen auch diese Einträge im DNS hinterlegt werden, dabei ist zu berücksichtigen, dass die eMail-Adresse „*postmaster@<domain>*“ entsprechend anzupassen ist. An diese Adresse sollen eventuelle Fehler beim der DomainKey-Überprüfung gemeldet werden. Hier wieder am Beispiel von netcup.

```
netcup ccp -> Domains -> mydomain.de -> DNS
Neue Zeile ->
Host: _domainkey
Type: TXT
Destination: o=~; r=postmaster@mydomain.de

Neue Zeile ->
Host: mail._domainkey
Type: TXT
Destination: v=DKIM1; k=rsa;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0q4X0SYzHnfc06bYcJfHRwmwngGv3Pz48KIw06tnfUPihqQK
ZjH5fcPRdXci87+0D5Ngxliuqe/R7sSshUYQAxk015BHXcMAOuBAbVlCBBaitOi3y6XWjiEIz9DxT+X1Xwo+cBo+AJMQZh
JJaQEJzDfq+6tEUVZSsNvXIhpCpuc6UN15/7l3pUrilgo19+3KrBelPPn8G84y26Fo9F2y/f1xtdp9meZpIWyAHshbRO2l
s5u7UuHqbZOuVZCMKa5dP1lyGivtByS6/CSVnmRb6M8B84T1s1BXrbsVFhI4XRohzzZueBMX9N1WyU/3qa+n9ZzGTLK+7z
7ZhaqDE5IBgwIDAQAB

[Speichern]
```

DMARC - Domain-based Message Authentication, Reporting and Conformance

Bei DMARC handelt es sich nicht um einen eigenständigen Prozess, sondern um eine Erweiterung von SPF und DKIM. Mit DMARC wird festgelegt, wie mit eMails zu verfahren ist, wenn sie die Prüfung mittels SPF und DKIM nicht bestanden haben.

Mit folgender Regel wird festgelegt, dass eMails, die die SPF- oder DKIM-Prüfung nicht bestanden haben als Spam markiert werden.

```
_dmarc                      10800 IN TXT "v=DMARC1; p=quarantine"
```

```
netcup ccp -> Domains -> mydomain.de -> DNS
Neue Zeile ->
Host: _dmarc
Type: TXT
Destination: v=DMARC1; p=quarantine
```

Greylisting

Ajenti bietet zwar keine Unterstützung für Greylisting, doch da es eine wirklich sehr effektive Möglichkeit der Spam-Bekämpfung ist (leider wird es zu Unrecht oft schlecht geredet) und die Installation und Konfiguration schnell erledigt ist, soll es hier dennoch beschrieben werden.

Beim Einsatz von Greylisting werden eMails von unbekannten Absendern zunächst vom SMTP-Dienst abgewiesen und erst beim nächsten Zustellversuch angenommen. Die Zeit bis zum nächsten Zustellversuch wird dabei vom sendenden Mailserver vorgegeben. Wurde die eMail dann angenommen, werden eMails vom gleichen Server künftig sofort zugestellt. Auch wenn diese Form der Spam-Bekämpfung sehr effektiv ist, sollte berücksichtigt werden, dass eMails von unbekannten Absender verzögert zugestellt werden und

es ist zwar eine große Ausnahme, aber es gibt noch Maildienste, die lediglich einen Zustellversuch unternehmen.

```
apt-get install greylistd
```

Nachdem der Greylisting-Daemon installiert wurde, muss dieser noch in die Exim4-Konfiguration eingebunden werden. Der übliche Weg über den Aufruf von *„greylistd-setup-exim4 “* funktioniert unter Ajenti nicht, daher muss die Konfigurationsdatei manuell angepasst werden. Dabei wird auch festgelegt, dass nur die Klasse-C-Netzwerkadresse vom absendenden Mailserver gespeichert wird. Dies ist erforderlich, da viele Provider mehrere Mailserver betreiben (z.B. für Lastverteilung) und es vorkommen kann, dass der zweite Zustellversuch von einem anderen Mailserver erfolgt und dieser dann auch abgewiesen würde.

```
nano /etc/exim4/exim4.config
```

```
#--CONFIGURATION

hostlist relay_from_hosts = 127.0.0.1
domainlist relay_to_domains =
```

```
begin acl

acl_local_deny_exceptions:
  accept
    hosts = ${if exists{/etc/exim4/host_local_deny_exceptions}\
                {/etc/exim4/host_local_deny_exceptions}\
                {}}
  accept
    senders = ${if exists{/etc/exim4/sender_local_deny_exceptions}\
                  {/etc/exim4/sender_local_deny_exceptions}\
                  {}}
```

```
acl_check_rcpt:
 # greylistd  begin
  defer
    message        = $sender_host_address is not yet authorized to deliver \
                     mail from <$sender_address> to <$local_part@$domain>. \
                     Please try later.
    log_message    = greylisted.
    !senders       = :
    !hosts         = : +relay_from_hosts : \
                     ${if exists {/etc/greylistd/whitelist-hosts}\
                                 {/etc/greylistd/whitelist-hosts}{}} : \
                     ${if exists {/var/lib/greylistd/whitelist-hosts}\
                                 {/var/lib/greylistd/whitelist-hosts}{}}
    !authenticated = *
    !acl           = acl_local_deny_exceptions
    domains        = +local_domains : +relay_to_domains
    verify         = recipient
    condition      = ${readsocket{/var/run/greylistd/socket}\
                                 {--grey \
                                  ${mask:$sender_host_address/24} \
                                  $sender_address \
                                  $local_part@$domain}\
                                 {5s}{}{false}}

 # Deny if blacklisted by greylist
 deny
   message = $sender_host_address is blacklisted from delivering \
                    mail from <$sender_address> to <$local_part@$domain>.
   log_message = blacklisted.
   !senders        = :
   !authenticated = *
   domains        = +local_domains : +relay_to_domains
   verify         = recipient
```

```
   condition      = ${readsocket{/var/run/greylistd/socket}\
                                {--black \
                                 $sender_host_address \
                                 $sender_address \
                                 $local_part@$domain}\
                                {5s}{}{false}}
 # greylistd end
```

```
acl_check_data:
 # greylistd  begin
  defer
    message        = $sender_host_address is not yet authorized to deliver \
                     mail from <$sender_address> to <$recipients>. \
                     Please try later.
    log_message    = greylisted.
    senders        = :
    !hosts         = : +relay_from_hosts : \
                     ${if exists {/etc/greylistd/whitelist-hosts}\
                                 {/etc/greylistd/whitelist-hosts}{}} : \
                     ${if exists {/var/lib/greylistd/whitelist-hosts}\
                                 {/var/lib/greylistd/whitelist-hosts}{}}
    !authenticated = *
    !acl           = acl_local_deny_exceptions
    condition      = ${readsocket{/var/run/greylistd/socket}\
                                 {--grey \
                                  ${mask:$sender_host_address/24} \
                                  $recipients}\
                                  {5s}{}{false}}

 # Deny if blacklisted by greylist
 deny
   message = $sender_host_address is blacklisted from delivering \
                    mail from <$sender_address> to <$recipients>.
   log_message = blacklisted.
   !senders        = :
   !authenticated = *
   condition      = ${readsocket{/var/run/greylistd/socket}\
                                {--black \
                                 $sender_host_address \
                                 $recipients}\
                                 {5s}{}{false}}
 # greylistd end
```

Konfigurationsdateien speichern anschließend Exim4 neu starten.

```
service exim4 restart
```

Wie lange der sendende Mailserver nach dem ersten Zustellversuch warten muss, kann in der Konfigurationsdatei von greylistd festgelegt werden.

```
nano /etc/greylistd/config
```

```
retryMin     = 600
```

Sollen Server von der temporären Sperre ausgenommen werden, kann man diese in eine Whitelist-Datei eingetragen.

```
nano /etc/greylistd/whitelist-hosts
```

Zudem bringt greylistd unter *„/var/lib/greylistd/whitelist-hosts“* eine eigene Whitelist mit, die bereits einige Ausnahmen enthält. Diese sollte aber nicht für eigene Einträge verwendet werden, da die Datei bei einem Update eventuell überschrieben wird.

15. System-Monitoring mit Munin

Wer einen Server betreibt will natürlich auch wissen, wie dieser ausgelastet ist und wo eventuell Optimierungsbedarf besteht. Linux selbst bietet hier zwar einige Werkzeuge, aber deutlich komfortabler geht es mit dem Programm Munin. Zudem wird es auch von Ajenti unterstützt und integriert sich sehr schön in dessen Oberfläche.

In der Standardkonfiguration liest Munin alle 5 Minuten verschiedene Systeminformationen aus und erstellt daraus statische HTML-Seiten mit einer Visualisierung der entsprechenden Leistungsdaten. Da Munin keinen eigenen Webservice mitbringt, müssen die generierten HTML-Seiten auf einem bereits vorhandenen Webserver eingebunden werden.

Installation von Munin

Die Installation von Munin erfolgt über die Debian-Paketquellen und für die spätere Integration in Ajenti wird noch ein Python-HTML-Parser benötigt.

```
apt-get install munin python-beautifulsoup
```

Damit Munin funktioniert, müssen verschiedene Einträge in der Konfigurationsdatei angepasst werden.

```
cp /etc/munin/munin.conf /etc/munin/munin.conf.dist
nano /etc/munin/munin.conf
```

Hier wird unter anderem festgelegt, wo Munin die erzeugten HTML-Seiten speichert und welche Systeme ausgewertet werden sollen - in diesem Fall natürlich der lokale Server.

```
dbdir   /var/lib/munin
htmldir /var/cache/munin/www
logdir /var/log/munin
rundir  /var/run/munin

tmpldir /etc/munin/templates

includedir /etc/munin/munin-conf.d

[v1234567890.yourvserver.net]
    address 127.0.0.1
    use_node_name yes
```

Sobald die Konfiguration gespeichert und der Dienst neu gestartet ist, wird Munin mit dem Monitoring beginnen.

```
service munin-node restart
```

Für die Integration in Ajenti ist es erforderlich eine Webseite zu erstellen, die auf den Ausgabepfad von Munin verweist. Da die Reports von Munin später aber ausschließlich in Ajenti abrufbar sein sollen, wird die Seite so konfiguriert, dass sie nur für den Server selbst verfügbar ist. Auf SSL kann bei dieser Seite daher auch verzichtet werden.

```
Ajenti -> WEB -> Webseiten

NEUE WEBSEITE -> Name: Munin -> [ERSTELLEN]
Munin -> [VERWALTEN]

Allgemein
ALLGEMEIN -> Wartungsmodus: [ ]
WEBSEITE-DATEIEN -> Pfad: /var/cache/munin/www
```

```
Domains
[HINZUFÜGEN] -> Domain: localhost

Erweitert
Benutzerdefinierte Konfiguration:
location / {
  allow   127.0.0.1;
  deny    all;
}

[ÄNDERUNGEN ÜBERNEHMEN]
```

Der Server sollte nun neu gestartet werden.

```
reboot
```

In Ajenti muss abschließend nur noch die URL der soeben erstellten Seite in der Munin-Konfiguration hinterlegt werden. Die in der Eingabemaske abgefragten Zugangsdaten sind nicht erforderlich und können ignoriert werden.

```
Ajenti -> SOFTWARE -> Munin

Konfigurieren
Base URL: http://localhost

[SAVE]
```

Sobald der erste Report erzeugt wurde, kann er wie folgt aufgerufen werden. Die einzelnen Diagramme können von hier auch als Widget zum Dashboard hinzugefügt werden.

```
Ajenti -> SOFTWARE -> Munin -> Ansehen -> v1234567890.yourvserver.net
```

16. Auswertung von Logfiles mit Logwatch

Bei der Vielzahl an Logdaten die auf einem Server anfallen verliert man schnell den Überblick und eine regelmäßige manuelle Überprüfung dieser Daten ist in annehmbarer Zeit kaum möglich. Eine Lösung für dieses Problem bietet Logwatch, denn es automatisiert die Erstellung von Übersichten auf Basis der bestehenden Logdaten und versendet diese an eine festgelegte eMail-Adresse.

Installation von Logwatch

Auch Logwatch ist als Debian-Paket verfügbar und somit schnell installiert.

```
apt-get install logwatch
```

Hier wird die Standardkonfiguration von Logwatch verwendet. Bei Bedarf kann diese aber wie folgt angepasst werden.

```
nano /usr/share/logwatch/default.conf/logwatch.conf
```

Über einen Cronjob wird Logwatch gestartet, dabei werden auch die Parameter für die Ausgabe von Logwatch übergeben. In diesem Beispiel wird eine Zusammenfassung der Logdaten vom vorherigen Tag mit einer geringen Detailtiefe erzeugt und als einfacher Text per Mail an den Account „*root*“ versendet.

```
nano /etc/cron.daily/00logwatch
```

```
/usr/sbin/logwatch --output mail --format text --detail low --range yesterday
```

Die Ausgabe kann auch im HTML-Format erfolgen (*--output html*). Diese kann, wenn erforderlich, über ein Template angepasst werden.

```
nano /usr/share/logwatch/default.conf/html/header.html
nano /usr/share/logwatch/default.conf/html/footer.html
```

17. Benachrichtigungen

Neben den Benachrichtigungen, die vom System selbst oder von den installierten Anwendungen erzeugt werden, können bei Bedarf auch eigene eMail-Benachrichtigungen eingerichtet werden.

