Albrecht Weinert

Ubuntu for remote services

Abstract and Introduction

This technical report is about installing Ubuntu 16.04.1 on a server for remote services. Extending this a bit, we have (Aug. 2017) three distributions running on five target machines.

The Ubuntu **distributions** were installed by DVDs burned from downloaded images
ubuntu-16.04.1-desktop-amd64.iso [18.10.2016 1.513.308.160]
ubuntu-16.04.1-server-amd64.iso [07.11.2016 699.400.192]
Additionally, we had Mint 18 (magazine DVD) [is based on Ubuntu 16.04]

The second distribution, the server one, comes with no graphical HMI and without any comfort.

The five **targets** were:

- A very old **Fujitsu-Siemens Lifebook E8110** laptop, we put the Mint 18 on.
- A very new **Lenovo Yoga 260** laptop with Ubuntu 64bit desktop.
- A middle aged "real" server **Fujitsu RX200S5** with two LSI MegaRaid extenders, we tried the Ubuntu 64bit desktop first *) and then the 64bit server distro.
- An old "real" server **Fujitsu Siemens RX300S3** with one LSI MegaRaid extender with the server distro.
- An pre-installed Ubuntu 16.04 **virtual server** rented from an external provider.

Note *): Originally, an important secondary goal was to have not only the same OS distribution but also the same look and feel on both workstations and servers. Nowadays or since at least 20 years one is entitled to find a decent HMI on servers, too, both remote and on site. And that isn't only expected cause we have this with Windows. Therefore we report quite a bit on the workstation installation, too. And that's why we used the workstation distribution for the RX200S5 server in the first run, too – which met with no success.

Goal

Even though having four target types, the main topic here is commissioning the server.
Or to be more precise, have an unattended 24/7 server in an inaccessible location.
The server's tasks in the end will be:

- Multi domain web server,
- SVN server,
- file services (FTP mandatory, may be DAV),
- some PHP tricks and some mostly Java based server automatisms (hooks) etc.
- .. and some more features.

These services and access to the OS should be based on an uniform Id/user/groups management.

All this and more we have running for many years on Windows servers and AD domains. Hence the easy way would be sticking with that approach for new installations as well as for migrating Windows Server 2003 installations essential for customers. Shamefully, the 2003 cannot be kept alive any longer. Looking for Ubuntu as an alternative is for cost and independence. Let's see how far we get there.

On the other hand: Ubuntu's badly implemented or not existing GUIs, loosing icons and all control after updates, graphics drivers not adapting to standard resolutions, no uniform clipboard handling, no clipboard with xRDP, some total LDAP server failures (freshly installed) … and much more else may call for for other distributions or ... for Windows. But would that be better in the long run?

Why Ubuntu

Looking server manufacturer's support, Ubuntu seems not the first choice. On the other hand on **rented server**s the Ubuntu variant is often the cheapest.

Hence, if Ubuntu could be deployed with all services and kept under control, and if one can cope with the big lot of intrinsic limitations, this would be paying.

Install the OS stand alone

On a decent server or PC/workstation insert the appropriate Linux DVD – and press reset.

This always went well on our **Fujitsu RX200S5**, originally having had a Windows Server 2008 R2 to be replaced without leaving residues and on the **Fujitsu Siemens RX300S3** having been an Windows 2003 file server and AD domain member, to be replaced in just this role.

On the dark side are machines with UEFI and/or a gladitorial pre-installed vendor OS. Start fighting and try not to go nuts by a constantly resurrecting W10 as was the case on our **Lenovo Yoga 260**.

Standard distribution on Yoga 260 – Resume

The downside, as said, was looping through some 50 restarts, off/ons with ever recurring BIOS/UEFI settings and the vendor installed W10. In the end we got rid of all the pre-installed trash and have just Ubuntu alone on the machine.

But, alas, we do not know which setting or intentional power down event finally made the race. It has, most probably, been a multi-step process involving a Windows 10 professional installation with complete formatting, which may have erased most of the vendor's special gifts.

Thereafter, to the stand-alone Ubuntu 16.04 we added:

- Oracle Java8,
- SVN, rabitSVN (Linux' tortoiseSVN),
- Eclipse with Java and SVN,
- Thunderbird + settings of some dozen mail accounts and calendars transferred from Windows,
- TeXWork, ● Chrome, ● Remmina (a RDP client), ● FileZilla, ● ownCloud as well as
- all available WLANs and their different settings at/for various locations.

We still look for an real equivalent of IrfanView (doing all but not more). We will replace LibreOffice (pre-installed) with OpenOffice as incompatibilities to other installations arose and add MS-Fonts.

Touchscreen and pen weren't much used – neither here nor on on an identical Yoga machine with Windows 10 professional. But, with no extra installation or configuration, pen and touchscreen do work with Ubuntu, too.

On the W10 twin, mirroring the screen on a low resolution beamer via HDMI is just plug in:
The display stays unspoiled in high resolution and the beamer gets the the same picture automatically scaled down in perfect (beamer's best) quality.

On Ubuntu, automatically, one gets small parts of the display on the screen / beamer:
So nothing is readable. One is forced to "play" many rounds with xrandr to get at best a result just readable on the screen. The desktop always gets spoiled. Neither the audience nor the operator/lecturer get a satisfying result – to put it mildly. When W10, with exactly the same hardware, gets the optimal result by just plugging in the beamer, question is why Ubuntu is having big problems.

To sum up, apart from presenting on a lower resolution beamer, Ubuntu on a powerful laptop may be used for writing, developing, remote administration and much else having been done so far on an equivalent Windows machine.

Mint 18 on an old Fujitsu Siemens Laptop – Resume

An old Laptop with all programmes just mentioned had an older Mint, featuring a bit more comfort and "Windows feeling" than pure Ubuntu. Like it or not, we wanted to give Ubuntu 16.04 a try on this platform, too, by upgrading to Mint18. As usual with Mint, the distribution upgrade crashed and it was a complete re-install in the end. Well, Mint 18 is nicer, but much, much slower.

Do not use it on old/low end hardware – its unusable there.

Considering the the successful working with pure Ubuntu (not regarding the shameful beamer performance) we see no reason to wrap Ubuntu further.

Standard distribution on RX200S5 – Resume

To be clear: We talk about "workstation = non server" distribution on a "real" server machine.

The good point is: The installation works and it is partly able to render contemporary HMI comfort.

But there's quite a list of what we didn't get to work:

The two RAID extender racks, ten times as large as the server itself, were not recognised by this Ubuntu distribution. They might be gotten to work by trying the one or other LSI MegaRAID Linux driver packages – none of which googled really seemed to fit the hardware and the OS distro.

Quite annoying was not being able the chose a decent resolution and screen position for the 24" monitor attached to RX200's front VGA for on site administration. We didn't fiddle to long with this bug as the server is mostly not meant for non remote access. The server room is too cold for humans. Graphics / graphic drivers are not Ubuntu's strong point, on any hardware.

And what never worked was LDAP (openLDAP). Freshly installed after the book the manager/ admin user, just born, could do nothing or wasn't existing. No trick or repair found in the net, the manuals or guidelines worked with 16.04 on any of our target systems.

After not having the RAID extenders, this LDAP failure, in the end, was the point were to give up this standard/non server distribution. Ironically, the server distro was no way better with LDAP.

Server distribution on RX200S5 – Resume

The work with the server distribution (now in 24/7 use in several installations) is much harder – and not only cause of stone age HMI. Remote with putty works with this distribution from start (when having ticked SSL-server). That, at least, we expected.

The two LSI MegaRAID extenders were recognised and do work with all previous files (Windows, NTFS). That's simply great! No modern Windows server distro did that! Keeping or migrating access rights and ownership from an existing AD domain for NTFS trees to keep will be a problem. No problem was re-formatting one MegaRAID (logic) drive to ext4.

The Ubuntu server distribution won't automatically handle nor recognise the server's four LAN ports, three of them in use in different LANs. Here the WS/PC distribution hadn't any problem, all three of four interfaces with outside connections just worked in the WS distro. This unexpected failure took many rounds of re-installation to get one of them ready. Here slightest configuration errors hinder the boot or crashes the installation process (thanks to grayed out or missing backward buttons). The rationale for not using or offering the information so obviously available by making the server distribution more stupid than the standard one, is not comprehensible.

As expected the server's front VGA for admin has no GUI. And the "terminal display" on the 24" monitor is horrible considering height, width, resolution, no scroll, no clipboard etc. Having just a terminal to start with one is forced to tolerate on all or most Linux server distributions.

> But this display is a shame. Hence, use putty, only!
> And work only on a remote system with decent graphics and clipboard support!

Here putty worked without further installation – at least as soon as one of the LANs is conquered. Could one assume it being better than with the standard distro and not spoiling anything else, one would say go for a graphical HMI and use RDP.

> At present we don't say so. neither directly nor by Docker ([30]). Stay in the sixties!

The worst point, having caused delays (and costs), is openLDAP displaying the same problems and bugs as with the standard distribution.

Server distribution on a rented virtual server – Resume

Besides finding the minimal OS pre-installed and "putty ready" almost all experiences and installation procedures with the "real" servers can be successfully re-used. A bit unexpectedly, the provider would charge extra for each backup (monthly and expensive). So, it seems a good tactic only to bring in things operationally well-tried on a real local machine – and save all beyond the basic installation outside the landlord's site on other machines.

Server distribution on RX300S3 – Resume

The Ubuntu server installation on older Siemens-Fujitsu servers went as well as with younger Fujitsu (~~Siemens~~) ones. Fiddling with two LAN ports was worse as with the younger one's four. All basic installations – putty server, ftp server etc. pp. – work.

Keeping its role as AD member and file server with NTFS (files, users, rights; W2003) intact is a goal. This work has succeeded in between and is reported in (supplement) [28].

And by the way: On this older and smaller one server with only one RAID extender re-boot is ages faster than with the RX200S5.

On the content

In **Part I** we describe
installations, procedures, recipes applicable to all above distros and targets

Part II deals just with the running server installations, our main goal.

In the course of the work with all distributions and targets reported here on, many mistakes were made and detours gone and as it turned out not only by us. Hence those detours and remedies may contain valuable information and were reported here, too until March 2017.

On the other hand these **Paradise lost** sections made this document [29] even larger and less readable. Instead of just deleting them they were evacuated to a supplement report [26].

Find References, Abbreviations, a collection of useful commands and the Table of Contents
in the **Appendix**.

Using names

Names used here [27..29] are not fictitious. This helps bringing real and working examples of commands, files, outputs etc. to you without errors introduced by obfuscating.
`albrecht` is an example for an Ubuntu group and user (being in sudo). `PD321S` is the name of the Fujitsu RX200S5 server, in AD domain `fb3-meva.fh-bochum.de` as long as it has been a Windows server. In the two most relevant LANs its addresses, `192.168.89.6` and `193.175.155.6`, also held in DNSs and (domain) DHCP were kept. For the Fujitsu-Siemens RX300S3 we have PD337S, `192.168.89.15` and `193.175.155.15` in the examples. PD337S has been kept domain member and doing its former (file server) duties.

When seeing those or other names, IP-addresses and values "hard-coded" in following descriptions, configurations and code, always replace them accordingly.

PART I

Before using putty (hen and egg)

This has to be done on all non server distributions. On most (not all) server installations this will be ready with the basic (CD) installation with the right ticks. Otherwise locally/on site (i.e. in the cold server room) do

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

Using remote shell (with putty)

On the Windows remote system:

Install putty by getting putty.exe and moving it to a directory in the path – best C:\util\.

Start putty.exe; connect to the Ubuntu server and set an acceptable layout, font, colours etc. Save these settings under a session name, like PD321S (best use the servers name).