Login-Benachrichtigung

Sehr nützlich ist zum Beispiel eine zeitnahe Information, sobald sich auf dem Root-Server ein Benutzer an der Konsole bzw. per SSH angemeldet hat. Der Inhalt der entsprechenden eMail wird mit folgendem Script festgelegt.

```
nano /usr/local/bin/mailon-login.sh
```

```
#!/bin/bash
REMOTEIP=`echo $SSH_CONNECTION | awk '{ print $1 }'`
echo "------------------------------------------------"
echo "Login detected"
echo "------------------------------------------------"
echo "Hostname: $(hostname) "
echo "Date: $(date +%Y-%m-%d)"
echo "Time: $(date +%H:%M)"
echo "User: $USER"
echo "Remote IP: $REMOTEIP"
echo
echo "------------------------------------------------"
echo "Last Activity"
echo "------------------------------------------------"
last
```

Da das Script beim jedem Login eines Benutzers ausgeführt werden soll, muss es auch für alle Benutzer ausführbar gemacht werden.

```
chmod 755 /usr/local/bin/mailon-login.sh
```

Der Aufruf erfolgt über folgende Datei.

```
nano /etc/profile
```

In dieser muss nachfolgende Zeile eingefügt werden, welche bei jedem Login ausgeführt wird. Die Benachrichtigung wird in diesem Fall an *„root“* gesendet. Der entsprechende Eintrag am Ende der Zeile kann natürlich auch angepasst werden.

```
/usr/local/bin/mailon-login.sh | mailx -s "Login on $(hostname)" root
```

Ab sofort wird bei jedem Login in der Konsole oder per SSH eine eMail versendet.

Reboot-Benachrichtigung

Nicht immer wird der Server durch den Admin selbst neu gestartet und ein Reboot kann gegebenenfalls auch zu Problemen führen. Eine Benachrichtigung über den Reboot des Systems kann daher sehr hilfreich sein. Auch hier wird der Inhalt der eMail mit folgendem Script definiert.

```
nano /usr/local/sbin/mailon-reboot.sh
```

```
#!/bin/bash
export HISTTIMEFORMAT='%F %T  '
echo "------------------------------------------------"
echo "System rebooted"
```

```
echo "------------------------------------------------"
echo "Hostname: $(hostname) "
echo "Date: $(date +%Y-%m-%d)"
echo "Time: $(date +%H:%M)"
echo
echo "------------------------------------------------"
echo "Last Activity"
echo "------------------------------------------------"
last
```

Das Script muss hier nur für den Benutzer „*root*“ ausführbar sein.

```
chmod u+x /usr/local/sbin/mailon-reboot.sh
```

Gestartet wird es über einen Eintrag in der systemweiten Cron-Tabelle.

```
nano /etc/crontab
```

In dieser wird ein spezieller Cronjob erstellt, der nach einem Reboot als „root“ ausgeführt wird. Auch hier kann der Empfänger der Benachrichtigung wieder am Ende der Zeile angepasst werden.

```
@reboot      root    /usr/local/sbin/mailon-reboot.sh | mailx -s "Reboot for $(hostname)" root
```

Nun wird das System nach jedem Reboot eine Benachrichtigung versenden.

18. Backup einrichten

Sobald man ein System produktiv betreiben will, sollte man sich auch Gedanken über die Datensicherung machen. Dabei ist im Wesentlichen zu klären, was, wann, wie und wohin gesichert werden soll.

Mit der folgenden Beispielkonfiguration wird jeden Tag um 02:00 Uhr ein Backup der Konfigurationsdateien, Logdateien, Datenverzeichnisse und der MySQL-Datenbanken erstellt und lokal unter *„/srv/backup“* abgelegt. Da auf dem Server nicht unbegrenzt Speicherplatz zur Verfügung steht, werden Backupdateien die älter als 7 Tage sind vom Server gelöscht. Daher sollten sie regelmäßig (z.B. per FTP) auf ein anderes System ausgelagert werden.

Vorbereitungen

Für das Backup wird rsync zum Kopieren der Daten und bzip2 für die Komprimierung benötigt.

```
apt-get install rsync bzip2
```

Zur Sicherung der MySQL-Datenbanken ist ein MySQL-Nutzer mit Leseberechtigung auf allen Datenbanken erforderlich. Hier sollte man einen extra Nutzer einrichten und das geht am schnellsten über die MySQL-Shell, an der man sich zunächst als MySQL-Root-Benutzer anmelden muss.

```
mysql --user=root --password
```

```
Enter password: ********
```

Mit folgendem SQL-Script wird ein neuer Nutzer *„backup“* erstellt und für alle Datenbanken auf dem Server mit lesenden Zugriff berechtigt. Das Passwort (********) sollte wie immer möglichst lang und komplex sein.

```
CREATE USER 'backup'@'localhost' IDENTIFIED BY '********';
GRANT SELECT ON * . * TO 'backup'@'localhost';
FLUSH PRIVILEGES;
EXIT
```

Da für die Sicherung der MySQL-Datenbanken eine Authentifizierung mit dem zuvor erstellten MySQL-Benutzer erforderlich ist und die Zugangsdaten hierfür nicht im Script hinterlegt werden sollten, kann man sie in einer MySQL-Optionsdatei speichern. Diese muss sich im Stammverzeichnis des Nutzers befinden, unter dessen Identität später das Backup-Script ausgeführt wird.

```
nano /root/.my.cnf
```

```
[mysqldump]
user = backup
password = ********
```

Auf diese Datei darf nur der Benutzer selbst (in diesem Fall *„root“*) Zugriff haben, daher ist sie mit entsprechenden Zugriffsrechten vor fremden Zugriff schützen.

```
chmod 0600 /root/.my.cnf
```

Soll das Backup verschlüsselt gespeichert werden, wird für die Verschlüsselung ein Sicherheitsschlüssel (Passphrase) benötigt. Auch dieser sollte nicht im Backup-Script hinterlegt werden, daher wird er ebenfalls in einer Datei gespeichert.

```
nano /root/.gpg.pwd
```

```
********
```

Und auch auf diese Datei darf natürlich nur Backup-Benutzer (in diesem Fall „*root*“) Zugriff haben.

```
chmod 0600 /root/.gpg.pwd
```

Backup-Script erstellen

Kern der Sicherung ist folgendes Shell-Script, in dem unter anderem festgelegt wird, was und wohin gesichert wird.

```
nano /usr/local/sbin/backup.sh
```

```
#!/bin/bash

# Target for the backup files
BACKUPDIR=/srv/data/backup
# Directory for temporary files
BACKUPTMPDIR=/tmp/backup

# Enable or disable encryption
GPGENABLE=1
# File to store encryption passphrase
GPGPWFILE=/root/.gpg.pwd

# Retention period in days
BACKUPRETENTION=7

DATE=`date +"%Y%m%d"`
TIME=`date +"%H%M%S"`

mkdir -p $BACKUPDIR
mkdir -p $BACKUPTMPDIR

rsync -av /srv $BACKUPTMPDIR --exclude=$BACKUPDIR
rsync -av /var/vmail $BACKUPTMPDIR
rsync -av /var/log $BACKUPTMPDIR
rsync -av /etc $BACKUPTMPDIR
rsync -av /usr/local/bin $BACKUPTMPDIR
rsync -av /usr/local/sbin $BACKUPTMPDIR

mysqldump --all-databases -- single-transaction > $BACKUPTMPDIR/mysqldump.sql

dpkg -l > $BACKUPTMPDIR/package.list

cd $BACKUPTMPDIR

if [ "$GPGENABLE" = "1" ]; then
  tar -cjp ./ | gpg --batch --no-tty -z 0 -c --passphrase-file $GPGPWFILE -o
$BACKUPDIR/backup_$DATE-$TIME.tar.bz2.gpg
else
  tar cjfp $BACKUPDIR/backup_$DATE-$TIME.tar.bz2 ./
fi

find -P $BACKUPDIR -maxdepth 1 -type f -iname 'backup_[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-
9]-[0-9][0-9][0-9][0-9][0-9][0-9].tar.bz2*' -type f -mtime +$BACKUPRETENTION -exec rm {} \;

rm -r -f $BACKUPTMPDIR
```

Nachdem die Datei gespeichert wurde, muss sie noch ausführbar gemacht werden.

```
chmod u+x /usr/local/sbin/backup.sh
```

Cronjob erstellen

Da das Script und damit auch die Sicherung jeden Tag um 02:00 Uhr starten soll, wird ein neuer Cronjob benötigt.

```
Ajenti -> SYSTEM -> Cron

root -> [AUSWÄHLEN]

Normal -> [HINZUFÜGEN]

0 0 1 1 1 false
Minute: 0
Stunde: 2
Tag: *
Monat: *
DoW: *
Befehl: sh /usr/local/sbin/backup.sh > /dev/null 2>&1

[SPEICHERN]
```

Die Sicherung ist nun fertig eingerichtet. Natürlich muss noch dafür gesorgt werden, dass die Backupdateien regelmäßig vom Server abgeholt werden.

19. Erste Domain und Webseite erstellen

Nachdem die Installation und Grundkonfiguration des Root-Servers weitestgehend abgeschlossen ist, kann nun die erste Domain und Website in Ajenti eingerichtet werden.

Webseite erstellen

Auch wenn zu diesem Zeitpunkt noch keine Inhalte veröffentlicht werden sollen, muss eine Webseite eingerichtet werden, denn mit der Einrichtung der Webseite erfolgt die Zuordnung der Domain, die unter anderem auch für die Erstellung eMail-Accounts benötigt wird.

```
Ajenti -> WEB -> Webseiten

NEUE WEBSEITE -> Name: mydomain.de -> [ERSTELLEN]
phpMyAdmin -> [VERWALTEN]

Allgemein
ALLGEMEIN -> Wartungsmodus: [ ]
WEBSEITE-DATEIEN -> Pfad: /srv/mydomain.de/www

Domains
[HINZUFÜGEN] -> Domain: mydomain.de
[HINZUFÜGEN] -> Domain: www.mydomain.de

Ports
[HINZUFÜGEN] -> Host: * | Port: 443 | SSL: [x]

SSL
SSL-Zertifikat-Pfad: /etc/ssl/certs/mydomain.de.crt
SSL-Schlüssel-Pfad: /etc/ssl/private/mydomain.de.key

Inhalt
PHP FastCGI - > [ERSTELLEN]
PHP -> PHP.ini-Werte -> alles löschen

Erweitert
Benutzerdefinierte Top-Level-Konfiguration:
server {
    server_name mydomain.de;
    return 301 $scheme://www.mydomain.de$request_uri;
}

[ÄNDERUNGEN ÜBERNEHMEN]
```

Um die Funktion des Webservers und von PHP zu testen wird ein kleines Script erstellt, welche beim Aufruf verschiedene Informationen zur PHP-Konfiguration ausgibt.

```
nano /srv/mydomain.de/www/info.php
```

```
<?php

phpinfo();

?>
```

```
chown www-data:www-data /srv/mydomain.de/www/info.php
```

Die Funktion der Webseiten kann im Browser überprüft werden.

```
URL: http://www.mydomain.de/info.php
```

Da dieses Script sehr viel über den Server verrät, sollte es nach dem Test wieder entfernt werden.

Standard eMail-Postfächer erstellen

Gemäß RFC 2142 sind grundsätzlich in einer Domain die Adressen „*postmaster@<domain>*“ für technische Anfragen und „*abuse@<domain>*“ für die Missbrauchsmeldungen anzulegen. Hierzu wird für den Postmaster ein eMail-Postfach und für die Abuse-Adresse eine eMail-Weiterleitung zum Postmaster eingerichtet.

```
Ajenti -> WEB -> E-Mail -> Allgemein

NEUES POSTFACH -> Name: postmaster@mydomain.de -> [POSTFACH]

postmaster@mydomain.de -> Passwort ändern -> Passwort: ********

NEUES POSTFACH -> Name: abuse@mydomain.de -> [WEITERLEITUNG]

abuse@mydomain.de -> Zieladresse -> [HINZUFÜGEN] -> postmaster@mydomain.de

[ÄNDERUNGEN ÜBERNEHMEN]
```

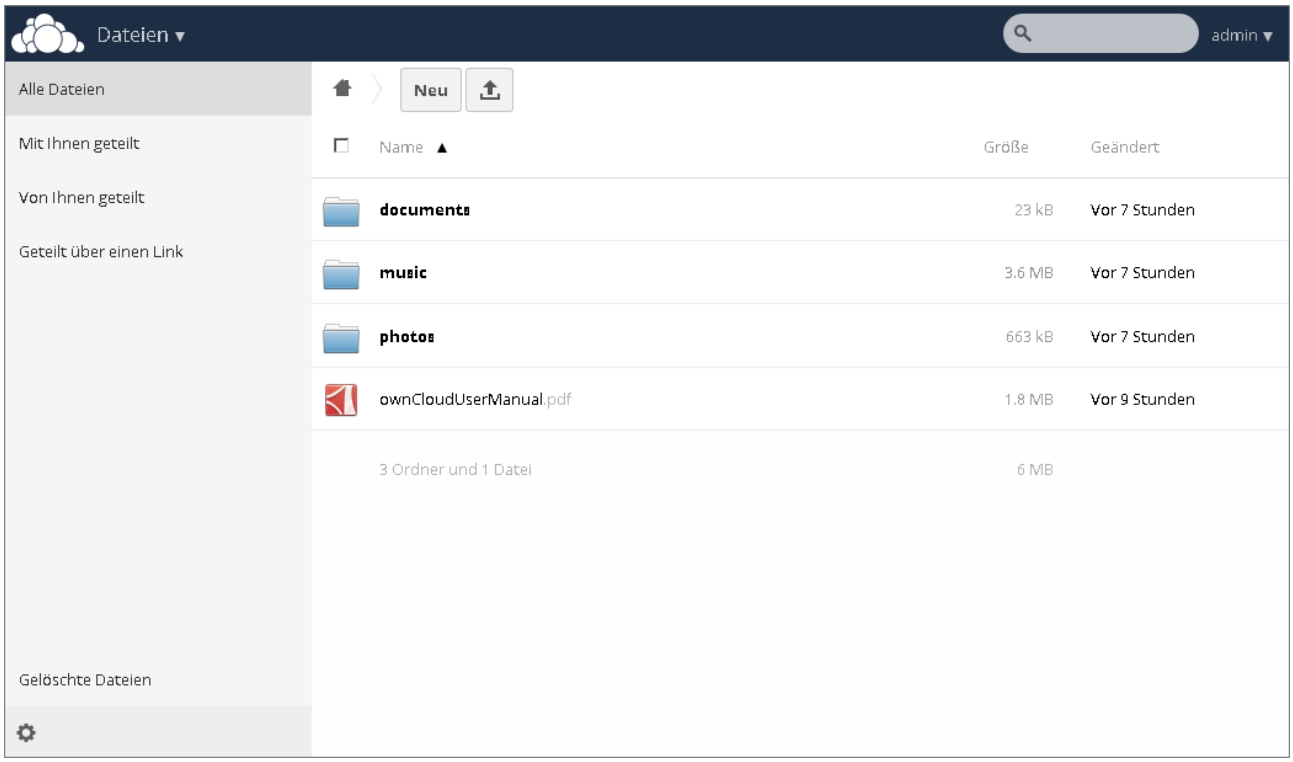
20. Beispiel-Anwendungen

Abschließend sollen hier noch ein paar interessante Anwendungen und deren Installation auf dem frisch eingerichteten Root-Server kurz vorgestellt werden.

ownCloud

Version: 7.0.2
Webseite: http://owncloud.org/

ownCloud ist wohl derzeit die bekannteste Cloudlösung im Bereich OpenSource. Die unter AGPL v3 lizensierte und ursprünglich als Online-Dateiverwaltung ausgelegte Software unterstützt mittlerweile eine Vielzahl von Funktionen, wie zum Beispiel Kalender, Aufgabenplaner, Adressbuch etc. Zudem lässt sich ownCloud mit vielen frei verfügbaren Apps erweitern und auch für die Anbindung mobiler Geräte und für die Synchronisation von Desktop-Systemen sind entsprechende Apps verfügbar.



ownCloud installieren

Die Installation beginnt im ersten Schritt mit dem Erstellen einer Webseite und einer MySQL-Datenbank in Ajenti. Da auch auf ownCloud natürlich nur verschlüsselt zugegriffen werden soll, wird auch hier eine entsprechende Umleitung von *„http“* auf *„https“* konfiguriert. Sollen später Dateien >1GB hochgeladen werden, müssen die Parameter *„post_max_size“*, *„upload_max_filesize“* und *„client_max_body_size“* entsprechend angepasst werden.

```
Ajenti -> WEB -> Webseiten

NEUE WEBSEITE -> Name: owncloud.mydomain.de -> [ERSTELLEN]
owncloud.mydomain.de -> [VERWALTEN]

Allgemein
ALLGEMEIN -> Wartungsmodus: [ ]
WEBSEITE-DATEIEN -> Pfad: /srv/mydomain.de/owncloud

Domains
```

```
[HINZUFÜGEN] -> Domain: owncloud.mydomain.de

Ports
[HINZUFÜGEN] -> Host: * | Port: 443 | SSL: [x]

SSL
SSL-Zertifikat-Pfad: /etc/ssl/certs/mydomain.de.crt
SSL-Schlüssel-Pfad: /etc/ssl/private/mydomain.de.key

Inhalt
PHP FastCGI - > [ERSTELLEN]
PHP -> PHP.ini-Werte:
file_uploads = On
post_max_size = 1G
upload_max_filesize = 1G
max_file_uploads = 20000
output_buffering = Off
allow_url_fopen = On
Erweitert -> Benutzerdefinierte Konfiguration
fastcgi_split_path_info ^(.+\.php)(/.+)$;

Erweitert
Benutzerdefinierte Konfiguration:
if ($server_port = 80) {
    rewrite ^ https://$host$request_uri permanent;
}

client_max_body_size 1G;

location ~ ^/(data|config|\.ht|db_structure\.xml|README) {
    deny all;
}

rewrite ^/caldav(.*)$ /remote.php/caldav$1 redirect;
rewrite ^/carddav(.*)$ /remote.php/carddav$1 redirect;
rewrite ^/webdav(.*)$ /remote.php/webdav$1 redirect;

MySQL
DATENBANKEN -> Name: owncloudmydomainde -> [ERSTELLEN]
BENUTZER -> Name: owncloudmydomainde -> [ERSTELLEN]

[ALLE BERECHTIGUNGEN ERTEILEN] -> [ÄNDERUNGEN ÜBERNEHMEN]
```

Anschließend wird das Root-Verzeichnis für die ownCloud-Installation erstellt, das Installationspaket heruntergeladen und entpackt.

```
mkdir /srv/mydomain.de/owncloud
cd /srv/mydomain.de/owncloud
wget https://download.owncloud.org/community/owncloud-7.0.2.tar.bz2
tar xfvj owncloud-7.0.2.tar.bz2
mv /srv/mydomain.de/owncloud/owncloud/* /srv/mydomain.de/owncloud/
rm owncloud -r
rm owncloud-7.0.2.tar.bz2
```

Damit Nginx den erforderlichen Zugriff auf die Webseite erhält, müssen die Besitzer- und Gruppenzugehörigkeit angepasst werden.

```
chown -R www-data:www-data /srv/mydomain.de/owncloud/
```

Nun kann Setup-Routine gestartet werden.

```
URL: https://owncloud.mydomain.de/
```

Hinweis: Das Kennwort für den für den Datenbankbenutzer ist in Ajenti, in der Verwaltung der Webseite unter MySQL hinterlegt.

```
Administrator-Konto anlegen
Benutzername: admin
Passwort: ********
Confirm*: ********

Speicher & Datenbank
Datenverzeichnis: /srv/mydomain.de/owncloud/data
Datenbank einrichten: MySQL/MariaDB
Datenbank-Benutzer: owncloudmydomainde
Datenbank-Passwort: ********
Datenbank-Name: owncloudmydomainde
Datenbank-Host: localhost

[Installation abschließen]
```

ownCloud führt standardmäßig Hintergrund-Aufgaben aus, die über Ajax-Aufrufe getriggert werden und daher immer dann laufen, wenn auf die ownCloud-Seite zugegriffen wird. Für eine bessere Performance wird aber empfohlen hierfür einen Cronjob zu nutzen.

```
Ajenti -> SYSTEM -> Cron

www-data -> [AUSWÄHLEN]

Normal -> [HINZUFÜGEN]

0 0 1 1 1 false
Minute: */15
Stunde: *
Tag: *
Monat: *
DoW: *
Befehl: /usr/bin/php /srv/mydomain.de/owncloud/cron.php > /dev/null 2>&1

[SPEICHERN]
```

Nachdem der Cronjob erstellt wurde, muss die dazugehörige Einstellung in ownCloud geändert werden.

```
ownCloud -> admin -> Administrator -> Cron
AJAX: [ ]
Webcron: [ ]
Cron: [x]
```

Die Installation ist damit abgeschlossen und ownCloud kann verwendet werden.

```
URL: https://owncloud.mydomain.de/
```

Fail2ban-Filter erstellen

Soll ownCloud mittels Fail2ban vor Brute-Force-Attaken geschützt werden, muss ein neuer Filter erstellt werden.

```
nano /etc/fail2ban/filter.d/owncloud-login.conf
```

```
# Fail2Ban configuration file
#
# Block failed ownCloud-Authentication
#

[Definition]
```

```
# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = {"app":"core","message":"Login failed: '.*' \(Remote IP: '<HOST>', X-Forwarded-
For: '.*'\)","level":2,"time":".*"}

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Nach dem der Filter für ownCloud erstellt wurde, muss dieser noch aktiviert werden.

```
nano /etc/fail2ban/jail.local
```

```
[owncloud-login]
enabled   = true
port      = http,https
filter    = owncloud-login
logpath   = /srv/mydomain.de/owncloud/data/owncloud.log
maxretry  = 3
```

Konfigurationsdateien speichern anschließend Fail2ban neu starten.

```
service fail2ban restart
```

Für die Logfiles, die durch Fail2ban überwacht werden, sollte die lokale Zeitzone verwendet werden. Bei ownCloud muss hierzu ein Parameter zur Konfigurationsdatei hinzugefügt werden.

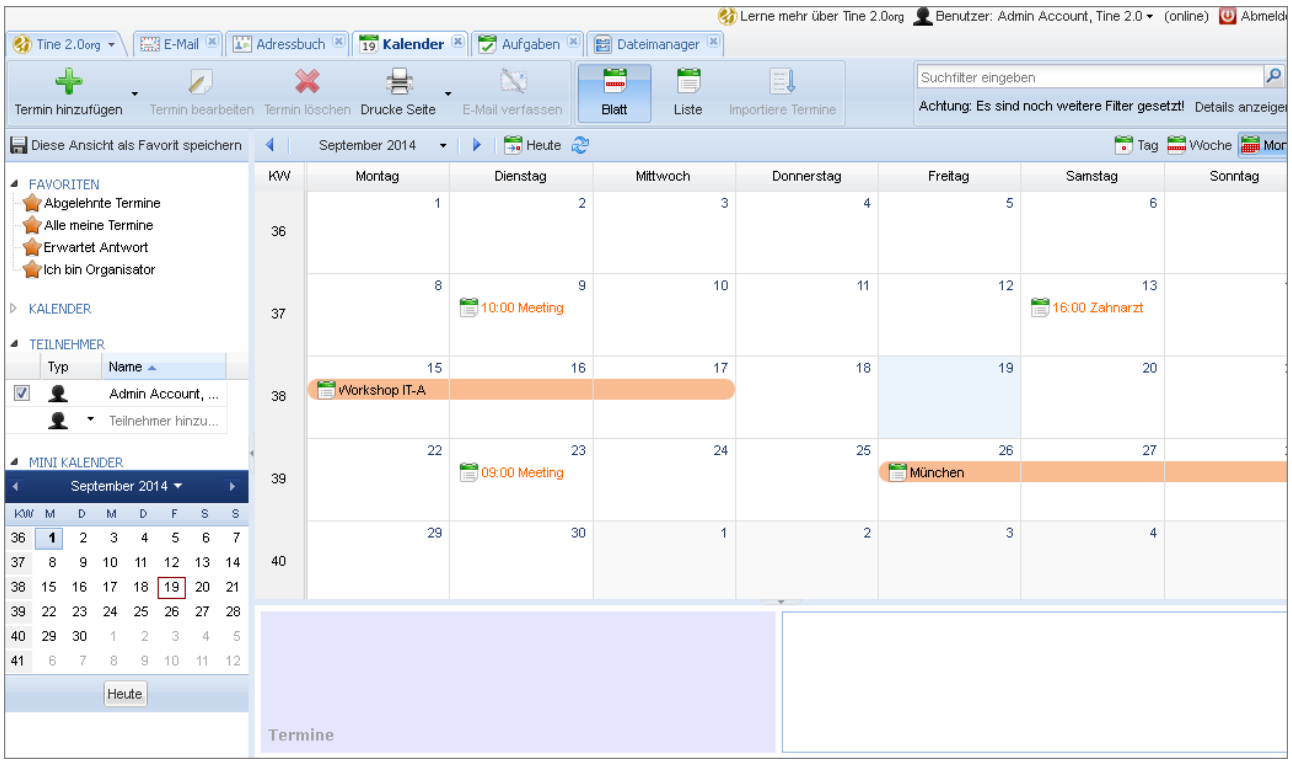
```
nano /srv/mydomain.de/owncloud/config/config.php
```

```
'logtimezone' => 'Europe/Berlin',
```

Tine 2.0

Version: Koriander (2013.09.1)
Webseite: https://www.tine20.org/

Tine 2.0 ist eine freie, unter AGPL v3 lizensierte und recht umfangreiche Groupware-Lösung, die sich hervorragend als Alternative zu datenhungrigen Diensten wie zum Beispiel Google anbietet. Neben den obligatorischen Funktionen wie eMail, Adressbuch, Kalender und Aufgaben bietet Tine 2.0 noch eine Reihe weiterer Funktionen die bei Bedarf installiert bzw. aktiviert werden können. So kann zum Beispiel mit dem ActiveSync-Modul die Synchronisierung mit dem Smartphone oder anderen Geräten sichergestellt werden. Es werden aber auch CalDAV, CardDAV und WebDAV unterstützt. Zudem biete Tine 2.0 eine durchdachte und sehr moderne Ajax-Oberfläche mit der es wirklich Spaß mach zu arbeiten.



Tine 2.0 installieren

Für die Installation von Tine 2.0 muss wieder eine Webseite und eine MySQL-Datenbank in Ajenti angelegt werden. Dabei wird auch eine Nginx-Regel erstellt, die automatisch von http auf https umleitet, so dass die Seite nur SSL-verschlüsselt erreichbar ist.