Afterwards call putty by

```
C:\util\putty.exe -load PD321S
C:\util\putty.exe -load PD321S -l albrecht
```

(second form has pre-set Ubuntu user) and best make an icon carrying that command.

On an Ubuntu WS/laptop do:

```
sudo apt install putty
```

Run it and make an acceptable layout, fonts and else setting. Save it as e.g. pd321s, which will create a file ~/.putty/sessions/pd321s. Those files can be copied and edited. Use it by

```
putty -load pd321s
```

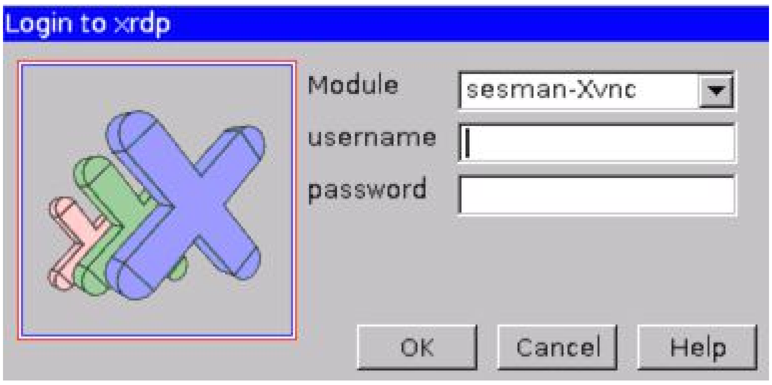
and make an icon doing that. Disappointing: putty on a Ubuntu WS does not support clipboard at all or in the n + 21th surprising variant making cross copying between various applications a nightmare. Here you are just much better off on a Windows machine – it has a perfect cross application clipboard and only two annoyingly different variants of copy/paste shortcuts.

"putty recommends Windows"

If thrown off regularly, something like 1721 (s) as keep alive value in settings might perhaps work.

Using RDP with Ubuntu server

This is a detour or at least less recommendable. If interested read supplement [26]. Then on your Windows client, start RDP client as usual just using the server's DNS name or IP – not minding it being Ubuntu. On the login frame use sesman-Xvnc + your Ubuntu name and password; ref. Fig.1.


Login to xrdp

Module: sesman-Xvnc

username:

password:

Fig. 1: xRDP-Login
username can be pre-filled using a .rdp configuration file accordingly

Install Java using (remote) shell, only

This was done successfully on all our machines and distros (including Mint).

Java, best with Frame4J and its tools, is always beneficial. Some tools and services won't work without, anyway. So let's start with it, and have a full remote (putty) installation example:

```
mkdir Downloads
cd    Downloads
```

```
wget --no-check-certificate --no-cookies  --header "Cookie:
oraclelicense=accept-securebackup-cookie"
http://download.oracle.com/otn-pub/java/jdk/8u102-b14/jdk-8u102-
linux-x64.tar.gz   ## four lines are one
```

```
sudo mkdir -p /usr/local/java
sudo cp -r jdk-8u102-linux-x64.tar.gz   /usr/local/java/
cd    /usr/local/java
sudo tar xvzf jdk-8u102-linux-x64.tar.gz
sudo mv jdk1.8.0_102 jdk
```

Now put the following at the end of /etc/profile:

```
JAVA_HOME=/usr/local/java/jdk
PATH=$PATH:$JAVA_HOME/bin
export JAVA_HOME
export PATH
```

After the change logout/login and test it by java -version. On success get and verify Frame4J:

```
cd  ~/Downloads
wget http://weinert-automation.de/software/frame4j/frame4j.jar
sudo cp -r frame4j.jar  /usr/local/java/jdk/jre/lib/ext/
java ShowProps
java AskAlert
```

If none works and you get different rights the those of funny uucp:143 to be checked by

```
dir /usr/local/java/jdk/jre/lib/ext/
```

```
-rw-r--r-- 1 uucp  143     8286 2016-06-23 01:53 dnsns.jar
-rw------- 1 root root   533138 2016-12-19 11:00 frame4j.jar
```

repair it by just:

```
sudo chmod 644 /usr/local/java/jdk/jre/lib/ext/frame4j.jar
```

AskAlert (with no parameters or -en only) will provide infos on Frame4J's version.
On an installation without GUI or on a remote putty (providing no GUI by nature) it will end in a warning on having no (X11) graphics.

The Frame4J tools will come quite handy on SVN hooks and other server work.

Installing the Java (JDK) documentation on such headless server just to remind logged in users of the lack of GUI and graphical tools to read HTML, pdf etc. seems not making any sense. But you can make it available to clients by simple Apache configuration. On a workstation one should always have the Java documentation locally. For the the basic Java documentation do:

```
wget --no-check-certificate --no-cookies   --header "Cookie:
oraclelicense=accept-securebackup-cookie"
http://download.oracle.com/otn-pub/java/jdk/8u112-b15/jdk-8u112-
docs-all.zip  ## four lines are one
```

```
cd /usr/local/java/jdk
sudo bin/jar xfv ~/Downloads/jdk-8u112-docs-all.zip
cd ~
```

On distributions having a pre-installed java (most non server ones) do not forget to start with:

```
sudo apt-get purge openjdk*
```

And beware of updates or new installations resurrecting openjdk. After such events always check by java -version and repair accordingly.

The following, best adapted to a newer rxtx, will also make ShowPorts run:

```
sudo apt-get install librxtx-java
sudo cp /usr/share/java/RXTXcomm-2.2pre2.jar
/usr/local/java/jdk/jre/lib/ext/
```

```
sudo mkdir /usr/lib64
dir  /usr/share/java/
sudo cp  /usr/lib/jni/librxtxSerial-2.2pre1.so
/usr/lib64/librxtxSerial
```

```
sudo cp  /usr/lib/jni/librxtxParallel-2.2pre1.so
/usr/lib64/librxtxParallel.so
```

```
java ShowPorts
```

Chrome

This was done on all installations having a graphical HMI. Type:

```
cd ~/Downloads
wget https://dl.google.com/linux/direct/google-chrome-
stable_current_amd64.deb ## two liens are one
```

and double click on the file delivered.

MS-Fonts

This was done on all installations having a graphical HMI as well as Office and/or .pdf-tools. For compatible exchange with partners, collogues, customers and oneself on one's Windows machines one cannot dispense with the standard fonts, lest to spoil layouts and having worse effects.

Microsoft approved procedure is:

```
sudo apt-get install ttf-mscorefonts-installer
fc-match arial # just check the success
```

As of now (March 2017) this won't work due to hard-coded wrong download URL. Hence copy your (licensed) C:\Windows\Fonts\ to a stick, and (auto-) mount it on your Ubuntu target. Then:

```
sudo apt-get remove  ttf-mscorefonts-installer
sudo apt autoremove
sudo mkdir /usr/share/fonts/truetype/msttcorefonts
```

```
sudo cp  /media/weinert/16D7-C87E/msttcorefonts/
/usr/share/fonts/truetype/
```

```
dir /usr/share/fonts/truetype/msttcorefonts
fc-match arial # just check the success
```

Now the last check should show the real arial.ttf and no doubtful substitute.

Network interfaces

This was done / had to be done with server distro, only. With one good LAN interface one gets

```
cat  /etc/network/interfaces
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface  MEVA-Net
auto ens1f0
iface ens1f0 inet static
   address 192.168.89.6
   netmask 255.255.255.0
   network 192.168.89.0
   broadcast 192.168.89.255
   gateway 192.168.89.11
   # dns-* options are implemented by the resolvconf package, if installed
   dns-nameservers 192.168.89.11
   dns-search fb3-meva.fh-bochum.de
```

Start adding at the end all further interfaces used (ens1f1, ens2f0):

```
# The secondary network interface  FH-Netz
auto ens1f1
iface ens1f1 inet static
   address 193.175.115.6
   netmask 255.255.255.0
   network 193.175.115.0
   broadcast 193.175.115.255
   #   gateway 193.175.115.1
   dns-nameservers 193.175.112.3 195.37.168.3
   dns-search hs-bochum.de
```

Do comment out # gateway if you have a working one on another interface.

Make changes happen by re-boot or (aborting your remote connections, too) by

```
sudo /etc/init.d/networking restart
```

LS MegaRaid

So far, this was done on Fujitsu servers with server distro, only.

With three variants of lsblk we get all drive infos, including (Windows) labels and UUIDs:

`lsblk`

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0     3T  0 disk
├─sda1   8:1    0   128M  0 part
└─sda2   8:2    0     3T  0 part
sdb      8:16   0   2,7T  0 disk
├─sdb1   8:17   0   128M  0 part
└─sdb2   8:18   0   2,7T  0 part
sdc      8:32   0 407,9G  0 disk
├─sdc1   8:33   0 375,9G  0 part /
├─sdc2   8:34   0     1K  0 part
└─sdc5   8:37   0    32G  0 part [SWAP]
sr0     11:0    1  1024M  0 rom
```

`sudo lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL`

```
NAME   FSTYPE   SIZE MOUNTPOINT LABEL
sda               3T
├─sda1          128M
└─sda2 ntfs       3T            D:ataBig
sdb             2,7T
├─sdb1          128M
└─sdb2 ntfs     2,7T            E:xtra
sdc           407,9G
├─sdc1 ext4   375,9G /
├─sdc2            1K
└─sdc5 swap      32G [SWAP]
sr0            1024M
```

`sudo lsblk -f`

```
NAME   FSTYPE LABEL    UUID                                 MOUNTPOINT
sda
├─sda1
└─sda2 ntfs   D:ataBig 10BC78DDBC78BEB2
sdb
├─sdb1
└─sdb2 ntfs   E:xtra   864E5B474E5B2F65
sdc
├─sdc1 ext4            3def4638-e25a-46bb-befb-be950341b73a /
├─sdc2
└─sdc5 swap            c454fc26-249e-462e-90a5-72e3cb4b212a [SWAP]
sr0
```

Check if we have NTFS support (may need other numbers instead of -45 and -31):

```
ls /lib/modules/4.4.0-45-generic/kernel/fs | grep nt
```

```
ntfs
```

```
ls /lib/modules/4.4.0-31-generic/kernel/fs | grep nt
```

```
ntfs
```

Make the two mount points for the two LSI MegaRAID extenders, here re-using the old Windows labels D:ataBig and E:xtra without colon:

```
sudo mkdir /megaRaid
sudo mkdir /megaRaid/DataBig
sudo mkdir /megaRaid/Extra
```

Mount and test one by command line:

```
sudo mount -t ntfs-3g /dev/sda2 /megaRaid/DataBig/
ls  /megaRaid/DataBig/Home/weinert/Vortrag
```

```
insgesamt 11440
-rwxrwxrwx 1 root  828928 Apr 26  2001 AutoJavaWe.ppt
-rwxrwxrwx 1 root 1185792 Jun 25  2001 inf_i_2.ppt
-rwxrwxrwx 1 root   27863 Jan  5  2001 java4ing-cover.jpg
-rwxrwxrwx 1 root  927232 Apr 26  2001 Java_Start_we.ppt
-rwxrwxrwx 1 root  475136 Jun 13  2001 JavaUebWe.ppt
               ----   snippet ---
```

Prepare fstab auto-mount by

```
sudo nano  /etc/fstab
```

and add two entries at the end:

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sdc1 during installation
UUID=3def4638-e25a-46bb-befb-be950341b73a /               ext4    errors=remount-ro 0       1
# swap was on /dev/sdc5 during installation
UUID=c454fc26-249e-462e-90a5-72e3cb4b212a none            swap    sw              0       0

UUID=864E5B474E5B2F65  /megaRaid/Extra  ntfs-3g
  defaults,windows_names,locale=de_DE.utf8  0 0

UUID=10BC78DDBC78BEB2  /megaRaid/DataBig ntfs-3g
defaults,windows_names,locale=de_DE.utf8  0 0
```

Then do:

```
sudo mount -a
```

The first time only you may see:

```
The disk contains an unclean file system (0, 0).
The file system wasn't safely closed on Windows. Fixing.
```

Test and try the mounts by

```
ls /megaRaid/Extra
ls /megaRaid/DataBig
```

and digging deeper. The Windows domain users owners and rights are gone or at least invisible so far; all belongs to root. Besides that, all files in the two LSI MegaRAID extenders formatted by NTFS (in a domain yet unknown) are accessible by the Ubuntu server.