```
Ajenti -> WEB -> Webseiten

NEUE WEBSEITE -> Name: tine.mydomain.de -> [ERSTELLEN]
tine.mydomain.de -> [VERWALTEN]

Allgemein
ALLGEMEIN -> Wartungsmodus: [ ]
WEBSEITE-DATEIEN -> Pfad: /srv/mydomain.de/tine

Domains
[HINZUFÜGEN] -> Domain: tine.mydomain.de

Ports
[HINZUFÜGEN] -> Host: * | Port: 443 | SSL: [x]

SSL
SSL-Zertifikat-Pfad: /etc/ssl/certs/mydomain.de.crt
SSL-Schlüssel-Pfad: /etc/ssl/private/mydomain.de.key
```

```
Inhalt
PHP FastCGI - > [ERSTELLEN]
PHP -> PHP.ini-Werte -> alles löschen

Erweitert
Benutzerdefinierte Konfiguration:
if ($server_port = 80) {
    rewrite ^ https://$host$request_uri permanent;
}

if ($request_method !~ "^(GET|POST)$"){
    rewrite ^/$ /index.php?frontend=webdav;
    rewrite ^/addressbooks /index.php?frontend=webdav;
    rewrite ^/calendars /index.php?frontend=webdav;
    rewrite ^/principals /index.php?frontend=webdav;
    rewrite ^/webdav /index.php?frontend=webdav;
}
```

```
rewrite ^/Microsoft-Server-ActiveSync?(.*)$ /Microsoft-Server-ActiveSync/index.php?$1;

MySQL
DATENBANKEN -> Name: tinemydomainde -> [ERSTELLEN]
BENUTZER -> Name: tinemydomainde -> [ERSTELLEN]

[ALLE BERECHTIGUNGEN ERTEILEN] -> [ÄNDERUNGEN ÜBERNEHMEN]
```

Anschließend wird das Root-Verzeichnis für die Tine 2.0-Installation erstellt, das Installationspaket heruntergeladen und entpackt.

```
mkdir /srv/mydomain.de/tine
cd /srv/mydomain.de/tine
wget http://www.tine20.org/downloads/2014.09.1/tine20-allinone_2014.09.1.zip
apt-get install unzip
unzip tine20-allinone_2014.09.1.zip
rm tine20-allinone_2014.09.1.zip
```

Nun muss die Konfigurationsdatei erstellt und angepasst werden, hierzu wird einfach eine Kopie der Vorlage erzeugt.

```
cp config.inc.php.dist config.inc.php
nano config.inc.php
```

In dieser Datei werden die Verbindung zum MySQL-Server, die Anmeldedaten für den Setup-User und verschiedene Speicherpfade festgelegt.

Der für die weitere Installation und Konfiguration erforderliche Setup-User wird im Abschnitt „*setupuser*“ festgelegt. Dessen Benutzername kann frei vergeben werden und das dazugehörige Kennwort sollte natürlich möglichst komplex sein.

Hinweis: Das Kennwort für den für den Datenbankbenutzer ist in Ajenti, in der Verwaltung der Webseite unter MySQL hinterlegt.

```
<?php
return array (
  'captcha' =>
  array (
    'count' => 0,
  ),
  'database' =>
  array (
    'host' => 'localhost',
    'dbname' => 'tinemydomainde',
```

```
    'username' => 'tinemydomainde',
    'password' => '********',
    'adapter' => 'pdo_mysql',
    'tableprefix' => 'tine20_',
    'port' => '',
  ),
  'setupuser' =>
  array (
    'username' => 'tine20setup',
    'password' => '********',
  ),
  'logger' =>
  array (
    'active' => true,
    'filename' => '/srv/mydomain.de/tine/data/log/tine20.log',
    'priority' => 4,
  ),
  'caching' =>
  array (
    'active' => true,
    'lifetime' => 3600,
    'backend' => 'File',
    'path' => '/srv/mydomain.de/tine/data/cache',
    'redis' =>
    array (
      'host' => 'localhost',
      'port' => 6379,
    ),
    'memcached' =>
    array (
      'host' => 'localhost',
      'port' => 11211,
    ),
  ),
  'actionqueue' =>
  array (
    'active' => false,
    'backend' => 'Redis',
    'host' => 'localhost',
    'port' => 6379,
  ),
  'session' =>
  array (
    'lifetime' => 86400,
    'backend' => 'File',
    'path' => '/srv/mydomain.de/tine/data/sessions',
    'host' => 'localhost',
    'port' => 6379,
  ),
  'tmpdir' => '/srv/mydomain.de/tine/data/tmp',
  'filesdir' => '/srv/mydomain.de/tine/data/files',
  'mapPanel' => '',
);
```

Die in der Konfigurationsdatei angegebenen Verzeichnisse sind noch zu erstellen und damit Nginx den erforderlichen Zugriff auf die Webseite erhält, müssen die Besitzer- und Gruppenzugehörigkeit angepasst werden.

```
mkdir -p /srv/mydomain.de/tine/data/log
mkdir -p /srv/mydomain.de/tine/data/cache
mkdir -p /srv/mydomain.de/tine/data/sessions
mkdir -p /srv/mydomain.de/tine/data/tmp
mkdir -p /srv/mydomain.de/tine/data/files
chown -R www-data:www-data /srv/mydomain.de/tine/
```

Das eigentliche Setup von Tine 2.0 wird nun gestartet

```
URL: https://tine.mydomain.de/setup.php
Username: tine20setup
Password: ********
```

Die obligatorischen Lizenz- und Datenschutzvereinbarungen müssen natürlich akzeptiert werden.

```
Terms and Conditions
License Agreement: [x]
Privacy Agreement: [x]

[Accept Terms and Conditions]
```

Die einzelnen Systemvoraussetzungen werden überprüft. Hier sollte im Ergebnis alles grün sein.

```
Setup Checks

[Run setup tests]
```

In der Konfigurationsverwaltung werden die Parameter der „*config.inc.php*“ festgelegt. Da dies bereits erfolgt ist, sind hier keine Anpassungen erforderlich.

```
Config Manager
```

Im Nächsten Abschnitt wird das Admin-Konto der Tine 2.0-Installation erzeugt.

```
Authentication/Accounts
Initial Admin User
Initial admin login name: admin
Initial admin Password: ********
Password confirmation: ********

[Save config and install]
```

Für die Benachrichtigungsfunktion (z.B. Kalendererinnerung) sind folgende eMail-Einstellungen notwendig.

ACHTUNG! Seit PHP 5.6.x ist die Funktion „*Peer Verification*“ standardmäßig aktiviert, so dass nun die Gültigkeit des SSL-Zertifikates überprüft wird. Daher muss als Hostname der Name verwendet werden, auf den das Zertifikat ausgestellt wurde. Der Eintrag „*localhost*“ funktioniert daher an dieser Stelle nicht mehr.

```
Email
Imap [x]
Backend: Standard IMAP
Hostname: v1234567890.yourvserver.net
Port: 993
Secure Connection: SSL
Use system account: No
Append domain to login name:

Smtp [x]
Backend: Standard SMTP
Hostname: v1234567890.yourvserver.net
Port: 465
Secure Connection: SSL
Authentication: None
Primary Domain:
Secondary Domains (comma separated):
Notifications service address: postmaster@mydomain.de
Notification Username:
Notification Password:
Notifications local client (hostname or IP address): localhost

SIEVE [ ]
```

```
[Save config]
```

Im letzten Setup-Abschnitt werden die benötigten Funktionen installiert.

```
Application Manager
ActiveSync: enabled
Addressbook: enabled
Admin: enabled
Calendar: enabled
Crm: disabled
Felamimail: enabled
Filemanager: enabled
Projects: disabled
Sales: disabled
Tasks: enabled
Timetracker: disabled
Tinebase: enabled
```

Die Installation ist damit abgeschlossen und Tine 2.0 kann verwendet werden.

```
URL: https://tine.mydomain.de/
```

ActiveSync konfigurieren

Leider wird Nginx bisher nicht offiziell von Tine 2.0 supported, weshalb ActiveSync auch erst nach einer kleinen Änderung im Quellcode funktioniert.

```
nano /srv/mydomain.de/tine/Tinebase/Core.php
```

Der folgende Abschnitt muss unterhalb von Zeile 215 (ActiveSync API) eingefügt werden.

```
} elseif(strpos($_SERVER['REQUEST_URI'],'/Microsoft-Server-ActiveSync') !== false) {
        $server = new ActiveSync_Server_Http();
```

Die bei der Erstellung der Webseite angegebene rewrite-Regel für ActiveSync verweist auf einen Pfad, der als symbolische Verknüpfung erstellt werden muss.

```
ln -s /srv/mydomain.de/tine /srv/mydomain.de/tine/Microsoft-Server-ActiveSync
```

Nach dem die Datei gespeichert wurde, sollte ActiveSync funktionieren, was man auch über folgenden Aufruf überprüfen kann.

```
URL: https://tine.mydomain.de/Microsoft-Server-ActiveSync
```

Nach der Authentifizierung mit einem Tine 2.0 Account muss folgende Meldung zu sehen sein.

```
It works!
Your userid is: e0f3a12eac3554a2376b56d1c48a642a4c7ce875 and your IP address is:
xxx.xxx.xxx.xxx
```

Kalenderalarme konfigurieren

Neben der Benachrichtigungsfunktion durch das Endgerät (z.B. Smartphone oder Tablet) können Kalenderbenachrichtigungen auch als eMail versendet werden. Hierfür ist ein Cronjob erforderlich, der in regelmäßigen Abständen ausgeführt werden muss. In dieser Konfiguration wurde ein Intervall von 5 Minuten gewählt.

```
Ajenti -> SYSTEM -> Cron

www-data -> [AUSWÄHLEN]
```

```
Normal -> [HINZUFÜGEN]

0 0 1 1 1 false
Minute: */5
Stunde: *
Tag: *
Monat: *
DoW: *
Befehl: /usr/bin/php /srv/mydomain.de/tine/tine20.php --method Tinebase.triggerAsyncEvents >
/dev/null 2>&1

[SPEICHERN]
```

Fail2ban-Filter erstellen

Soll Tine 2.0 auch mittels Fail2ban vor Brute-Force-Attaken geschützt werden, muss ein neuer Filter erstellt werden.