Try a reboot and check the auto-mount after the spinning up and logging in again:

```
sudo shutdown --reboot now
# sudo drink coffee for 5...10 minutes before re-loggin in
sudo lsblk -f
```

```
NAME   FSTYPE LABEL    UUID                                 MOUNTPOINT
sda
├─sda1
└─sda2 ntfs   D:ataBig 10BC78DDBC78BEB2                     /megaRaid/DataBig
sdb
├─sdb1
└─sdb2 ntfs   E:xtra   864E5B474E5B2F65                     /megaRaid/Extra
sdc
├─sdc1 ext4            3def4638-e25a-46bb-befb-be950341b73a /
├─sdc2
└─sdc5 swap            c454fc26-249e-462e-90a5-72e3cb4b212a [SWAP]
sr0
```

The quite long time to shut-down and re-boot is mostly due to the LSI MegaRAID extenders.

This delay is a bit bothersome when having to test something by re-boot. For a 24/7 server normally not booted this is no problem.

Collabnet Subversion Edge

This was done on Fujitsu server with the standard distro, only.

And in the end, with Ubuntu 16.04 using Collabnet distribubions, to get Apache and SVN server or to get SVN after having Apache 2.4 running was just a detour. If interested, see [26].

Part I's intermediate results – and due decisions

We had installed almost all services needed with one non-server distribution of Ubuntu. And we had all parts to remotely configure and administrate our server – so we could start working, experimenting, playing, and preparing for next deployments being as homologous as feasible. Having no sensible configuration for web and SVN should be mendable by installing Apache 2 and SVN separately – it was.

But besides missing the RAID drives, we have had at least two user bases one for Ubuntu, including remote access and FTP (vsftp) – and a totally different one for https and svn. We do need

- **+** a common user / group base for system and all services
  As usual a common ID management (whatsoever) will be a challenge –

LDAP seems the best choice with the chance to unite the separate user bases, as the OS and most applications are assumed to be LDAP aware. Hence we threw the standard installation on the server away, abdicating the graphical HMI.

- **+** We had some fights to get the LAN interfaces running and we got our RAID extenders operational. This is reported above.
- **-** But with 16.04, both standard and server, openLDAP (server) could not be made operational.

As of March 2017, the reason seems the (implementation of) the new so-called "slapd-config method", where LDAP (server) configuration seems to be no longer kept in text files but in separate LDAP DIT, no one has admin access to by default, and no one seems able to get them by slapd-config. This is just hypothesis. Nevertheless trying all googled tricks to get around this bug has cost us days with no success here and without, of course, progress in other fields.

We could:

- **-** wait some months for the bug ([3], [4]) go away and the documentation become fully consistent with the new configuration approach: the "slapd-config method"
  (No, we could not.)
- **-** wait for the 99th published trick / work around to get all working, *)
- **-** look for an alternative approach or ....?

The approach chosen:

In this situation we decided to install openLDAP without LDAP server respectively, to be precise, and with no extras (TLS) nor configurations for use as server. This way, LDAP is just to be used as a local common Id base for all users except the one originally born by Ubuntu installation. All additional Ubuntu users are to be made in LDAP and "PAMmed" to the OS and hence be available for those applications able to use OS accounts. All others should be LDAP aware and content with a rudimentary local server function. With this approach we got the OS, FTP, Apache, SVN and else (also) with LDAP users on one of the “real” Fujitsu servers.

On other machines LDAP didn't work at all. So we omitted it completely. From the application's point of view no LDAP or LDAP "PAMmed" only makes no difference.

*) In between, in [12], we read a consistent explanation of the openLDAP disaster and a potential remedy in the course of installation. As too late for the current installations we haven't given it a try,yet.

As long as having no application that isn't able to use local Id be no other method as LDAP "PAMmed LDAP" is worse than having none. Non withstanding [12]'s potential solution, we recommend not to use openLDAP as long as not repaired nor consistently documented.

PART II

Part I was on

installations, procedures, recipes applicable to all above distros and targets and on detours and errors to avoid respectively to learn from or to base decisions on.

Part II deals with the installation of the server(s) now running and being used 24/7.

Mounting drives and directories

Mounting extra (LSI MegaRAID) drives was done on Fujitsu server with the Ubuntu 16.04 server distribution, see more in Part I. Mounting directories instead of linking them is mandatory to make them visible by FTP (vsFTPd see below).
To make those mount permanent add entries in etc/fstab by

```
sudo nano  /etc/fstab
```

But beware: Minimal errors here are good for serious if not catastrophic boot problems. And just copying lines from /proc/mounts or /etc/mtab to /etc/fstab – a tip seen often – will not always work.
The addition of one (MegaRAID) directory to appear in one user's home and ftp tree is:

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the UUID for a device; this may be used with UUID= as a robust
# way to name devices; works if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# by OS installation:
# / was on /dev/sdc1 during installation
UUID=3def4638-e25a-46bb-befb-be950341b73a /            ext4    errors=remount-ro 0   1
# swap was on /dev/sdc5 during installation
UUID=c454fc26-249e-462e-90a5-72e3cb4b212a none         swap    sw              0     0

# added to have the two LSI megaRAID extenders (Extra re-formatted):
UUID=6290229a-ad86-4fc3-8107-31811a134e97  /megaRaid/Extra  ext4 errors=remount-ro  0 0

UUID=10BC78DDBC78BEB2  /megaRaid/DataBig ntfs-3g
  defaults,windows_names,locale=de_DE.utf8  0 0

# added to mirror (bind link) rirectory trees in user's home/FTP sight:
# mount /megaRaid/DataBig(/ftp) as /home/albrecht/ftp   for fileZilla
/megaRaid/DataBig/ftp /home/albrecht/ftp none bind 0 0
```

Do test all changes by

```
sudo mount -a
```

If and only if this gives no errors and all seems all well, try a reboot – no use in postponing Armageddon.
The quite long time to shut-down and re-boot is mostly due to the LSI MegaRAID extenders. In our case they are good for a 2'40 pause while booting with no visible nor audible actions.

openLDAP

The goal to use openLDAP as uniform ID management for all services and targets met with no success in our (and many other's) cases, no matter the toils and work weeks spend.
See [26] if interested in this tale of woe.

FTP – by vsFTPd

The installation of the "very secure FTP server" vsFTPd worked on all server targets. vsFTPd uses OS users, groups and rights. Hence vsFTPd brings no need for LDAP, but, if there, it works with the "PAMmed" LDAP users (cf. [26]), seamlessly. To install vsFTPd do:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install vsftpd
sudo apt autoremove
sudo service vsftpd stop
```

If server key and certificate are here, use them. Otherwise, get or make them e.g. by:

```
sudo openssl req -x509 -nodes -days 1138 -newkey rsa:1024
-keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/certs/ssl-cert-
vsftpd.pem  ## three lines are one
```

Now we have two files:

```
-rw-r--r-- 1 root root 2027  2016-11-17 13:45 /etc/ssl/private/vsftpd.pem
-rw-r--r-- 1 root root 1111  2016-11-25 11:58
                                   ./etc/ssl/certs/ssl-cert-vsftpd.pem
```

and change the file containing the private key by:

```
sudo chown   root:ssl-cert   /etc/ssl/private/vsftpd.pem
sudo chmod   o-r         /etc/ssl/private/vsftpd.pem
```

Remark: We didn't invent, but just used this. Coming from Windows NT and/or its heirs one is aghast to see the ubiquitous "all can read" as remedy to Linux' missing groups in groups and minimalistic file rights concept even with private keys and other critical data. Looking for better solutions isn't easy: Whenever one removes "all can ..", as in the second command above convinced to remove just danger and seeing all run on, it has happened that customers do complain days after that long forgotten change: "Suddenly we cannot ...".
Another, really evil, trap is a user in the group having less rights as the owner with the same bits [sic!].

Now work on vsFTPs configuration by:

```
sudo cp    /etc/vsftpd.conf  /etc/vsftpd.conf.orig
sudo nano   /etc/vsftpd.conf
```

Make /etc/vsftpd.conf look that way (most comments omitted or deleted):

```
# /etc/vsftpd.conf  25.11.2016   Albrecht Weinert     with TSL   on PD321S

listen=NO
listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
chroot_local_user=YES
allow_writeable_chroot=YES
chroot_list_enable=NO

pasv_enable=Yes
pasv_min_port=40000
pasv_max_port=40100
```

```
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO

require_ssl_reuse=NO
ssl_ciphers=HIGH
```

If the server running vsFTPd has a firewall or is as virtual server behind its host's one do not forget to punch through the ports 20 and 21. If using passive mode, make a shorter port range, like 50000..50009 e.g., and drill a hole for that, too. Then do:

```
sudo service vsftpd start
```

Use fileZilla on a Windows workstation to test more than one user. If all is done and mounted as described in previous chapters, both users can see and manipulate their home directories and are confined to that. Additionally, the users see and and may manipulate

a) a directory tree outside there home or
b) a directory in one of the big LSI megaRAID drives

if and only if that tree was mounted inside home.

Note 1: b) is logically equivalent to a), but it is fine to see LSI megaRAIDs work with vsFTPd.
Note again: Mounting this way works with vsFTPd, just linking does not.

Using user's home + mounts as strategy

vsFTPd makes the user's home directory available via FTP. This gives a normal user all she could see and change if logged in. Hence as said, no extra rights are necessary nor extra dangers are produced. For each tree outside home to be made visible the user in question needs an empty (!) directory, ~/ftp/ and ~/mountSioux/etcApache2/ in that "add to /etc/fstab" example:

```
/megaRaid/DataBig/ftp  /home/albrecht/ftp  none bind 0 0
/megaRaid/DataBig/ftp  /home/weinert/ftp   none bind 0 0
/etc/apache2   /home/albrecht/mountSioux/etcApache2   none bind 0 0
```

FTP users with no login

To exclude "FTP" users from logging in give them /bin/false as shell. To exclude them from graphical (RDP) login, if available put all others in group tsusers, as described in Part I.