```
nano /etc/fail2ban/filter.d/tine-login.conf
```

```
# Fail2Ban configuration file
#
# Block failed Tine 2.0-Authentication
#

[Definition]

# Option:  failregex
# Notes.:  Regexp to match often probed.
# Values:  TEXT
#
failregex = Login with username .* from <HOST> failed

# Option:  ignoreregex
# Notes.:  Regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Der Filter muss natürlich noch eingebunden bzw. aktiviert werden.

```
nano /etc/fail2ban/jail.local
```

```
[tine-login]
enabled   = true
port      = http,https
filter    = tine-login
logpath   = /srv/mydomain.de/tine/data/log/tine20.log
findtime  = 7200
maxretry  = 3
```

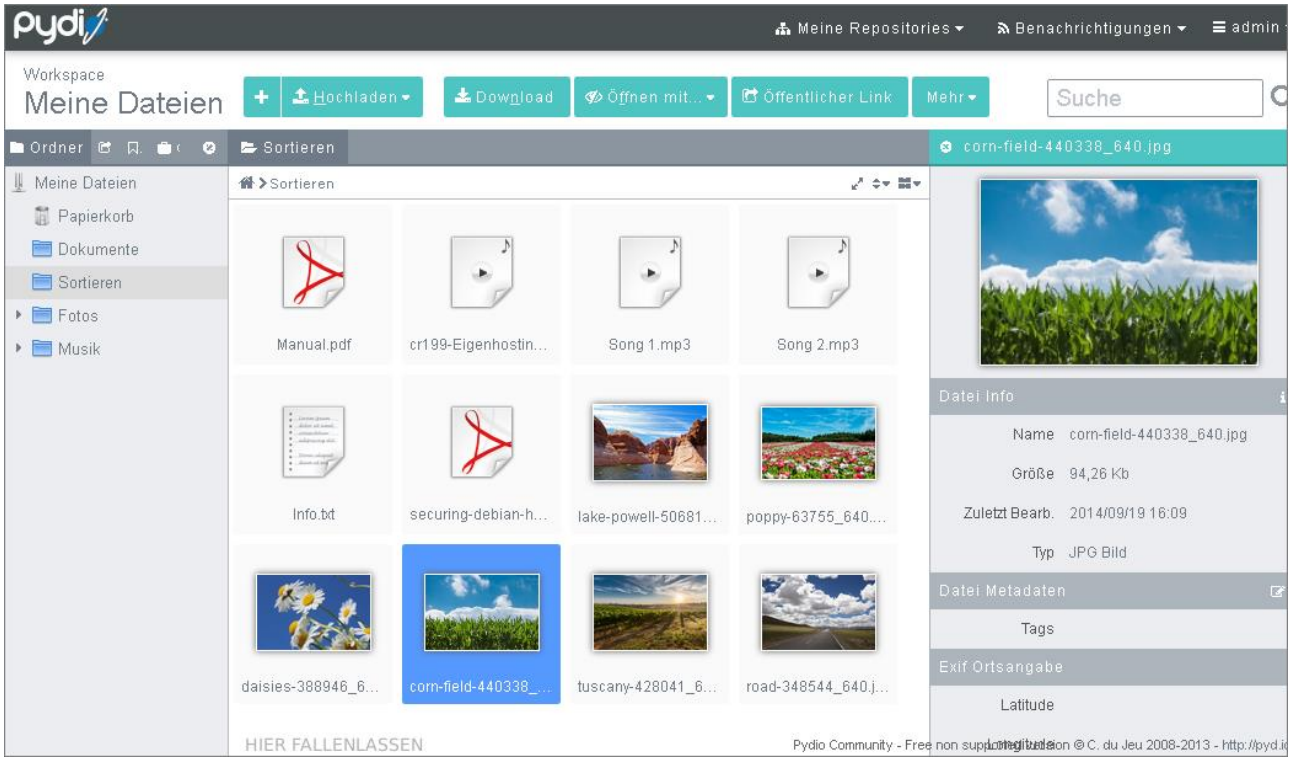
Konfigurationsdateien speichern anschließend Fail2ban neu starten.

```
service fail2ban restart
```

Pydio

Version: 5.2.4
Webseite: http://pyd.io/

Pydio ist eine Online-Dateiverwaltung die als Alternative zu Dropbox und Co. sehr gut geeignet ist, um eine eigene Cloud zu betreiben. Die Software ist ebenfalls unter AGPL v3 lizensiert und bietet viele Features wie zum Beispiel eine Versionierung, eine App für mobile Geräte und auch die Möglichkeiten zur Anbindung von verschiedenen Speicherdiensten sind sehr umfangreich.



Pydio installieren

Erster Schritt der Installation von Pydio ist auch hier wieder das Erstellen einer Webseite und einer MySQL-Datenbank in Ajenti. Auch auf diese Seite soll nur SSL-verschlüsselt zugegriffen werden, daher wird auch hier eine entsprechende Umleitung konfiguriert. Sollen später Dateien >1GB hochgeladen werden, müssen die Parameter „*post_max_size*“, „*upload_max_filesize*“ und „*client_max_body_size*“ entsprechend angepasst werden.

```
Ajenti -> WEB -> Webseiten

NEUE WEBSEITE -> Name: pydio.mydomain.de -> [ERSTELLEN]
pydio.mydomain.de -> [VERWALTEN]

Allgemein
ALLGEMEIN -> Wartungsmodus: [ ]
WEBSEITE-DATEIEN -> Pfad: /srv/mydomain.de/pydio

Domains
[HINZUFÜGEN] -> Domain: pydio.mydomain.de

Ports
[HINZUFÜGEN] -> Host: * | Port: 443 | SSL: [x]

SSL
SSL-Zertifikat-Pfad: /etc/ssl/certs/mydomain.de.crt
SSL-Schlüssel-Pfad: /etc/ssl/private/mydomain.de.key
```

```
Inhalt
PHP FastCGI - > [ERSTELLEN]
PHP -> PHP.ini-Werte:
file_uploads = On
post_max_size = 1G
upload_max_filesize = 1G
max_file_uploads = 20000
output_buffering = Off

Erweitert
Benutzerdefinierte Konfiguration:
if ($server_port = 80) {
    rewrite ^ https://$host$request_uri permanent;
}

client_max_body_size 1G;

location ^~ /conf/      { deny all; }
location ^~ /data/      { deny all; }
location = /robots.txt  { access_log off; log_not_found off; }
location = /favicon.ico { access_log off; log_not_found off; }
location ~ /\.          { access_log off; log_not_found off; deny all; }
location ~ ~$           { access_log off; log_not_found off; deny all; }

location ~* \.(?:ico|css|js|gif|jpe?g|png)$ {
    expires max;
    add_header Pragma public;
    add_header Cache-Control "public, must-revalidate, proxy-revalidate";
}

MySQL
DATENBANKEN -> Name: pydiomydomainde -> [ERSTELLEN]
BENUTZER -> Name: pydiomydomainde -> [ERSTELLEN]

[ALLE BERECHTIGUNGEN ERTEILEN] -> [ÄNDERUNGEN ÜBERNEHMEN]
```

Anschließend wird das Root-Verzeichnis für die Pydio-Installation erstellt, das Installationspaket heruntergeladen und entpackt.

```
mkdir /srv/mydomain.de/pydio
cd /srv/mydomain.de/pydio
wget http://freefr.dl.sourceforge.net/project/ajaxplorer/pydio/stable-channel/5.2.4/pydio-core-5.2.4.zip
unzip pydio-core-5.2.4.zip
mv /srv/mydomain.de/pydio/pydio-core-5.2.4/* /srv/mydomain.de/pydio/
rm pydio-core-5.2.4 -r
rm pydio-core-5.2.4.zip
```

In der Konfigurationsdatei müssen die Spracheinstellungen und der Pfad für temporäre Dateien angepasst werden.

```
nano /srv/mydomain.de/pydio/conf/bootstrap_conf.php
```

```
define("AJXP_LOCALE", "de_DE.UTF-8");
define("AJXP_TMP_DIR", "/srv/tmp");
```

Damit Nginx den erforderlichen Zugriff auf die Webseite erhält, müssen die Besitzer- und Gruppenzugehörigkeit angepasst werden.

```
chown -R www-data:www-data /srv/mydomain.de/pydio/
```

Das eigentliche Setup von Pydio kann nun gestartet werden.

```
URL: https://pydio.mydomain.de/
```

ACHTUNG! Soll „*PHP Command Line*“ für die Ausführung von Prozessen im Hintergrund verwendet werden, muss in der Datei „*/etc/php5/fpm/php.ini*“ die Einträge „*exec*“ und „*popen*“ aus der Zeile „*disable_functions =*“ entfernt werden. Pydio kann aber auch problemlos ohne diese Funktion installiert und genutzt werden. Hier muss man abwägen, ob einem die Funktion oder die zusätzliche Sicherheit wichtiger ist.

Hinweis: Das Kennwort für den für den Datenbankbenutzer ist in Ajenti, in der Verwaltung der Webseite unter MySQL hinterlegt.

```
Pydio Diagnostic Tool -> [click here to continue to Pydio.]
Pydio Setup Wizard -> [Deutsch] -> [Start Wizard!]

Admin access
Admin Login*: admin
Admin Display Name*: admin
Admin Password*: ********
Confirm*: ********

Global options
Application Title: Pydio
Welcome Message: Willkommen bei Pydio
Default Language*: Deutsch
Enable emails*: No (you can enable mails later)

Configurations storage
Storage Type: Databse (Requires MySQL, PostgreSQL or SQLite)
Enable Notifications: Yes
Database*: MySQL
Host*: localhost
Database*: pydiomydomainde
User*: pydiomydomainde
Password*: ********
Test SQL Connexion -> [Try connecting to the database]

[Install Pydio Now]
```

Die Installation ist damit abgeschlossen und Pydio kann verwendet werden.

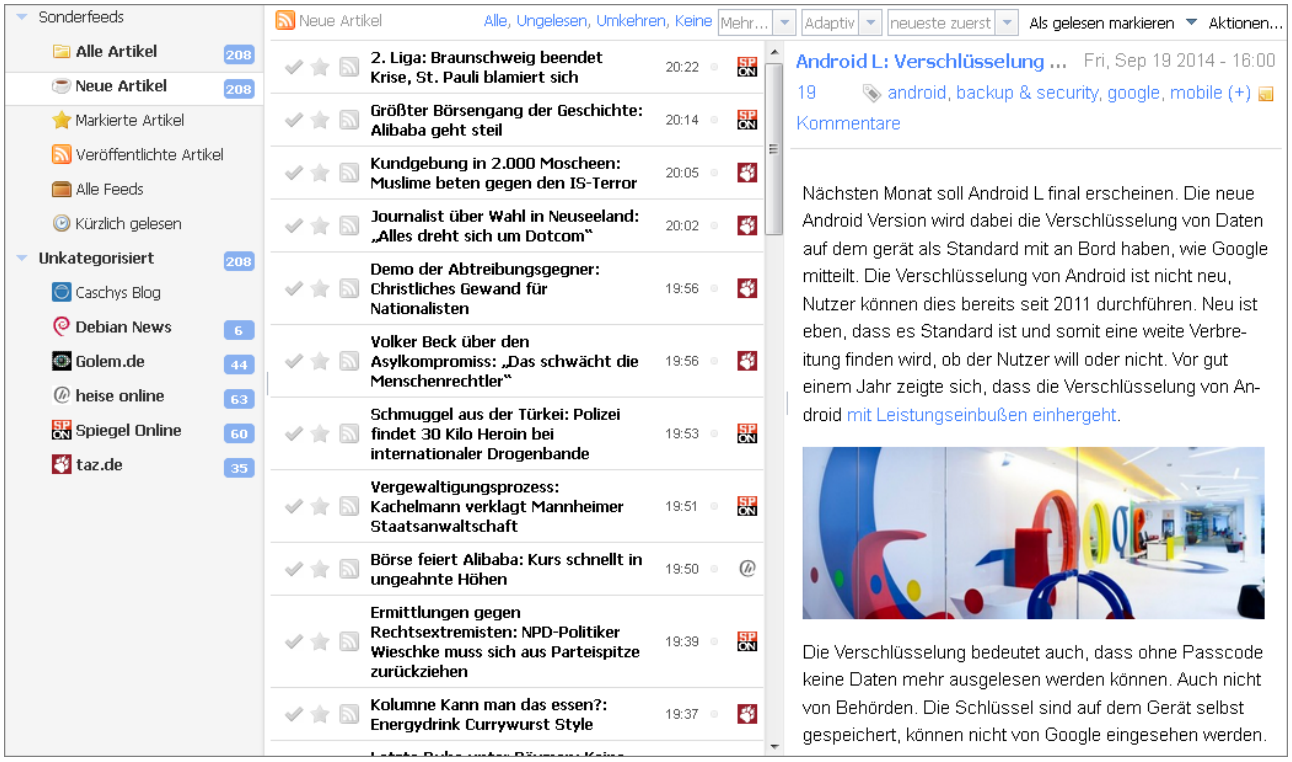
```
URL: https://pydio.mydomain.de/
```

Hinweis: Für Anbindung einiger Speicherdienste wie zum Beispiel SFTP (SSH) oder Samba sind zusätzliche Regeln für ausgehende Verbindungen in der Firewall erforderlich.

Tiny Tiny RSS

Version: 1.14
Webseite: http://tt-rss.org/

Tiny Tiny RSS ist ein webbasierter Reader und Aggregator für RSS-Feeds mit großem Funktionsumfang, der auch im Multiuserbetrieb genutzt werden. Für die Verwendung auf mobilen Geräten sind mehrere Apps für iOS und Android verfügbar und wie die anderen hier vorgestellten Applikationen, ist auch Tiny Tiny RSS Open Source (GNU GPL).



Tiny Tiny RSS installieren

Die Installation startet wieder mit dem Erstellen der Webseite und einer MySQL Datenbank.

```
Ajenti -> WEB -> Webseiten

NEUE WEBSEITE -> Name: rss.mydomain.de -> [ERSTELLEN]
rss.mydomain.de -> [VERWALTEN]

Allgemein
ALLGEMEIN -> Wartungsmodus: [ ]
WEBSEITE-DATEIEN -> Pfad: /srv/mydomain.de/rss

Domains
[HINZUFÜGEN] -> Domain: rss.mydomain.de

Ports
[HINZUFÜGEN] -> Host: * | Port: 443 | SSL: [x]

SSL
SSL-Zertifikat-Pfad: /etc/ssl/certs/mydomain.de.crt
SSL-Schlüssel-Pfad: /etc/ssl/private/mydomain.de.key

Inhalt
PHP FastCGI - > [ERSTELLEN]
PHP -> PHP.ini-Werte -> alles löschen
Erweitert -> Benutzerdefinierte Konfiguration
fastcgi_read_timeout 500;
fastcgi_send_timeout 500;
```

```
Erweitert
Benutzerdefinierte Konfiguration:
if ($server_port = 80) {
    rewrite ^ https://$host$request_uri permanent;
}

MySQL
DATENBANKEN -> Name: rssmydomainde -> [ERSTELLEN]
BENUTZER -> Name: rssmydomainde -> [ERSTELLEN]

[ALLE BERECHTIGUNGEN ERTEILEN] -> [ÄNDERUNGEN ÜBERNEHMEN]
```

Anschließend wird das Root-Verzeichnis für die Tiny Tiny RSS-Installation erstellt, das Installationspaket heruntergeladen und entpackt.

```
mkdir /srv/mydomain.de/rss
cd /srv/mydomain.de/rss
wget https://github.com/gothfox/Tiny-Tiny-RSS/archive/1.14.tar.gz
tar zxfv 1.14.tar.gz
mv /srv/mydomain.de/rss/Tiny-Tiny-RSS-1.14/* /srv/mydomain.de/rss/
rm Tiny-Tiny-RSS-1.14 -r
rm 1.14.tar.gz
```

Damit Nginx den erforderlichen Zugriff auf die Webseite erhält, müssen die Besitzer- und Gruppenzugehörigkeit angepasst werden.

```
chown -R www-data:www-data /srv/mydomain.de/rss/
```

Das eigentliche Setup von Tiny Tiny RSS kann nun gestartet werden.

```
URL: https://rss.mydomain.de/install
```

Hinweis: Das Kennwort für den für den Datenbankbenutzer ist in Ajenti, in der Verwaltung der Webseite unter MySQL hinterlegt.

```
Database settings
Database type: MySQL
Username: rssmydomainde
Password: ********
Database name: rssmydomainde
Host name: localhost
Port: 3306

Other settings
Tiny Tiny RSS URL: https://rss.mydomain.de/

[Test configuration] -> [Initialize database] -> [Save configuration]
```

Nachdem das Setup erfolgreich abgeschlossen wurde, sollte die Funktion überprüft werden.

```
URL: https://rss.mydomain.de
Username: admin
Password: password
```

Das initale Admin-Kennwort kann dabei gleich geändert werden.

```
Tiny Tiny RSS

Aktionen... -> Einstellungen...
Persönliche Daten / Authentifizierung

Passwort
```

```
Altes Passwort: password
Neues Passwort: ********
Passwort bestätigen: ********

[Passwort ändern]
```

Die automatische Aktualisierung der in Tiny Tiny RSS abonnierten Feeds wird über einen Cronjob sichergestellt. In diesem Beispiel soll dieser alle 10 Minuten ausgeführt werden.

```
Ajenti -> SYSTEM -> Cron

www-data -> [AUSWÄHLEN]

Normal -> [HINZUFÜGEN]

0 0 1 1 1 false
Minute: */10
Stunde: *
Tag: *
Monat: *
DoW: *
Befehl: /usr/bin/php /srv/mydomain.de/rss/update.php --feeds --quiet > /dev/null 2>&1

[SPEICHERN]
```

Die Installation ist damit abgeschlossen und Tiny Tiny RSS kann verwendet werden.

```
URL: https://rss.mydomain.de/
```

Anhänge

Übersicht der installierten Pakete

Name	Version	Architektur	Beschreibung
acpi	1.6-1	i386	displays information on ACPI devices
acpi-support-base	0.140-5+deb7u3	all	scripts for handling base ACPI events such as the power button
acpid	1:2.0.16-1+deb7u1	i386	Advanced Configuration and Power Interface event daemon
adduser	3.113+nmu3	all	add and remove users and groups
ajenti	1.2.22.16	all	Server administration web interface
ajenti-v	0.2.52	all	Virtual hosting solution for Ajenti
ajenti-v-ftp-pureftpd	0.1.4	all	PureFTPd support for Ajenti V
ajenti-v-mail	0.1.30	all	Mail for Ajenti V
ajenti-v-mysql	0.3.1	all	MySQL support for Ajenti V
ajenti-v-nginx	0.1.37	all	NGINX support for Ajenti V
ajenti-v-php-fpm	0.1.21	all	PHP support for Ajenti V (via PHP-FPM)
apt	0.9.7.9+deb7u6	i386	commandline package manager
apt-show-versions	0.20	all	lists available package versions with distribution
apt-utils	0.9.7.9+deb7u6	i386	package managment related utility programs
aptitude	0.6.8.2-1	i386	terminal-based package manager
aptitude-common	0.6.8.2-1	all	architecture indepedent files for the aptitude package manager
aspell	0.60.7~20110707-1	i386	GNU Aspell spell-checker
aspell-de	20120607-1	all	German dictionary for aspell
aspell-de-alt	1:2-28	all	German dictionary for aspell (old spelling)
base-files	7.1wheezy7	i386	Debian base system miscellaneous files
base-passwd	3.5.26	i386	Debian base system master password and group files
bash	4.2+dfsg-0.1+deb7u3	i386	GNU Bourne Again SHell
bsdmainutils	9.0.3	i386	collection of more utilities from FreeBSD
bsdutils	1:2.20.1-5.3	i386	Basic utilities from 4.4BSD-Lite
busybox	1:1.20.0-7	i386	Tiny utilities for small and embedded systems
bzip2	1.0.6-4	i386	high-quality block-sorting file compressor - utilities
ca-certificates	20130119+deb7u1	all	Common CA certificates
console-setup	1.88	all	console font and keymap setup program
console-setup-linux	1.88	all	Linux specific part of console-setup
coreutils	8.13-3.5	i386	GNU core utilities
courier-authdaemon	0.63.0-6+b1	i386	Courier authentication daemon
courier-authlib	0.63.0-6+b1	i386	Courier authentication library
courier-authlib-userdb	0.63.0-6+b1	i386	userdb support for the Courier authentication library
courier-base	0.68.2-1	i386	Courier mail server - base system
courier-imap	4.10.0-20120615-1	i386	Courier mail server - IMAP server
courier-imap-ssl	4.10.0-20120615-1	i386	Courier mail server - IMAP over SSL
courier-pop	0.68.2-1	i386	Courier mail server - POP3 server
courier-pop-ssl	0.68.2-1	i386	Courier mail server - POP3 over SSL
courier-ssl	0.68.2-1	i386	Courier mail server - SSL/TLS Support
cpio	2.11+dfsg-0.1	i386	GNU cpio -- a program to manage archives of files
cron	3.0pl1-124	i386	process scheduling daemon
cron-apt	0.9.1	all	automatic update of packages using apt-get
dash	0.5.7-3	i386	POSIX-compliant shell
dbconfig-common	1.8.47+nmu1	all	common framework for packaging database applications
dbus	1.6.8-1+deb7u4	i386	simple interprocess messaging system (daemon and utilities)
debconf	1.5.49	all	Debian configuration management system
debconf-i18n	1.5.49	all	full internationalization support for debconf
debian-archive-keyring	2014.1~deb7u1	all	GnuPG archive keys of the Debian archive
debianutils	4.3.2	i386	Miscellaneous utilities specific to Debian
dictionaries-common	1.12.11	all	Common utilities for spelling dictionary tools
diffutils	1:3.2-6	i386	File comparison utilities
discover	2.1.2-5.2	i386	hardware identification system
discover-data	2.2010.10.18	all	Data lists for Discover hardware detection system
dmidecode	2.11-9	i386	SMBIOS/DMI table decoder
dmsetup	2:1.02.74-8	i386	Linux Kernel Device Mapper userspace library
dpkg	1.16.15	i386	Debian package management system
e2fslibs:i386	1.42.5-1.1	i386	ext2/ext3/ext4 file system libraries
e2fsprogs	1.42.5-1.1	i386	ext2/ext3/ext4 file system utilities
eject	2.1.5+deb1+cvs2008110	i386	ejects CDs and operates CD-Changers under Linux
exim4	4.80-7+deb7u1	all	metapackage to ease Exim MTA (v4) installation
exim4-base	4.80-7+deb7u1	i386	support files for all Exim MTA (v4) packages
exim4-config	4.80-7+deb7u1	all	configuration for the Exim MTA (v4)
exim4-daemon-light	4.80-7+deb7u1	i386	lightweight Exim MTA (v4) daemon
expect	5.45-2	i386	Automates interactive applications
fail2ban	0.8.6-3wheezy3	all	ban hosts that cause multiple authentication errors
file	5.11-2+deb7u5	i386	Determines file type using "magic" numbers
findutils	4.4.2-4	i386	utilities for finding files--find, xargs
fontconfig	2.9.0-7.1	i386	generic font configuration library - support binaries
fontconfig-config	2.9.0-7.1	all	generic font configuration library - configuration
gamin	0.1.10-4.1	i386	File and directory monitoring system
gawk	1:4.0.1+dfsg-2.1	i386	GNU awk, a pattern scanning and processing language
gcc-4.7-base:i386	4.7.2-5	i386	GCC, the GNU Compiler Collection (base package)
geoip-database	20130213-1	all	IP lookup command line tools that use the GeoIP library (country databa
gettext-base	0.18.1.1-9	i386	GNU Internationalization utilities for the base system
gir1.2-glib-2.0	1.32.1-1	i386	Introspection data for GLib, GObject, Gio and GModule
gnupg	1.4.12-7+deb7u6	i386	GNU privacy guard - a free PGP replacement
gpgv	1.4.12-7+deb7u6	i386	GNU privacy guard - signature verification tool
grep	2.12-2	i386	GNU grep, egrep and fgrep
greylistd	0.8.8	all	Greylisting daemon for use with Exim 4
groff-base	1.21-9	i386	GNU troff text-formatting system (base system components)
grub-common	1.99-27+deb7u2	i386	GRand Unified Bootloader (common files)
grub-pc	1.99-27+deb7u2	i386	GRand Unified Bootloader, version 2 (PC/BIOS version)
grub-pc-bin	1.99-27+deb7u2	i386	GRand Unified Bootloader, version 2 (PC/BIOS binaries)
grub2-common	1.99-27+deb7u2	i386	GRand Unified Bootloader (common files for version 2)
gzip	1.5-1.1	i386	GNU compression utilities
heirloom-mailx	12.5-2	i386	feature-rich BSD mail(1)
hostname	3.11	i386	utility to set/show the host name or domain name
ifupdown	0.7.8	i386	high level tools to configure network interfaces
info	4.13a.dfsg.1-10	i386	Standalone GNU Info documentation browser
ingerman	20120607-1	all	New German orthography dictionary for ispell
initramfs-tools	0.109.1	all	generic modular initramfs generator
initscripts	2.88dsf-41+deb7u1	i386	scripts for initializing and shutting down the system
insserv	1.14.0-5	i386	boot sequence organizer using LSB init.d script dependency information
install-info	4.13a.dfsg.1-10	i386	Manage installed documentation in info format
installation-report	2.49	all	system installation report
iproute	20120521-3+b3	i386	networking and traffic control tools
iptables	1.4.14-3.1	i386	administration tools for packet filtering and NAT
iputils-ping	3:20101006-1+b1	i386	Tools to test the reachability of network hosts
isc-dhcp-client	4.2.2.dfsg.1-5+deb70u	i386	ISC DHCP client
isc-dhcp-common	4.2.2.dfsg.1-5+deb70u	i386	common files used by all the isc-dhcp* packages
ispell	3.3.02-6	i386	International Ispell (an interactive spelling corrector)
kbd	1.15.3-9	i386	Linux console font and keytable utilities
keyboard-configuration	1.88	all	system-wide keyboard preferences
klibc-utils	2.0.1-3.1	i386	small utilities built with klibc for early boot
kmod	9-3	i386	tools for managing Linux kernel modules
krb5-locales	1.10.1+dfsg-5+deb7u2	all	Internationalization support for MIT Kerberos
laptop-detect	0.13.7	i386	attempt to detect a laptop
libacl1:i386	2.2.51-8	i386	Access control list shared library
libaio1:i386	0.3.109-3	i386	Linux kernel AIO access library - shared library
libapt-inst1.5:i386	0.9.7.9+deb7u6	i386	deb package format runtime library
libapt-pkg-perl	0.1.26+b1	i386	Perl interface to libapt-pkg
libapt-pkg4.12:i386	0.9.7.9+deb7u6	i386	package managment runtime library