For non graphical HMI (command shell and putty) the main point is adding to /etc/shells the line:

```
/bin/false
```

For making LDAP FTP only users put following line in the .ldif:

```
loginShell: /bin/false
homeDirectory: /home/ftp21
```

Using the second allows for non standard home=ftp directory. For non LDAP use something like:

```
usermod -m -d /newhome/username username
```

Apache 2

Besides being a SVN server, SubversionEdge has formerly been one of the best Apache distributions and we used it in both roles. Do not do this with Ubuntu (16.04) now, see [26]. We install Apache 2 separately.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install apache2
```

In the course of the installation we see a long list of modules and configs already enabled:

```
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
   # ::: > 10 omissions
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default
```

Some modules are prepared but commented out (as localized-error-pages). Some more disturbing than helpful (as other-vhosts-access-log) are enabled. iI so you may a2dis... them, (later!).
A first check:

```
sudo apache2ctl configtest
```

says OK, contrary to [6]'s prediction.

And right now Apache 2 should be running. It just serves its test page on port 80 (http) and nothing more. While setting up FTP access to Apache, we now may stop the server by:

```
sudo service apache2 stop
```

Make Apache 2 configuration and content accessible by other users

We want some other users give access to Apache 2's

- web server configuration   based on /etc/apache2/

and

- web content   based on /var/www/html   by installation default

Here html/ is the root of the prepared all domain (i.e. no domain) exemplary web content. For (optionally) multi domain web servers we make

- multiple web contents   under /var/www/sites/...   one subdirectory per host

and to prepare for a multiple repositories SVN server, run by Apache (later), we make

- multiple SVN repos   under /var/www/repos/...   one subdirectory per repo

This requires, of course, according changes in /etc/apache2/apache2.conf as well as all used or made *.conf files in /etc/apache2/sites-available.

The Apache 2 server runs as www-data:www-data and, hence, always needs the other's `r--` respectively `r-x` to operate in case of files or directories put to /etc/apache2 or /var/www/sites/ having other ownership. And to note it here, already:
For SVN, i.e. /var/www/repos/ in our configuration the ownership (www-data) must fit.

These are the boundary conditions to fit the other users local and FTP access in.

To sum up, we need

- read access by Apache 2 to all its content and configuration,
- write access also by Apache 2 to special content as e.g. repositories and
- FTP access to most or all of it to special administrative users,
  preferably controlled by group membership.

Hence the /etc/var/www/ situation as installed is no good. The services working depend on

- the "all can read" or on
- the services or parts of them (SVN?) running as sudo/root..

and

- FTP access for administration or maintenance can't be implemented on that base.

Before going further, only on the "real server" with megaRAID extenders, we will transfer all web content and repositories from /etc/var/www/ to /megaRaid/Extra/sites/.

```
sudo mkdir /megaRaid/Extra/sites
sudo mkdir /megaRaid/Extra/sites/html
sudo mkdir /megaRaid/Extra/sites/repos
sudo cp -R /var/www/html/    /megaRaid/Extra/sites/
diR    /megaRaid/Extra/sites/html/
```

On other servers one may modify or just omit this change. Anyway it will be handy to have some variables for bash users and scripts:

```
WEB_SITES=/var/www/sites  # web sites root
WEB_DIR_OWNER=www-data    # Apache2 runs so
WEB_DIR_GROUP=web_admin   # this is for administrative file access (FTP)
SVN_REPOS_DIR=/var/www/repos      # where the server repos are
SVN_WORK_DIR=/var/www/svnWork     # here (automated) local work is done
SVN_REPOS_OWNER=www-data  # we access SVN from outside by Apache only
SVN_REPOS_GROUP=svn_admin # this is for administrative file access (FTP)
```

It might be feasible to to use or keep www-data for both ..GROUPs above.

Now if some rights or ownerships are spoiled we can repair all by:

```
chown "$WEB_DIR_OWNER:$WEB_DIR_GROUP" -R $WEB_SITES/
find $WEB_SITES -type f -exec chmod 770 {} +
find $WEB_SITES -type d -exec chmod 775 {} +
```

or for the configuration by

```
chown "$WEB_DIR_OWNER:$WEB_DIR_GROUP" -R /etc/apache2/
find /etc/apache2 -type f -exec chmod 770 {} +
find /etc/apache2 -type d -exec chmod 775 {} +
```

Best put this recipe in one or more scripts or sourceable functions preferable with a --all or a dir parameter, some checks and comfortably with a --help.

Those problems to repair usually occur after a privileged user made or copied files by e.g.

```
sudo cp  /etc/apache2/sites-available/default-ssl.conf
/etc/apache2/sites-available/pd321s-ssl.conf
```

thus giving the file just made root:root ownership. That may have been avoided by:

```
sudo cp  -p  /etc/apache2/sites-available/default-ssl.conf /etc/..
```

... Access Apache 2 configuration and content by FTP

Let's assume FTP being operational as described above. To remember: vsFTPd lets each authenticated user see his home directory via FTP and if we need to see more than one drive respectively two or more separate trees we mount those therein.

And lets assume the FTP connection to a comfortable workstation is fast and secure. Our workstation features multi windows viewing, editing, printing copy and paste from everywhere to everywhere etc. (like Windows7 with two big monitors; see the remarks on graphical HMI and xRDP in Part I).

So we enable handling Apache 2 by FTP and we facilitate researching the Apache 2 configuration on an Ubuntu server with almost no HMI. Make it FTP visible within a standard home by four or later two commands respectively by this script:

```
# ~/bin/mountSioux     Rev. 2.0   23.11.2016
# (c) Albrecht Weinert 2016     a-weinert.de
#
# Mount  Apache 2 ...  into   /home/~/   open to FTP
# configuration = /etc/apache2  as ~/mountSioux/etcApache2
# web content    = /var/www      as ~/mountSioux/varWWW
#
# Make the mounts (... to ...) by   mountSioux
# or by                  sudo ~/bin/mountSioux

mkdir  -p  ~/mountSioux/etcApache2
sudo mount --bind  /etc/apache2  ~/mountSioux/etcApache2

mkdir  -p  ~/mountSioux/varWWW
sudo mount --bind  /var/www  ~/mountSioux/varWWW
# sudo mount --bind  /megaRaid/Extra/sites  ~/mountSioux/varWWW
```

Note1: If on the "big server" with /megaRaid/Extra/ drives interchange the un-commenting of the last two lines.

Note 2: Before getting too enthusiastic, that `mount --bind` and hence the script require sudo. After running the script, on an remote workstation, we can see Apache 2's content and configuration via FTP, e.g. by FileZilla (Fig.2).

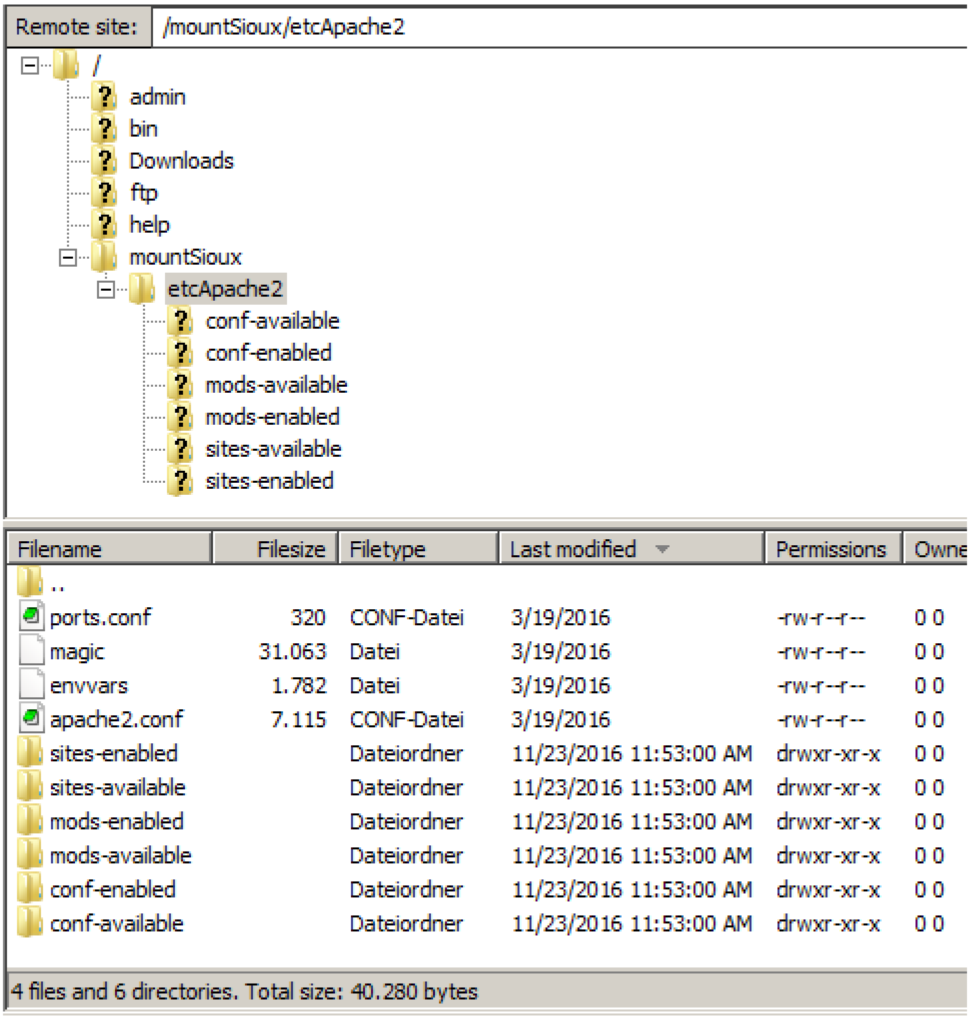
Make ~/bin/mountSioux executable by

```
chmod 775  ~/bin/mountSioux
```

To have some documentation remotely available, one may put it in one's ~/document directory. Get some extra documentation:

```
mkdir -p  ~/docu/apache2
cp  /usr/share/doc/apache2/README.Debian.gz  ~/docu/apache2
gunzip    ~/docu/apache2/README.Debian.gz
```

After that the unzipped document/apache/README.Debian is visible at the workstation via FTP, too.



Remote site: /mountSioux/etcApache2

```
/
  admin
  bin
  Downloads
  ftp
  help
  mountSioux
    etcApache2
      conf-available
      conf-enabled
      mods-available
      mods-enabled
      sites-available
      sites-enabled
```

Filename	Filesize	Filetype	Last modified	Permissions	Owne
..					
ports.conf	320	CONF-Datei	3/19/2016	-rw-r--r--	0 0
magic	31.063	Datei	3/19/2016	-rw-r--r--	0 0
envvars	1.782	Datei	3/19/2016	-rw-r--r--	0 0
apache2.conf	7.115	CONF-Datei	3/19/2016	-rw-r--r--	0 0
sites-enabled		Dateiordner	11/23/2016 11:53:00 AM	drwxr-xr-x	0 0
sites-available		Dateiordner	11/23/2016 11:53:00 AM	drwxr-xr-x	0 0
mods-enabled		Dateiordner	11/23/2016 11:53:00 AM	drwxr-xr-x	0 0
mods-available		Dateiordner	11/23/2016 11:53:00 AM	drwxr-xr-x	0 0
conf-enabled		Dateiordner	11/23/2016 11:53:00 AM	drwxr-xr-x	0 0
conf-available		Dateiordner	11/23/2016 11:53:00 AM	drwxr-xr-x	0 0

4 files and 6 directories. Total size: 40.280 bytes

Fig2: mountSioux

Making a link from an "...-enabled" directory to a file of the same name in an "...-available" directory is Apache 2's rite to choose or switch modules, sites and configuration. Among others, this approach replaces loads of lines in a central configuration file saying "LoadModule X_module mod_X.so".

Copying above tree via fileZilla (Fig. 2) client to a temporary directory in the workstation to play with will make files (copies) instead of the original links. Beware of modifying those files. Make changes in the ...-available directories. The best comfort may be achieved on a Windows workstation with FileZilla's edit/view feature.

Automating the FTP access by mounts

So far a user wanting access the Apache configuration and or content by FTP has to login and run the mountSioux script:

```
dir mountSioux/etcApache2/
```

```
total 0
```

```
mountSioux
```

```
[sudo] Password for weinert:
```

```
dir mountSioux/etcApache2/
```

```
total 80
-rw-r--r-- 1 root root  7115 2016-03-19 10:48 apache2.conf
drwxr-xr-x 2 root root  4096 2016-11-23 10:53 conf-available
drwxr-xr-x 2 root root  4096 2016-11-23 10:53 conf-enabled
-rw-r--r-- 1 root root  1782 2016-03-19 10:48 envvars
-rw-r--r-- 1 root root 3.......
```

When remote connecting in this state by FTP as the user in question (weinert or albrecht in our examples) all that is visible. Without prior running this script it is not. *1)

To automate the "mountSioux" approach one might:

- put this script's mount actions (also) to /etc/fstab

or

- run a similar script at user's remote (FTP) login authentication.

The first implementation as been done here already, it works reliably and it is applicable for non sudoable and LDAP based users and even for those without login. The disadvantage is every user requires entries in fstab and every change of users or their FTP requirements or rights lead to modifications of this central and mission critical configuration file.

The second approach is not proven yet, it might be complicated but in case of success it will be quite flexible and powerful in the end. Anyway if ever, it only work with "sudoable" users, in the end. In any case this approach's implementation would start by:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install libpam-script
```

More to be done and reported on in later versions (in case of good success).

Hint *1) : With exemplary "weinert" as the basic Ubuntu and hence non-LDAP user as is Apache's www-data we enter the field of mixing LDAP non LDAP. To put any user in a LDAP group do ldif by uid; To put any user in a non LDAP group use just the adduser command. This works and has effects with correctly installed PAM modules.

FTP access by explicit home

This is an alternative for pure FTP users without sudo and without the right to log in. Such kind of user is sometimes called "virtual". If they need just one tree / drive outside /home/ we just make this tree the user's home.

See the recipe below in Chapter "Miscellaneous commands", "Change user's home directory".

FTP access by forced fstab mount

This is an alternative also for non sudo users with login right and standard home directory. Add to /etc/fstab something like:

```
/etc/apache2  /home/albrecht/mountSioux/etcApache2  none bind 0 0
/var/www      /home/albrecht/mountSioux/varWWW  none bind 0 0
# /megaRaid/Extra/sites      /home/albrecht/mountSioux/varWWW  none bind 0 0
```

As said /etc/fstab is a mission critical central OS configuration file. After changes always test with

```
mount -a
```

And best have a second administrative terminal open for a last chance to undo mistakes.

An incorrect fstab may impede OS re-boot. Hence this approach is good for some stable selected users only.

Adding TSL

As soon as Ubuntu does something with SSL it uses a well known daftly named pubic/private key pair coming with every installation:

```
-rw-r----- 1 root ssl-cert   1704 2016-11-21 12:07
                                   /etc/ssl/private/ssl-cert-snakeoil.key
-rw-r--r-- 1 root root       1054 2016-11-21 12:07
                                   /etc/ssl/certs/ssl-cert-snakeoil.pem
```

Traditionally Apache 2 and other programmes using TLS will use this pair, too, by default. Of course, it should be replaced by a "real" or self-signed server certificate and all applications in question running on that server should use it. We want the same for Apache as for FTP and have to edit

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

changing two lines to

```
#         SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
#         SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
SSLCertificateFile   /etc/ssl/certs/ssl-cert-vsftpd.pem
SSLCertificateKeyFile       /etc/ssl/private/vsftpd.pem
```

Then enable SSL in the default / test Apache 2 configuration by:

```
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo service apache2 restart
```

As well as the filleZilla client before, Chrome (browser) wants the self signed certificate explicitly confirmed, then it works. To be more exact: Chrome complains about the certificate -- in detail:

Security Overview: This page is insecure (broken HTTPS), Certificate Error

- There are issues with the site's certificate chain (net::ERR_CERT_AUTHORITY_INVALID).

+ The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA), and a strong cipher (AES_128_GCM).

+ Secure Resources: All resources on this page are served securely.

Hence, Chrome just complains on self-signing. All else with our home made pair is considered OK.

Multi-domain self-signed certificate

As soon as we have multiple domains on one server / in one Apache 2 we should either have one certificate per domain or (better) a multi-domain certificate for all domains and virtual 443 hosts. Otherwise one gets more than one "self-signed" complaint. In the end we replace (cf. above)

```
  SSLCertificateFile   /etc/ssl/certs/ssl-cert-vsftpd.pem
  SSLCertificateKeyFile       /etc/ssl/private/vsftpd.pem
```

by

```
  SSLCertificateFile  /etc/ssl/certs/ai2t.multidomain.crt
  SSLCertificateKeyFile       /etc/ssl/private/vsftpd.pem
  SSLCertificateKeyFile       /etc/ssl/private/vsftpd.pem
```

in all 443 virtual hosts.
To get there we prepare two files in ~/work. First is req.conf:

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
prompt = no
[req_distinguished_name]
C = DE
ST = NRW
L = Bochum
O = weinert - automation
OU = MEVA-Lab
CN = ai2t.de
[v3_req]
keyUsage = keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
[alt_names]
DNS.1 = ai2t.de
DNS.2 = www.ai2t.de
DNS.3 = dgmev.de
DNS.4 = www.dgmev.de
DNS.5 = www.dgmev.org
DNS.6 = dgmev.org
DNS.7 = www.fbt-mechatronik.de
DNS.8 = fbt-mechatronik.de
DNS.9 = www.mechatronics-net.de
DNS.10 = mechatronics-net.de
DNS.11 = www.rem2015.de
DNS.12 = rem2015.de
```

The second file 2t.multidomain.cnf is:

```
subjectAltName=DNS:ai2t.de,DNS:www.ai2t.de,DNS:dgmev.de,DNS:www.dgmev.de,DNS:www.dgmev.org,DNS:dgmev.org,DNS:www.fbt-mechatronik.de,DNS:fbt-mechatronik.de,DNS:www.mechatronics-net.de,DNS:mechatronics-net.de,DNS:www.rem2015.de,DNS:rem2015.de
```

Note, in req.conf, the CN being also listed as DNS.1. Note also the need to list each and every sub-domain you have or might have in near future. This includes the immortal "www." even if always rewritten.

The second file, ai2t.multidomain.cnf, is a all in one liner, to be generated consistent with req.conf.

After having prepared those two files accordingly, do the following:

```
cd work/
touch req.conf #### and put content above in
cat req.conf
dir  /etc/ssl/private/vsftpd.pem
openssl req -new -out ai2t.de.csr -key /etc/ssl/private/vsftpd.pem
-config req.conf
```

```
# make request
sudo  openssl req -new -out ai2t.de.csr -key
/etc/ssl/private/vsftpd.pem -config req.conf
# check request
openssl req -text -noout -in ai2t.de.csr
touch ai2t.multidomain.cnf #### and put above content in
sudo   openssl x509 -req -days 7300 -in ai2t.de.csr -signkey
/etc/ssl/private/vsftpd.pem  -text -extfile ai2t.multidomain.cnf
-out ai2t.multidomain.crt
```

```
cat ai2t.multidomain.crt
sudo cp ai2t.multidomain.crt  /etc/ssl/certs/
dir  /etc/ssl/certs/ai2t.multidomain.crt
```

Note on "#### and put content above in":
Here you can, of course, use nano.
But with much more comfort, have a FileZilla open and use a decent editor on your (Windows) workstation with FileZilla's "edit/view" feature.

Adding PHP

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install php libapache2-mod-php php-mcrypt
```

Then, in the file by /etc/apache2/mods-available/dir.conf, one might change the default index file priority to:

```
    DirectoryIndex index.php index.html index.htm index.pl index.xhtml
```

Then we prepare the standard php test file by

```
sudo mkdir /var/www/html/info/
sudo nano /var/www/html/info/info.php
```

giving it the content

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

Test it by http(s)://PD321S/info/info.php. It works both by http and by https.

Adding authentication with LDAP

If you do or wan't to use [open]LDAP – a detour in all our cases – red the chapter in [26].

Apache 2 authentication for virtual server with no LDAP

On targets, where even the minimal openLDAP approach won't run or if just don't want it, we'd like to have OS/Ubuntu users and groups for authentication. As this works for FilleZilla, why have extra password files and Id management for Apache 2? The following was done and tested on a rented virtual Ubuntu 16.04 server with no LDAP. In this stage we hadn’t domains attached to the server and used the public IP, here the exemplary 82.165.72.47.

Rationale: We didn’t want domains transferred here, before this is running and tested.

We start by making two new available sites, for no domain to start with:

```
sudo a2enmod rewrite
sudo cp -p /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/noDomain.conf
sudo cp -p /etc/apache2/sites-available/default-ssl.conf
/etc/apache2/sites-available/noDomain-ssl.conf
```

and while at it,by saving the original configs by:

```
sudo cp -p /etc/apache2/apache2.conf
/etc/apache2/apache2.conf.orig
sudo cp -p /etc/apache2/envvars /etc/apache2/envvars.orig
sudo cp -p /etc/apache2/ports.conf /etc/apache2/ports.conf.orig
```

Now make /etc/apache2/sites-available/noDomain.conf look something like:

```
<VirtualHost *:80>
  #ServerName www.example.com  # 82.165.72.47 ubu1u1
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html

# all under info/ gets https and authentication, there
  <Location /info>
   RewriteEngine On
   RewriteCond %{HTTPS} off
   RewriteRule (.*) https://82.165.72.47%{REQUEST_URI} [R=301,L]
  </Location>
</VirtualHost>
```

Then do:

```
sudo a2dissite 000-default
sudo a2ensite noDomain
sudo service apache2 restart
```

Now test http://82.165.72.47/info/ being forced to https://82.165.72.47/info/. This is the precondition for basic authentication and all based on it.
Now prepare OS/Ubuntu user authentication (cf. [11]) by:

```
sudo apt-get install libapache2-mod-authnz-external pwauth
sudo apt-get install libapache2-mod-authz-unixgroup
sudo a2enmod authnz_external authz_unixgroup
```

Now make /etc/apache2/sites-available/noDomain-ssl.conf look something like:

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html

  SSLEngine on
  SSLCertificateFile   /etc/ssl/certs/ssl-cert-vsftpd.pem
  SSLCertificateKeyFile       /etc/ssl/private/vsftpd.pem

  <FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
  </FilesMatch>
  <Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
  </Directory>

  <IfModule mod_authnz_external.c>
    AddExternalAuth pwauth /usr/sbin/pwauth
    SetExternalAuthMethod pwauth pipe
  </IfModule>

  <Directory /var/www/html/info>
    AuthType Basic
    AuthName "Restricted infos"
    AuthBasicProvider external
    AuthExternal pwauth
    Require user albrecht
  </Directory>

</VirtualHost>
</IfModule>
```

Then do:

```
sudo a2dissite default-ssl
sudo a2ensite noDomain-ssl
sudo service apache2 restart
```

Enjoy if 82.165.72.47/info/info.php is both forwarded to https and requires Ubuntu user's name and password (over https!).
But having to name users is only half the joy. In the exemplary case of albrecht

```
groups albrecht
```

```
albrecht : albrecht sudo web_admin
```

we'd like to use a group (web_admin in our example) to allow all users with this privilege in just one line. Mostly according to [11] do:

```
cd Downloads/
wget https://storage.googleapis.com/google-code-archive-
downloads/v2/code.google.com/pwauth/pwauth-2.3.11.tar.gz
```

```
tar xzvf ./pwauth-2.3.11.tar.gz
sudo cp  pwauth-2.3.11/unixgroup  /usr/sbin/
dir  /usr/sbin/unixgroup
```

Then change two blocks in /etc/apache2/sites-available/noDomain-ssl.conf:

```
<IfModule mod_authnz_external.c>
   AddExternalAuth pwauth /usr/sbin/pwauth
   SetExternalAuthMethod pwauth pipe
   AddExternalGroup unixgroup /usr/sbin/unixgroup
   SetExternalGroupMethod unixgroup environment
 </IfModule>

 <Directory /var/www/html/info>
   AuthType Basic
   AuthName "Restricted infos"
   AuthBasicProvider external
   AuthExternal pwauth
   GroupExternal unixgroup
   # Require user albrecht
   Require unix-group  web_admin
 </Directory>
```

Again re-start Apache 2 and make your browser forget all to re-test 82.165.72.47/info/info.php with your server's IP.

It should work. Other versions of pwauth/unixgroup might need "Require group ..". instead of "Require unix-group ..". If this is the wrong way round a) authentication may be bypassed, b) authentication may loop endlessly or c) Apache 2 won't re-start with "control process exited with error code" and no further information.

Redirect www. to non www.

Many people cannot refrain from putting www. in front of everything in a web, even if there's no such (sub) domain. Of course, if www.PD321S won't be put to this server by DNS we do not have to handle this in Apache. To produce the problem or to satisfy the offender one must add www... to DNS or (for home made tests) to the hosts file.

Then, to politely serve and inform the misguided user, we put appropriate rewrites/redirects in our virtual hosts, by something like this:

```
<VirtualHost *:443>
  ServerName weinert-automation.de
  ServerAdmin webmaster@weinert-automation.de
  DocumentRoot /var/www/sites/weAut

  ServerAlias  www.weinert-automation.de www.weinert-automation.net
  ServerAlias  weinert-automation.net
  ServerAlias  weAut.de www.weAut.de weAut.net www.weAut.netion.net
# we do not want www.weinert-automation.de nor the extra reserved domains.