libaspell15	0.60.7~20110707-1	i386	GNU Aspell spell-checker runtime library
libasprintf0c2:i386	0.18.1.1-9	i386	GNU library to use fprintf and friends in C++
libattr1:i386	1:2.4.46-8	i386	Extended attribute shared library
libblkid1:i386	2.20.1-5.3	i386	block device id library
libboost-iostreams1.49.0	1.49.0-3.2	i386	Boost.Iostreams Library
libbsd0:i386	0.4.2-1	i386	utility functions from BSD systems - shared library
libbz2-1.0:i386	1.0.6-4	i386	high-quality block-sorting file compressor library - runtime
libc-bin	2.13-38+deb7u6	i386	Embedded GNU C Library: Binaries
libc6:i386	2.13-38+deb7u6	i386	Embedded GNU C Library: Shared libraries
libc6-i686:i386	2.13-38+deb7u6	i386	Embedded GNU C Library: Shared libraries [i686 optimized]
libcairo2:i386	1.12.2-3	i386	The Cairo 2D vector graphics library
libcap2:i386	1:2.22-1.2	i386	support for getting/setting POSIX.1e capabilities
libcgi-fast-perl	5.14.2-21+deb7u2	all	CGI::Fast Perl module
libclass-isa-perl	0.36-3	all	report the search path for a class's ISA tree
libclass-load-perl	0.17-1	all	module for loading modules by name
libcomerr2:i386	1.42.5-1.1	i386	common error description library
libcurl3:i386	7.26.0-1+wheezy10	i386	easy-to-use client-side URL transfer library (OpenSSL flavour)
libcwidget3	0.5.16-3.4	i386	high-level terminal interface library for C++ (runtime files)
libdata-optlist-perl	0.107-1	all	module to parse and validate simple name/value option pairs
libdate-manip-perl	6.32-1	all	module for manipulating dates
libdatrie1:i386	0.2.5-3	i386	Double-array trie library
libdb5.1:i386	5.1.29-5	i386	Berkeley v5.1 Database Libraries [runtime]
libdbd-mysql-perl	4.021-1+b1	i386	Perl5 database interface to the MySQL database
libdbi-perl	1.622-1+deb7u1	i386	Perl Database Interface (DBI)
libdbi1	0.8.4-6	i386	DB Independent Abstraction Layer for C -- shared library
libdbus-1-3:i386	1.6.8-1+deb7u4	i386	simple interprocess messaging system (library)
libdbus-glib-1-2:i386	0.100.2-1	i386	simple interprocess messaging system (GLib-based shared library)
libdevmapper1.02.1:i386	2:1.02.74-8	i386	Linux Kernel Device Mapper userspace library
libdiscover2	2.1.2-5.2	i386	hardware identification library
libedit2:i386	2.11-20080614-5	i386	BSD editline and history libraries
libept1.4.12	1.0.9	i386	High-level library for managing Debian package information
libevent-2.0-5:i386	2.0.19-stable-3	i386	Asynchronous event notification library
libexpat1:i386	2.1.0-1+deb7u1	i386	XML parsing C library - runtime library
libfam0	2.7.0-17	i386	Client library to control the FAM daemon
libfcgi-perl	0.74-1+b1	i386	helper module for FastCGI
libffi5:i386	3.0.10-3	i386	Foreign Function Interface library runtime
libfile-copy-recursive-perl	0.38-1	all	Perl extension for recursively copying files and directories
libfontconfig1:i386	2.9.0-7.1	i386	generic font configuration library - runtime
libfreetype6:i386	2.4.9-1.1	i386	FreeType 2 font engine, shared library files
libfuse2:i386	2.9.0-2+deb7u1	i386	Filesystem in Userspace (library)
libgamin0	0.1.10-4.1	i386	Client library for the gamin file and directory monitoring system
libgcc1:i386	1:4.7.2-5	i386	GCC support library
libgcrypt11:i386	1.5.0-5+deb7u1	i386	LGPL Crypto library - runtime library
libgd2-noxpm:i386	2.0.36~rc1~dfsg-6.1	i386	GD Graphics Library version 2 (without XPM support)
libgdbm3:i386	1.8.3-11	i386	GNU dbm database routines (runtime version)
libgeoip1	1.4.8+dfsg-3	i386	non-DNS IP-to-country resolver library
libgirepository-1.0-1	1.32.1-1	i386	Library for handling GObject introspection data (runtime library)
libglib2.0-0:i386	2.33.12+really2.32.4-	i386	GLib library of C routines
libglib2.0-data	2.33.12+really2.32.4-	all	Common files for GLib library
libgmp10:i386	2:5.0.5+dfsg-2	i386	Multiprecision arithmetic library
libgnutls26:i386	2.12.20-8+deb7u2	i386	GNU TLS library - runtime library
libgpg-error0:i386	1.10-3.1	i386	library for common error values and messages in GnuPG components
libgssapi-krb5-2:i386	1.10.1+dfsg-5+deb7u2	i386	MIT Kerberos runtime libraries - krb5 GSS-API Mechanism
libhtml-template-perl	2.91-1	all	module for using HTML templates with Perl
libidn11:i386	1.25-2	i386	GNU Libidn library, implementation of IETF IDN specifications
libio-multiplex-perl	1.13-1	all	object-oriented interface to select() for Perl
libio-socket-inet6-perl	2.69-2	all	object interface for AF_INET6 domain sockets
libipc-shareable-perl	0.60-8	all	module to access IPC shared memory segments through perl
libjpeg8:i386	8d-1+deb7u1	i386	Independent JPEG Group's JPEG runtime library
libk5crypto3:i386	1.10.1+dfsg-5+deb7u2	i386	MIT Kerberos runtime libraries - Crypto Library
libkeyutils1:i386	1.5.5-3+deb7u1	i386	Linux Key Management Utilities (library)
libklibc	2.0.1-3.1	i386	minimal libc subset for use with initramfs
libkmod2:i386	9-3	i386	libkmod shared library
libkrb5-3:i386	1.10.1+dfsg-5+deb7u2	i386	MIT Kerberos runtime libraries
libkrb5support0:i386	1.10.1+dfsg-5+deb7u2	i386	MIT Kerberos runtime libraries - Support library
liblcms1:i386	1.19.dfsg-1.2	i386	Little CMS color management library
libldap-2.4-2:i386	2.4.31-1+nmu2	i386	OpenLDAP libraries
liblist-moreutils-perl	0.33-1+b1	i386	Perl module with additional list functions not found in List::Util
liblocale-gettext-perl	1.05-7+b1	i386	module using libc functions for internationalization in Perl
liblockfile-bin	1.09-5	i386	support binaries for and cli utilities based on liblockfile
liblockfile1:i386	1.09-5	i386	NFS-safe locking library
liblog-dispatch-perl	2.32-1	all	message dispatcher to multiple Log::Dispatch::* objects
liblog-log4perl-perl	1.29-1	all	A Perl port of the widely popular log4j logging package.
libltdl7:i386	2.4.2-1.1	i386	A system independent dlopen wrapper for GNU libtool
liblzma5:i386	5.1.1alpha+20120614-2	i386	XZ-format compression library
libmagic1:i386	5.11-2+deb7u5	i386	File type determination library using "magic" numbers
libmcrypt4	2.5.8-3.1	i386	De-/Encryption Library
libmodule-implementation-perl	0.06-1	all	module for loading one of several alternate implementations of a module
libmodule-runtime-perl	0.013-1	all	Perl module for runtime module handling
libmount1	2.20.1-5.3	i386	block device id library
libmysqlclient18:i386	5.6.19-1~dotdeb.1	i386	MySQL database client library
libncurses5:i386	5.9-10	i386	shared libraries for terminal handling
libncursesw5:i386	5.9-10	i386	shared libraries for terminal handling (wide character support)
libnet-cidr-perl	0.15-1	all	Manipulate IPv4/IPv6 netblocks in CIDR notation
libnet-server-perl	2.006-1+deb7u1	all	extensible, general perl server engine
libnet-snmp-perl	6.0.1-2	all	Script SNMP connections
libnewt0.52	0.52.14-11.1	i386	Not Erik's Windowing Toolkit - text mode windowing with slang
libnfnetlink0	1.0.0-1.1	i386	Netfilter netlink library
libonig2	5.9.1-1	i386	Oniguruma regular expressions library
libossp-uuid16	1.6.2-1.3	i386	OSSP uuid ISO-C and C++ - shared library
libp11-kit0:i386	0.12-3	i386	Library for loading and coordinating access to PKCS#11 modules - runtim
libpackage-deprecationmanager-per	0.13-1	all	module for managing deprecation warnings for Perl distributions
libpackage-stash-perl	0.33-1	all	module providing routines for manipulating stashes
libpackage-stash-xs-perl	0.24-1+b1	i386	Perl module providing routines for manipulating stashes (XS version)
libpam-modules:i386	1.1.3-7.1	i386	Pluggable Authentication Modules for PAM
libpam-modules-bin	1.1.3-7.1	i386	Pluggable Authentication Modules for PAM - helper binaries
libpam-runtime	1.1.3-7.1	all	Runtime support for the PAM library
libpam0g:i386	1.1.3-7.1	i386	Pluggable Authentication Modules library
libpango1.0-0:i386	1.30.0-1	i386	Layout and rendering of internationalized text
libparams-classify-perl	0.013-4	i386	Perl module for argument type classification
libparams-util-perl	1.07-1	i386	Perl extension for simple stand-alone param checking functions
libparams-validate-perl	1.06-1	i386	Perl module to validate parameters to Perl method/function calls
libpci3:i386	1:3.1.9-6	i386	Linux PCI Utilities (shared library)
libpcre3:i386	1:8.30-5	i386	Perl 5 Compatible Regular Expression Library - runtime files
libperl4-corelibs-perl	0.003-1	all	libraries historically supplied with Perl 4
libpipeline1:i386	1.2.1-1	i386	pipeline manipulation library
libpixman-1-0:i386	0.26.0-4+deb7u1	i386	pixel-manipulation library for X and cairo
libpng12-0:i386	1.2.49-1	i386	PNG library - runtime
libpopt0:i386	1.16-7	i386	lib for parsing cmdline parameters
libprocps0:i386	1:3.3.3-3	i386	library for accessing process information from /proc
libqdbm14	1.8.78-2	i386	QDBM Database Libraries without GDBM wrapper[runtime]
libreadline6:i386	6.2+dfsg-0.1	i386	GNU readline and history libraries, run-time libraries
librrd4	1.4.7-2	i386	time-series data storage and display system (runtime library)
librrds-perl	1.4.7-2	i386	time-series data storage and display system (Perl interface, shared)
librtmp0:i386	2.4+20111222.git4e06e	i386	toolkit for RTMP streams (shared library)
libsasl2-2:i386	2.1.25.dfsg1-6+deb7u1	i386	Cyrus SASL - authentication abstraction library
libsasl2-modules:i386	2.1.25.dfsg1-6+deb7u1	i386	Cyrus SASL - pluggable authentication modules
libselinux1:i386	2.1.9-5	i386	SELinux runtime shared libraries
libsemanage-common	2.1.6-6	all	Common files for SELinux policy management libraries
libsemanage1:i386	2.1.6-6	i386	SELinux policy management library
libsepol1:i386	2.1.4-3	i386	SELinux library for manipulating binary security policies
libsigc++-2.0-0c2a:i386	2.2.10-0.2	i386	type-safe Signal Framework for C++ - runtime
libsigsegv2	2.9-4	i386	Library for handling page faults in a portable way
libslang2:i386	2.2.4-15	i386	S-Lang programming library - runtime version