  RewriteEngine On
  RewriteCond %{HTTP_HOST}   !^weinert-automation\.de [NC]
  RewriteRule ^(.*) https://weinert-automation.de/%{REQUEST_URI}  [R=301,L]
  :::::  etc. pp.
```

Adding lessc

We do/you should compile larger CSS from the more compact, less error prone and better readable **.less** language. And one should be able do do the compilation on the server, too (e.g. in a post commit hook). Hence we install the .less compiler lessc:

```
sudo apt-get update
sudo apt-get install node-less
type -p lessc
which lessc
lessc -v
```

```
lessc 1.6.3 (LESS Compiler) [JavaScript]
```

If the last command won't run and "type .." or "which ..." name a non existent file in /usr/local/bin/ we mend this by:

```
sudo ln   -s  /usr/lib/nodejs/less/bin/lessc
/usr/local/bin/lessc
```

This problem seems to happen quite often for reasons unknown. We observed it on the "real" Fujitsu target.

Ubuntu Subversion

Having Apache 2.4 Ubuntu 16.04 ready and running, we add (pure) Subversion:

```
sudo apt-get update
sudo apt-get install subversion subversion-tools libapache2-svn
```

```
:::::::
exim4-config (4.86.2-2ubuntu2) wird eingerichtet ...
Adding system-user for exim (v4)
:::::
```

Afterwards the SVN command line tools work:

```
svn[admin] --version
```

```
svn, Version 1.9.3 (r1718519)
   übersetzt am Mar 14 2016, um 07:39:01 auf x86_64-pc-linux-gnu

Copyright (C) 2015 The Apache Software Foundation.
```

Now we make the root directory for repositories and one test repository. On the "big server" we use a RAID extender by basing at:

```
mkdir -p  /megaRaid/Extra/sites/repos/test
```

On a normal server we base at /var/www/repos/:

```
sudo mkdir /var/www/repos/test
sudo svnadmin create  /var/www/repos/test
diR  /var/www/repos/
```

```
 ::::::::::::::::::::::::::
-rwxrwxrwx 1 root root   2 2016-12-05 09:23 format
drwxrwxrwx 1 root root 520 2016-12-05 09:23 hooks
```

Here the owner and the right seem a bit strange. We want Apache 2 (www-data:www_admin for web) handling SVN in the end. If not yet done we create a group svn_admin and put users to access/administrate SVN locally and via vsFTPd in. Without LDAP this would be, e.g.:

```
getent group
sudo groupadd -g 1002 svn_admin
sudo usermod -a -G svn_admin albrecht
```

The new group memberships works fully after a (re-) login. Afterwards we:

```
sudo chown www-data:svn_admin -R /var/www/repos/
sudo chmod -R 770 /var/www/repos/
sudo chmod 775 $(sudo find /var/www/repos/ -type d)
```

Now user albrecht in example can see the new test repository via FTP, can inspect and modify hooks, best by FilleZilla's "Edit/View", etc..

Problem is, that (part of) these commands may have to be repeated after certain SVN server actions, probably after creating repositories. We automated them by some scripts to create repositories or refresh them after, e.g., remotely editing or adding hooks.

As it turns out the ownership and rights given so far may suffice normal Apache operation, but sometimes commits (permission denied txn-current-lock) or something other will fail. And, alas, users and groups running Apache and SVN and their minimal privileges needed are hardly documented, and if often outright wrong.
Nevertheless with above settings all worked.

Integrating SVN in Apache 2

By above installation and settings of ownership and rights Apache 2 can handle SVN. Do not forget (see Apache chapter above also) to have

```
# all under svn/ gets https and authentication
  <Location /svn>
   RewriteEngine On
   RewriteCond %{HTTPS} off
   RewriteRule (.*) https://weinert-automation.de%{REQUEST_URI} [R=301,L]
  </Location>
```

(80) and (443)

```
<Location /svn>
 DAV svn
 SVNParentPath /var/www/repos
 SVNListParentPath on

 ######   rest see in chapter on SVN / below
 # SVNIndexXSLT /conf/svnindex.xsl
</Location>
```

in the respective virtual hosts.

```
sudo service apache2 restart
```

Problem is, that (part of) these commands may have to be repeated after certain SVN server actions, probably after creating repositories.

Nevertheless with above settings SVN basically worked. Check out an empty repository made on the server before above settings, check it out on a remote (Windows) machine by:

```
cd /D D:\tmp
svn checkout https://weinert-automation.de/svn/test/
```

```
Error validating server certificate for 'https://weinert-automation.de:443'
 - The certificate is not   :::::::         until Thu, 30 Jan 2020 13:28:0
 - Issuer: MEVAlab, weinert-automation, Bochum, NRW, DE
 - Fingerprint: 4e:e2:56:98:48:24:ae:39:fa:77:f5:fe:e5:f4:f9:d4:d3:da:4e:94
(R)eject, accept (t)emporarily or accept (p)ermanently? p
```

```
cd .\test
```

Fill the empty working copy by some 80 files from another repo's working copy:

```
java Update D:\eclipseWS\javaSA2iop\+  -r -v  -omitDirs .svn .
```

This copy command requires Frame4J (see Part I, Java) on the workstation. "-r" makes the update/copy recursive, "-v" verbosely lists all files and directories affected; "-omitDirs .svn" leaves
After having so added content to the new repo's new working copy,

a) add content (on Windows best with TortoiseSVN),
b) set your favourite svn properties everywhere you want, best
 svn:keywords=Date Author Revision Id HeadURL on all pure text/source files

and finally do a full remote commit:

```
D:\tmp\test>svn commit -m "first ubuntuServer fill (0)"
```

```
  :::::::::::::::::::::::
Adding         javaSA2iopDoc.list
Adding         umlexamp.txt
Transmitting file
data ..........................................................
Committed revision 1.
```

Now its possible to browse the head revision in a browser and make further ckeckouts, updates etc.

Note: For all operations described here for the workstation (remote from the server's point of view) you a) have to accept the server's (PD321S) certificate and b) log in with a (PD321S) user + password with right to https (Apache) and with rw rights to the repository in question.

So we have an operable SVN server. But, in this situation all users logged in of having (FTP) access to /var/www/repos/ can do anything with the repositories. This might be acceptable for some remote servers with few trustable admins. Nevertheless we need to restrict the access to repository files.

A beautiful repository listing by Apache

As said it is possible to browse the readable repositories with a browser. The standard listings generated are quite ugly. For some cosmetic uncomment "SVNIndexXSLT /conf/svnindex.xsl" in location svn (see above) and deliver the transforming style sheet in /conf/svnindex.xsl. Remember here / is the websites conten root:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE xsl:stylesheet [ <!ENTITY nbsp "&#160;"> ]>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns:xs='http://www.w3.org/2001/XMLSchema'
 version="2.0">

<!-- XML transformation style sheet
  for displaying the Subversion directory listing generated by
  mod_dav_svn if "SVNIndexXSLT" directive is used.
  31.10.2008: A. Weinert
  06.03.2010: A. Weinert (modif. AI2T)
  23.12.2016: transferred to ubuntu weinert-automation.de 82.165.72.47
  V.$Revision: 3 $ ($Date: 2016-12-23) $, $Author: albrecht $
-->

  <xsl:template match="*"/>

  <xsl:template match="svn">
    <html><head>
      <title>
         <xsl:if test="string-length(index/@base) != 0">
           <xsl:value-of select="index/@base"/>
           <xsl:text>: </xsl:text>
         </xsl:if>
         <xsl:value-of select="index/@path"/>
         <xsl:text> &#160; &#x2014; &#160;</xsl:text>
      </title>
      <link rel="stylesheet" type="text/css" href="/conf/svnindex.css"/>
      <script type="text/javascript">

   function getLoadDate() {
      var now = new Date();
      var Std = now.getHours();
      var Min = now.getMinutes();
      var StdF = ((Std &lt; 10) ? "0" + Std : Std);
      var MinF = ((Min &lt; 10) ? "0" + Min : Min);
      var month = now.getMonth() + 1;
      var MonF = ((month &lt; 10) ? "0" + month : month);
      return  now.getDate() + "." + MonF
       + "." + (now.getYear() + 1900) + "  " + StdF  + ":" + MinF ;
   }

   var nowDate = getLoadDate();

   function aendere() {
     document.getElementById("whereDate").firstChild.nodeValue =
                    "generated at " + nowDate;
   }
      </script>
    </head>

    <body>
      <div class="svn">
        <xsl:attribute
name="onMouseOver">javascript:aendere()</xsl:attribute>
        <xsl:apply-templates/>
      </div>
      <div class="footer">
        <xsl:attribute 
name="onMouseOver">javascript:aendere()</xsl:attribute>
        <xsl:text>Powered by </xsl:text>
        <xsl:element name="a">
           <xsl:attribute name="href">http://weinert-
automation.de/</xsl:attribute>
           weinert - automation</xsl:element><xsl:text>'s&#160;
           </xsl:text>
        <xsl:element name="a">
          <xsl:attribute 
name="href">https://subversion.apache.org/</xsl:attribute>
          <xsl:text>Subversion</xsl:text>
        </xsl:element>
        <xsl:text> &#160; </xsl:text>
        <xsl:value-of select="@version"/>
        <xsl:text> &#160; </xsl:text>
        <xsl:element name="span"><xsl:attribute 
name="id">whereDate</xsl:attribute>
          <xsl:attribute 
name="onMouseOver">javascript:aendere()</xsl:attribute>
          generated at: tt.mm.jjjj hh:mm
        </xsl:element>
        </div>
    </body></html>
  </xsl:template>

  <xsl:template match="index">
    <div class="rev">
      <xsl:value-of select="@name"/>
      <xsl:if test="@base">
        <xsl:if test="@name">
          <xsl:text>:&#xA0; </xsl:text>
        </xsl:if>
        <xsl:value-of select="@base" />
        <xsl:text> &#xA0;</xsl:text>
      </xsl:if>

      <xsl:if test="@rev > 0">
        <xsl:if test="@base | @name">
          <xsl:text> &#160; &#x2014; &#160; </xsl:text>
        </xsl:if>
        <xsl:text>Revision </xsl:text>
        <xsl:value-of select="@rev"/>
      </xsl:if>
      <xsl:if test="@rev = 0">
          <xsl:element name="a">
  <xsl:attribute name="href">https://weinert-automation.de/</xsl:attribute>
            weinert - automation
          </xsl:element>
          <xsl:text>&#160; &#x2014; &#160; </xsl:text>
          <xsl:element name="a">
            <xsl:attribute name="href">
              <xsl:value-of select="@href"/>
            </xsl:attribute>
            <xsl:text>Subversion repositories</xsl:text>
          </xsl:element>
          <xsl:text> </xsl:text>
      </xsl:if>
    </div>

    <div class="path">
       <xsl:element name="a">
       <xsl:attribute name="href">..</xsl:attribute>
       <xsl:attribute name="title">one directory up (if
possible)</xsl:attribute>
       <xsl:value-of select="@path"/>
       </xsl:element>
    </div>

    <div class="updir">
      <xsl:text>.. </xsl:text>
      <xsl:choose>
        <xsl:when test="updir">
          <xsl:element name="a">
            <xsl:attribute name="href">..</xsl:attribute>
            <xsl:attribute name="title">to the parent
directory</xsl:attribute>
            <xsl:text>[one director</xsl:text>
          </xsl:element>
          <xsl:text>y up]</xsl:text>
      </xsl:when>
      <xsl:otherwise>
      <xsl:element name="a">
        <xsl:attribute name="href">..</xsl:attribute>
        <xsl:attribute name="title">to svn / server (if
allowed)</xsl:attribute>
        <xsl:text>[parent director</xsl:text>
      </xsl:element>
      <xsl:text>y] (may be forbidden)</xsl:text>
     </xsl:otherwise>
     </xsl:choose>
    </div>

    <xsl:apply-templates select="dir"/>
    <xsl:apply-templates select="file"/>
  </xsl:template>

  <xsl:template match="dir">
    <xsl:variable name="dirclass">
      <xsl:choose>
        <xsl:when test="@rev > 0 or not(starts-with(@name,
'ai2t_adm'))">dir</xsl:when>
```

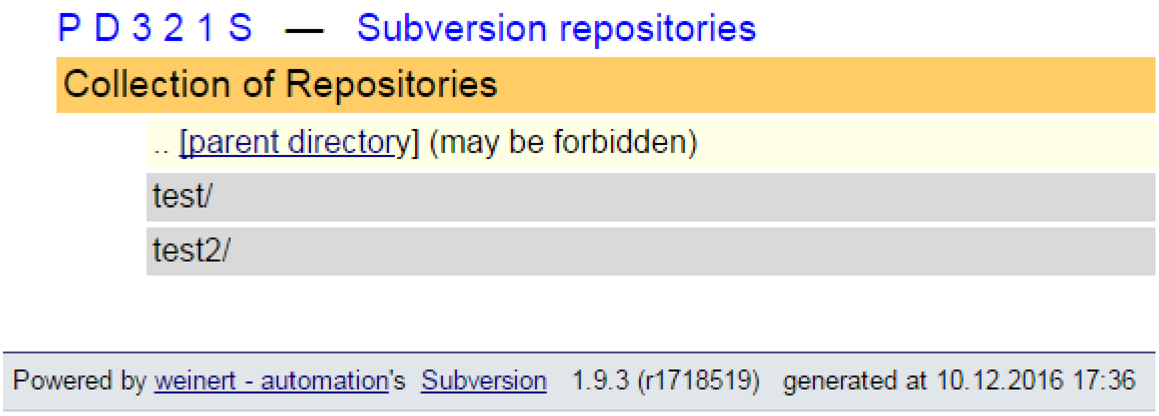
```
        <xsl:otherwise>dirsec</xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:element name="div">
      <xsl:attribute name="class">
        <xsl:value-of select="$dirclass" />
      </xsl:attribute>
      <xsl:element name="a">
        <xsl:attribute name="href">
          <xsl:value-of select="@href"/>
        </xsl:attribute>
        <xsl:value-of select="@name"/>
        <xsl:text>/</xsl:text>
      </xsl:element>
    </xsl:element>
  </xsl:template>

  <xsl:template match="file">
    <div class="file">
      <xsl:element name="a">
        <xsl:attribute name="href">
          <xsl:value-of select="@href"/>
        </xsl:attribute>
        <xsl:value-of select="@name"/>
      </xsl:element>
    </div>
  </xsl:template>
</xsl:stylesheet>
```

And we need a nice styles sheet conf/svnindex.css as mentioned in above .xsl:

```
/* style sheet for displaying the Subversion directory listing
   modif.  1.10.2008:  Albrecht Weinert
   08.12.2016: transfered to PD321S
   V.$Revision: 3 $ ($Date: 2016-10-13 11:34:13 $, $Author: albrecht $
*/
body { margin: 0; padding: 0; font-family:arial, helvetica, sans-serif; }
a { color: navy; }
.footer {  margin-top: 3em;  padding: 0.5em 1em 0.5em;
  border: 1px solid;  border-width: 1px 0; clear: both;
  border-color: rgb(30%,30%,50%) navy rgb(75%,80%,85%) navy;
  background: rgb(88%,90%,92%);  font-size: 80%;
}
.svn { margin: 2em; }
.rev { margin-right: 3px; padding-left: 3px; text-align: left;
  font-size: 120%;
}
.rev a { text-decoration: none; color: blue; }
.dir a, .file a, .path a, .dirsec a { text-decoration: none;
  color: black;
}
.path { margin: 3px; padding: 3px; background: #FFCC66; font-size: 120%;
}

.updir { margin: 3px; padding: 3px; margin-left: 3em; background: #FFEEAA;
}

.file { margin: 3px; padding: 3px; margin-left: 3em;
  background: rgb(95%,95%,95%);
}

   ::::::::  to long get it from our servers
```

Figures 3 and 4 give an impression of the style sheet's display for in that case from the big Fujitsu server by https://pd321s/svn/.....

Note: In any case only those repositories readable by the user logged in are listed (Fig. 3) and browsable (Fig. 4). Here only the head revision's files may be seen, read or downloaded by the browser respectively wget or Frame4J's Ucopy.



PD321S — Subversion repositories

Collection of Repositories

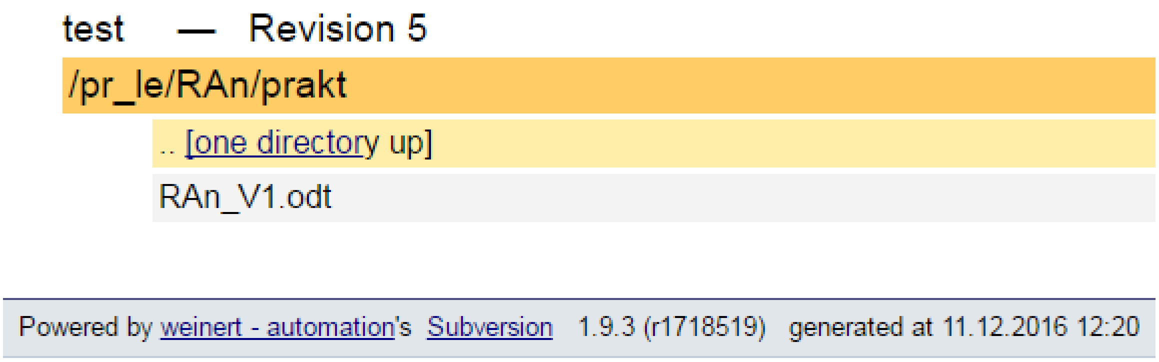
.. [parent directory] (may be forbidden)

test/

test2/

Powered by weinert - automation's Subversion 1.9.3 (r1718519) generated at 10.12.2016 17:36

Fig. 3:
Listing of repositories



test — Revision 5

/pr_le/RAn/prakt

.. [one directory up]

RAn_V1.odt

Powered by weinert - automation's Subversion 1.9.3 (r1718519) generated at 11.12.2016 12:20

Fig. 4:
Listing of a directory within one repository

Additional organisation of access rights

The configuration described now can as well be done before above tests, respectively these test have to be repeated afterwards. We already

- force https: for svn,
- enable a repository browsing -- and do it nicely with CSS.

The first point is done in the right virtual host 80 file. If not sure, check it as it is absolutely essential before using basic authentication. We now need to

- organise Apache right on a per repository base with LDAP users and groups or
- organise Apache right on a per repository base with OS users and groups on a machine without LDAP.

In the appropriate virtual host 443 we'll have a longer location svn part:

```
<Location /svn>
 DAV svn
 SVNParentPath /megaRaid/Extra/sites/repos
 SVNListParentPath on
 SVNIndexXSLT /conf/svnindex.xsl

 # Access control is done at 3 levels: (1) Apache authentication, via
 # any of several methods.  A "Basic Auth" section is commented out
 # below.  (2) Apache <Limit> and <LimitExcept>, also commented out
 # below.  (3) mod_authz_svn is a svn-specific authorization module
 # which offers fine-grained but slow read/write access control for
 # paths within a repository. (not recommended)

 # Basic Authentication is repository-wide. It is only secure as we
 # force https.

 AuthType Basic

 AuthName "weAut restricted"
 AuthBasicProvider file external
 AuthUserFile restUsers
 AuthExternal pwauth
 GroupExternal unixgroup
 require valid-user

 # common LDAP / non LDAP
 require valid-user

 # Enable authorization via mod_authz_svn:
 <IfModule mod_authz_svn.c>
   AuthzSVNAccessFile /etc/apache2/dav_svn.authz
 </IfModule>
#  SVNPathAuthz off  # this fails, turns also repo off, not just path


 #<LimitExcept GET PROPFIND OPTIONS REPORT>
   #Require valid-user
 #</LimitExcept>

</Location>
```

The simplest (not the fastest, but always reliably working) way to organise repository granular rights is the file /etc/apache2/dav_svn.authz.

Hence, we add add one:

```
# /etc/apache2/dav_svn.authz
# 06.12.2016 Albrecht Weinert
[groups]
meTheBoss=albrecht, mapa
all=albrecht, mapa, ftp21, ftp22
others= ftp21, ftp22

[/]
*=r

[test:/]
*=
albrecht=rw
```

Add the other repositories accordingly.

The result – and where we are with our server(s)

We do have

- LDAP for just user and group handling, but no directory server, on just one target
  Do consider openLDAP a resource eating detour (as of February 2017)
- vsFTPd and Apache 2 both using TLS and LDAP users
- SVN server based on Apache 2 using LDAP users for authentication

We have (Fujitsu server)

- two external RAID drives, one of which we reformatted from NTFS to ext4.

On another server we have

- no LDAP not even locally – we only tried openLDAP which installed
  but was by no means accessible or managable (and removed totally)
- vsFTPd and Apache 2 both using TLS and OS/Ubuntu users and groups
- SVN server based on Apache 2 using OS/Ubuntu users for authentication
  SVN groups are yet (to be) done on the file dav_svn.authz.
- Apache 2 as multi domain web server
  As expected, this was no problem. On the contrary, it was easier than with elder versions as Apache 2.4 has a better handling of multiple virtual hosts.

The work to get so far as reported here took much more time than predicted. This was partly to some stupid faults (easier made than detected) and mostly by some detours. The detours or errors, now documented in supplement [26], were costly regarding money and time.

The reason for these detours were

▼ long term experience with other OS and older versions of essential programmes now inappropriate as well as

▼ fundamental / architectural changes in some of those programmes, often insufficiently documented and tested.

In the “detour due to previous experience” category fall the usage of SubversionEdge for both Apache an SVN; it once was the best or on Windows almost the only way to marry Apache and SVN.

At the moment (March 2017) Edge seems just not on Apache 2.4's state of development. After having Apache 2 using Collabnet's rpm packages for pure SVN was a detour, too. Nevertheless, the reports on some detours were kept here for reference, yet.

Considering LDAP the best way to get a uniform ID management with openLDAP, was a painful experience. On most targets it just would not run even restricted to pure local ID management. In combination with 16.04 it never worked as server.
Using LDAP just locally is senseless by itself and can only be justified if it is the common denominator for authentication for all applications. After finding and getting to work the common PAM approach ([11]) we need the local LDAP running on one target no more. It is to be assumed this approach not working on a Windows system.