libsocket6-perl	0.23-1+b2	i386	Perl extensions for IPv6
libsqlite3-0:i386	3.7.13-1+deb7u1	i386	SQLite 3 shared library
libss2:i386	1.42.5-1.1	i386	command-line interface parsing library
libssh2-1:i386	1.4.2-1.1	i386	SSH2 client-side library
libssl1.0.0:i386	1.0.1e-2+deb7u13	i386	SSL shared libraries
libstdc++6:i386	4.7.2-5	i386	GNU Standard C++ Library v3
libsub-install-perl	0.926-1	all	module for installing subroutines into packages easily
libswitch-perl	2.16-2	all	switch statement for Perl
libsystemd-login0:i386	44-11+deb7u4	i386	systemd login utility library
libt1-5	5.1.2-3.6	i386	Type 1 font rasterizer library - runtime
libtasn1-3:i386	2.13-2	i386	Manage ASN.1 structures (runtime)
libtext-charwidth-perl	0.04-7+b1	i386	get display widths of characters on the terminal
libtext-iconv-perl	1.7-5	i386	converts between character sets in Perl
libtext-wrapi18n-perl	0.06-7	all	internationalized substitute of Text::Wrap
libthai-data	0.1.18-2	all	Data files for Thai language support library
libthai0:i386	0.1.18-2	i386	Thai language support library
libtinfo5:i386	5.9-10	i386	shared low-level terminfo library for terminal handling
libtry-tiny-perl	0.11-1	all	module providing minimalistic try/catch
libudev0:i386	175-7.2	i386	libudev shared library
liburi-perl	1.60-1	all	module to manipulate and access URI strings
libusb-0.1-4:i386	2:0.1.12-20+nmu1	i386	userspace USB programming library
libusb-1.0-0:i386	2:1.0.11-1	i386	userspace USB programming library
libustr-1.0-1:i386	1.0.4-3	i386	Micro string library: shared library
libuuid-perl	0.02-5	i386	Perl extension for using UUID interfaces as defined in e2fsprogs
libuuid1:i386	2.20.1-5.3	i386	Universally Unique ID library
libvpx1:i386	1.1.0-1	i386	VP8 video codec (shared library)
libwrap0:i386	7.6.q-24	i386	Wietse Venema's TCP wrappers library
libx11-6:i386	2:1.5.0-1+deb7u1	i386	X11 client-side library
libx11-data	2:1.5.0-1+deb7u1	all	X11 client-side library
libxapian22	1.2.12-2	i386	Search engine library
libxau6:i386	1:1.0.7-1	i386	X11 authorisation library
libxcb-render0:i386	1.8.1-2+deb7u1	i386	X C Binding, render extension
libxcb-shm0:i386	1.8.1-2+deb7u1	i386	X C Binding, shm extension
libxcb1:i386	1.8.1-2+deb7u1	i386	X C Binding
libxdmcp6:i386	1:1.1.1-1	i386	X11 Display Manager Control Protocol library
libxext6:i386	2:1.3.1-2+deb7u1	i386	X11 miscellaneous extension library
libxft2:i386	2.3.1-1	i386	FreeType-based font drawing library for X
libxml2:i386	2.8.0+dfsg1-7+wheezy1	i386	GNOME XML library
libxmuu1:i386	2:1.1.1-1	i386	X11 miscellaneous micro-utility library
libxpm4:i386	1:3.5.10-1	i386	X11 pixmap library
libxrender1:i386	1:0.9.7-1+deb7u1	i386	X Rendering Extension client library
libxslt1.1:i386	1.1.26-14.1	i386	XSLT 1.0 processing library - runtime library
libyaml-syck-perl	1.20-1	i386	Perl module providing a fast, lightweight YAML loader and dumper
linux-base	3.5	all	Linux image base package
linux-image-3.2.0-4-686-pae	3.2.63-2	i386	Linux 3.2 for modern PCs
linux-image-686-pae	3.2+46	i386	Linux for modern PCs (meta-package)
locales	2.13-38+deb7u6	all	Embedded GNU C Library: National Language (locale) data [support]
login	1:4.1.5.1-1	i386	system login tools
logrotate	3.8.1-4	i386	Log rotation utility
logwatch	7.4.0+svn20120502rev1	all	log analyser with nice output written in Perl
lsb-base	4.1+Debian8+deb7u1	all	Linux Standard Base 4.1 init script functionality
lsof	4.86+dfsg-1	i386	Utility to list open files
man-db	2.6.2-1	i386	on-line manual pager
manpages	3.44-1	all	Manual pages about using a GNU/Linux system
manpages-de	1.2-1	all	German manpages
mawk	1.3.3-17	i386	a pattern scanning and text processing language
mime-support	3.52-1	all	MIME files 'mime.types' & 'mailcap', and support programs
module-init-tools	9-3	all	transitional dummy package (module-init-tools to kmod)
mount	2.20.1-5.3	i386	Tools for mounting and manipulating filesystems
multiarch-support	2.13-38+deb7u6	i386	Transitional package to ensure multiarch compatibility
munin	2.0.6-4+deb7u2	all	network-wide graphing framework (grapher/gatherer)
munin-common	2.0.6-4+deb7u2	all	network-wide graphing framework (common)
munin-doc	2.0.6-4+deb7u2	all	network-wide graphing framework (documentation)
munin-node	2.0.6-4+deb7u2	all	network-wide graphing framework (node)
munin-plugins-core	2.0.6-4+deb7u2	all	network-wide graphing framework (plugins for node)
munin-plugins-extra	2.0.6-4+deb7u2	all	network-wide graphing framework (user contributed plugins for node)
mysql-client-5.6	5.6.19-1~dotdeb.1	i386	MySQL database client binaries
mysql-client-core-5.6	5.6.19-1~dotdeb.1	i386	MySQL database core client binaries
mysql-common	5.6.19-1~dotdeb.1	all	MySQL database common files, e.g. /etc/mysql/my.cnf
mysql-server	5.6.19-1~dotdeb.1	all	MySQL database server (metapackage depending on the latest version)
mysql-server-5.6	5.6.19-1~dotdeb.1	i386	MySQL database server binaries and system database setup
mysql-server-core-5.6	5.6.19-1~dotdeb.1	i386	MySQL database server binaries
nano	2.2.6-1+b1	i386	small, friendly text editor inspired by Pico
ncurses-base	5.9-10	all	basic terminal type definitions
ncurses-bin	5.9-10	i386	terminal-related programs and man pages
ncurses-term	5.9-10	all	additional terminal type definitions
net-tools	1.60-24.2	i386	The NET-3 networking toolkit
netbase	5.0	all	Basic TCP/IP networking system
netcat-traditional	1.10-40	i386	TCP/IP swiss army knife
nginx	1.6.2-1~dotdeb.1	all	small, powerful, scalable web/proxy server
nginx-common	1.6.2-1~dotdeb.1	all	small, powerful, scalable web/proxy server - common files
nginx-full	1.6.2-1~dotdeb.1	i386	nginx web/proxy server (standard version)
openbsd-inetd	0.20091229-2	i386	OpenBSD Internet Superserver
openssh-blacklist	0.4.1+nmu1	all	list of default blacklisted OpenSSH RSA and DSA keys
openssh-blacklist-extra	0.4.1+nmu1	all	list of non-default blacklisted OpenSSH RSA and DSA keys
openssh-client	1:6.0p1-4+deb7u2	i386	secure shell (SSH) client, for secure access to remote machines
openssh-server	1:6.0p1-4+deb7u2	i386	secure shell (SSH) server, for secure access from remote machines
openssl	1.0.1e-2+deb7u13	i386	Secure Socket Layer (SSL) binary and related cryptographic tools
os-prober	1.58	i386	utility to detect other OSes on a set of drives
passwd	1:4.1.5.1-1	i386	change and administer password and group data
pciutils	1:3.1.9-6	i386	Linux PCI Utilities
perl	5.14.2-21+deb7u2	i386	Larry Wall's Practical Extraction and Report Language
perl-base	5.14.2-21+deb7u2	i386	minimal Perl system
perl-modules	5.14.2-21+deb7u2	all	Core Perl modules
php5-cli	5.6.2-1~dotdeb.1	i386	command-line interpreter for the php5 scripting language
php5-common	5.6.2-1~dotdeb.1	i386	Common files for packages built from the php5 source
php5-curl	5.6.2-1~dotdeb.1	i386	CURL module for php5
php5-fpm	5.6.2-1~dotdeb.1	i386	server-side, HTML-embedded scripting language (FPM-CGI binary)
php5-gd	5.6.2-1~dotdeb.1	i386	GD module for php5
php5-mcrypt	5.6.2-1~dotdeb.1	i386	MCrypt module for php5
php5-mysqlnd	5.6.2-1~dotdeb.1	i386	MySQL module for php5 (Native Driver)
php5-readline	5.6.2-1~dotdeb.1	i386	Readline module for php5
php5-sqlite	5.6.2-1~dotdeb.1	i386	SQLite module for php5
phpmyadmin	4:3.4.11.1-2+deb7u1	all	MySQL web administration tool
procps	1:3.3.3-3	i386	/proc file system utilities
psmisc	22.19-1+deb7u1	i386	utilities that use the proc file system
pure-ftpd	1.0.36-1.1	i386	Secure and efficient FTP server
pure-ftpd-common	1.0.36-1.1	all	Pure-FTPd FTP server (Common Files)
python	2.7.3-4+deb7u1	all	interactive high-level object-oriented language (default version)
python-beautifulsoup	3.2.1-1	all	error-tolerant HTML parser for Python
python-catcher	0.1.7	all	Beautiful stack traces for Python
python-central	0.6.17	all	register and build utility for Python packages
python-chardet	2.0.1-2	all	universal character encoding detector
python-crypto	2.6-4+deb7u3	i386	cryptographic algorithms and protocols for Python
python-daemon	1.5.5-1	all	library for making a Unix daemon process
python-dbus	1.1.1-1	i386	simple interprocess messaging system (Python interface)
python-dbus-dev	1.1.1-1	all	main loop integration development files for python-dbus
python-exconsole	0.1.5	all	Emergency/postmortem Python console
python-gamin	0.1.10-4.1	i386	Python binding for the gamin client library
python-gevent	0.13.6-1+nmu3	i386	gevent is a coroutine-based Python networking library
python-gevent-socketio	0.3.6-1	all	SocketIO server based on the Gevent pywsgi server, a Python
python-gevent-websocket	0.3.6-1	all	Websocket handler for the gevent pywsgi server, a Python net
python-gi	3.2.2-2	i386	Python 2.x bindings for gobject-introspection libraries
python-greenlet	0.4.0-1	i386	Lightweight in-process concurrent programming

python-imaging	1.1.7-4+deb7u1	i386	Python Imaging Library
python-ldap	2.4.10-1	i386	LDAP interface module for Python
python-lockfile	1:0.8-2	all	file locking library for Python
python-lxml	2.3.2-1+deb7u1	i386	pythonic binding for the libxml2 and libxslt libraries
python-mako	0.7.0-1.1	all	fast and lightweight templating for the Python platform
python-markupsafe	0.15-1	i386	XML/HTML/XHTML Markup safe string for Python
python-medusa	1:0.5.4-7	all	Framework for implementing asynchronous servers
python-meld3	0.6.5-3.1	i386	An HTML/XML templating system for Python
python-minimal	2.7.3-4+deb7u1	all	minimal subset of the Python language (default version)
python-oauthlib	0.1.2-1	all	generic, spec-compliant implementation of OAuth for Python
python-passlib	1.5.3-2	all	comprehensive password hashing framework
python-pkg-resources	0.6.24-1	all	Package Discovery and Resource Access using pkg_resources
python-psutil	0.6.1-2	i386	module providing convenience functions for managing processes
python-reconfigure	0.1.62	all	Simple config file management library
python-requests	0.12.1-1	all	elegant and simple HTTP library for Python, built for human beings
python-six	1.1.0-2	all	Python 2 and 3 compatibility library (Python 2 interface)
python-support	1.0.15	all	automated rebuilding support for Python modules
python2.7	2.7.3-6+deb7u2	i386	Interactive high-level object-oriented language (version 2.7)
python2.7-minimal	2.7.3-6+deb7u2	i386	Minimal subset of the Python language (version 2.7)
readline-common	6.2+dfsg-0.1	all	GNU readline and history libraries, common files
rrdtool	1.4.7-2	i386	time-series data storage and display system (programs)
rsync	3.0.9-4	i386	fast, versatile, remote (and local) file-copying tool
rsyslog	5.8.11-3+deb7u2	i386	reliable system and kernel logging daemon
sed	4.2.1-10	i386	The GNU sed stream editor
sensible-utils	0.0.7	all	Utilities for sensible alternative selection
sgml-base	1.26+nmu4	all	SGML infrastructure and SGML catalog file support
shared-mime-info	1.0-1+b1	i386	FreeDesktop.org shared MIME database and spec
supervisor	3.0a8-1.1+deb7u1	all	A system for controlling process state
sysv-rc	2.88dsf-41+deb7u1	all	System-V-like runlevel change mechanism
sysvinit	2.88dsf-41+deb7u1	i386	System-V-like init utilities
sysvinit-utils	2.88dsf-41+deb7u1	i386	System-V-like utilities
tar	1.26+dfsg-0.1	i386	GNU version of the tar archiving utility
task-german	3.14.1	all	German environment
tasksel	3.14.1	all	Tool for selecting tasks for installation on Debian systems
tasksel-data	3.14.1	all	Official tasks used for installation of Debian systems
tcl8.5	8.5.11-2	i386	Tcl (the Tool Command Language) v8.5 - run-time files
tcpd	7.6.q-24	i386	Wietse Venema's TCP wrapper utilities
traceroute	1:2.0.18-3	i386	Traces the route taken by packets over an IPv4/IPv6 network
ttf-dejavu	2.33-3	all	Metapackage to pull in ttf-dejavu-core and ttf-dejavu-extra
ttf-dejavu-core	2.33-3	all	Vera font family derivate with additional characters
ttf-dejavu-extra	2.33-3	all	Vera font family derivate with additional characters
tzdata	2014h-0wheezy1	all	time zone and daylight-saving time data
ucf	3.0025+nmu3	all	Update Configuration File: preserve user changes to config files.
udev	175-7.2	i386	/dev/ and hotplug management daemon
unzip	6.0-8	i386	De-archiver for .zip files
update-inetd	4.43	all	inetd configuration file updater
usbutils	1:005-3	i386	Linux USB utilities
util-linux	2.20.1-5.3	i386	Miscellaneous system utilities
vim-common	2:7.3.547-7	i386	Vi IMproved - Common files
vim-tiny	2:7.3.547-7	i386	Vi IMproved - enhanced vi editor - compact version
wget	1.13.4-3+deb7u1	i386	retrieves files from the web
whiptail	0.52.14-11.1	i386	Displays user-friendly dialog boxes from shell scripts
whois	5.1.1~deb7u1	i386	intelligent WHOIS client
wngerman	20120607-1	all	New German orthography wordlist
xauth	1:1.0.7-1	i386	X authentication utility
xkb-data	2.5.1-3	all	X Keyboard Extension (XKB) configuration data
xml-core	0.13+nmu2	all	XML infrastructure and XML catalog file support
xz-utils	5.1.1alpha+20120614-2	i386	XZ-format compression utilities
zip	3.0-6	i386	Archiver for .zip files