A short detour, not painful but disappointing, was using a non server Ubuntu for "real" servers. This is of no avail, as is the warrantable wish to install graphical HMI plus RDP on the server. As Linux has no OS integrated HMI this will be more trouble than joy. In the and we just puttyed with the servers and got all comfort by FilleZilla on a Windows workstation and by automation via SVN hooks.

So all detours, now avoidable as documented, in the end they hedged the decisions having lead to reliably running servers with Web, FTP, SVN and automated processes and hooks implemented on PHP and Java8.

Appendix

Miscellaneous commands

This is more or less an anthology of useful and proven tips.

Common checks and maintenance

See all drives:

```
lsblk
blkid
```

See all network interfaces:

```
ifconfig -s -a
ip link
```

Clean the bash command history, removing doublets and manually cleaning trash:

```
nl ~/.bash_history | sort -k 2 | uniq -f 1 | sort -n | cut -f 2 >
temp_file
```

```
nano  temp_file
cat temp_file > ~/.bash_history
```

Much went wrong with OS

```
sudo journalctl -xb
```

may give some 10000 lines of all past miss-fortunes. (Have fun.)

Format extra drive

In our example it was a megaRaid external drive originally used with NTFS inherited from the server's previous life with Windows server 2008 R2. We could have used these two drives in our first server installation with NTFS on and on. The disadvantages of using NTFS with Ubuntu for normal operation is slower access and restricted handling of normal rights.
So we want ext4 on one of them:

```
sudo umount  /dev/sdb2
dir /megaRaid/Extra/
```

```
insgesamt 0
```

```
sudo mkfs.ext4 /dev/sdb2
```

```
mke2fs 1.42.13 (17-May-2015)
/dev/sdb2 hat ein ntfs-Dateisystem mit Namen „E:xtra“
Trotzdem fortfahren? (j,n) j
Dateisystems mit 731020800 (4k) Blöcken und 182755328 Inodes wird erzeugt.
UUID des Dateisystems: 6290229a-ad86-4fc3-8107-31811a134e97
Superblock-Sicherungskopien gespeichert in den Blöcken: 32768, 98304,
163840, 229376, 294912, 819200, 884736, 1605632, 2654208, 4096000, 7962624,
11239424, ... beim Anfordern von Speicher für die Gruppentabellen: erledigt
Inode-Tabellen werden geschrieben: erledigt
Das Journal (32768 Blöcke) wird angelegt: erledigt
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt
```

With sudo nano /etc/fstab make a new line for /dev/sdb2 = /megaRaid/Extra/:

```
#UUID=864E5B474E5B2F65  /megaRaid/Extra  ntfs-3g
defaults,windows_names,locale=de_DE.utf8  0 0

UUID=6290229a-ad86-4fc3-8107-31811a134e97  /megaRaid/Extra  ext4
errors=remount-ro  0 0
```

Note that formatting might change the drive's (/dev/sdb2 here) UUID.
It is recommended to use UUID for permanent mounts in fstab. Use

```
sudo diR /megaRaid/Extra/
df -h
```

to see the newly formatted drive is almost empty. And the do the final test

```
sudo shutdown --reboot now
```

Again alive after some 10 minutes? Well done! But, ...Before formatting a drive (LSI megaRaid here) from previous NTFS to ext4 do not forget to save valuable files. Copying some Gigabytes from one NTFS megaRaid to the other was dead slow (on Ubuntu it took more than four hours). Formatting the drive to ext4, on the other hand was incredibly fast. And the drive is faster with ext4.

Make consistent and comparable directory listings

Make a good file listing command command by:

```
alias dir='ls -lA --time-style=long-iso'
```

To make it permanent and and have some comfort and the usability of sudo with it, best add the following in ~/.bash_aliases:

```
alias dir='ls -lA --time-style=long-iso'
alias diR='ls -lAR --time-style=long-iso'
alias sudo='sudo '
```

To have it for all users make it skeleton or alternatively make it a file /etc/profile.d/bash_aliases.sh instead.

Delete a directory (tree) / copy all directory (tree) content

```
sudo rmdir /home/deadUser
```

will be disappointing in most cases. `rmdir` works on totally empty directories only.
Do delete the directory including content do

```
sudo rm -r /home/deadUser
```

To copy the content of source to an existing directory dest do

```
cp -a /source/. /dest/
```

"-a" says archive, meaning recursive, keep owners and rights and preserve links. The "/." at the end of the source directory parameter is a kind of *.* including hidden files.
Note: hidden, in Linux, is no attribute as in Windows. Here files are hidden by a name starting with "." [sic!].

See all users

```
compgen -u
getent passwd

ldapsearch -x -b dc=weinert-automation,dc=de  -s sub
"objectclass=posixAccount" ## this only with LADAP and right dc=
```

See all groups

```
compgen -g
getent group
groups
```

```
ldapsearch -x -b dc=weinert-automation,dc=de  -s sub
"objectclass=posixGroup"  ## this only with LDAP and right dc=
```

Have a look at LDAP (if we have one)

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn
```

```
sudo ldapsearch -x dn
ldapsearch -H ldap:/// -x -s base -b "cn=subschema" objectclasses
| grep -i group
```

```
ldapsearch -H ldap://127.0.0.1  -x -s base -b "cn=subschema"
objectclasses
```

See all the LDAP with all detail

```
sudo slapcat  -b dc=weinert-automation,dc=de
```

The (.ldif) output of this command may be stored as backup file and fed (<) to slapadd for restore.

Info on one LDAP group or one user

```
ldapsearch -x -b dc=weinert-automation,dc=de  -s sub cn=web_admin
ldapsearch -x -b dc=weinert-automation,dc=de  -s sub "uid=ftp22"
```

Make and delete LDAP user or group

Making one or more users etc. is best done by preparing and applying .ldif:

```
ldapadd -x -D cn=admin,dc=weinert-automation,dc=de -W -f
admin/ldap/batch/add_ftp22.ldif
```

Deleting may require the full DN even when uids are unique:

```
ldapdelete -x -D cn=admin,dc=weinert-automation,dc=de -W
"uid=ftp22,ou=People,dc=weinert-automation,dc=de"
```

Add user(s) to LDAP group

```
ldapmodify -x -D cn=admin,dc=weinert-automation,dc=de -W -f
admin/ldap/batch/add_ftp22toWebAdmin.ldif
```

```
# ~/admin/ldap/batch/add_ftp22toWebAdmin.ldif
# Add user:   ftp22 into goup web_admin

dn: cn=web_admin,ou=Groups,dc=weinert-automation,dc=de
changetype: modify
add: memberUid
memberUid: ftp22
```

Change user's home directory

Suppose we have a user (ftp22 in example) with a standard home directory (/home/<user>)

```
ldapmodify -x -D cn=admin,dc=weinert-automation,dc=de -W -f
admin/ldap/batch/changeHome_ftp22.ldif
```

```
# ~/admin/ldap/batch/changeHome_ftp22.ldif
# Change home directory of user  ftp22 to \var\www

dn: uid=ftp22,ou=People,dc=weinert-automation,dc=de
changetype: modify
```

```
replace: homeDirectory
homeDirectory: /var/www
-
replace: loginShell
loginShell: /bin/false
```

```
sudo rm -r /home/ftp22
```

The standard home so destroyed should be skeleton with no valuable files. And of course the user should / must have access to the new home, often best organised qua group, for example by:

```
sudo chown root:web_admin -R /var/www/
sudo chmod -R g+wr /var/www/
sudo chmod g+wrx,o+rx $(find /var/www -type d)
```

Make OS users and groups

First have or check a concept of group numbering. The 1000++ below is just an example.

```
getent group
```

Put albrecht additionally in svn_admin:

```
sudo groupadd -g 1002 svn_admin
sudo usermod -a -G svn_admin albrecht
```

Make a a group guest with a user guest, that can't log in, and make a password (best also guest) that can't be changed by this user (not in ten years):

```
sudo groupadd -g 1003 guest
sudo useradd -c "Domain guest" -g guest -s /bin/false guest
sudo usermod -a -G guest guest
sudo passwd guest
sudo passwd guest -n 3660

sudo deluser --remove-home guest ## removes user guest (sec risk)
```

Have a look at certificates and keys

Check a key and print it

```
sudo openssl rsa -in /etc/ssl/private/vsftpd.pem -check
```

Check a certificate:

```
openssl x509 -in /etc/ssl/certs/ssl-cert-vsftpd.pem -text -noout
```

Dark on light

We like dark letters on light background. Most Linux programmes love "anything on black". One of the worst error prone examples are nano's bright yellow strings unreadable in a decently set terminal. Change them by:

```
sudo nano  /usr/share/nano/sh.nanorc
```

The colours in this file are plain text and their usage is quite clear by comments and regular expressions where to append to.

There are recipes for changing unreadable colours in ls/dir, but that configuration seems voodoo.

Text with Linux' "new line"

Linux having another form of new line in pure text files than Windows, nowadays, seems less problematic than in the past. Nevertheless, when editing/preparing scripts on a Windows workstation one gets (many) errors:

```
-bash: '\r': command not found
```

The only sure way to get rid of this absurdity:

```
sudo fromdos theSriptFile ## with Windows new lines
```

If not available get it and the inverse function todos by:

```
sudo apt-get update
sudo apt-get install tofrodos
```

Have a look at Apache 2

Checking configuration:

```
sudo apache2ctl configtest
```

See all enabled modules:

```
sudo apache2ctl -M
sudo apachectl -t -D DUMP_MODULES
```

The sudo is needed when having TLS and the user issuing the command cannot read the private key file. It is then considered non-existing. The question "Which modules are activated?" gets no answer. Confusingly, a syntax error in a perfect configuration file is reported.

Disabling and enabling installed modules:

```
sudo a2dismod imagemap
sudo a2enmod authnz_ldap
sudo service apache2 restart
```

Disabling and enabling sites respectively configurations:

```
sudo a2dissite imagemap
sudo a2ensite authnz_ldap
sudo service apache2 restart
```

Checking some configuration:

```
cat /etc/apache2/dav_svn.authz
cat /etc/apache2/mods-available/dav_svn.confdi
cat  /megaRaid/Extra/sites/html/conf/svnindex.xsl
cat  /megaRaid/Extra/sites/html/conf/svnindex.css
```

See want went on or wrong:

```
cat /var/log/apache2/access.log
cat /var/log/apache2/error.log
```

Stop and start:

```
sudo service apache2 stop
sudo service apache2 start
sudo service apache2 restart
```

Apache user:

```
htpasswd -b  /etc/apache2/bestUsers ovo dockDockDelle ## make ovo
htpasswd -D /etc/apache2/bestUsers gast              ##delete user
```

Abbreviations

24/7	24 hours on 7 days a week; uninterrupted service (by hardware or software)
AD	Active Directory
BIOS	Basic Input/Output System; Basic drivers and start-up software, commonly in ROM
CSS	Cascading style sheet
DAV	Distributed Authoring and Versioning
DC	Domain controller (of an AD domain)
DHCP	Dynamic Host Configuration Protocol
DIT	Directory Information Tree; for LDAP (and X.500)
DNS	Domain Name System; also used in the sense of domain name server.
FTP	File Transfer Protocol; RFC1579
GUI	Graphical user interface / graphical HMI
HMI	Human Machine Interface (without political correctness formerly MMI)
ID	Identity; in the sense of a domain's ID management
LAN	Local Area network; here in the sense of just Ethernet
LDAP	Lightweight Directory Access Protocol; V:3 RFC 4511
LDIF	LDAP data interchange format; RFV2859
LSI	Corporation; Calif. based manufacturer of storage systems; then Avago; now Broadcom
MS	Microsoft
NTFS	NT File System; full featured file system with fine grained access rights, links and all else used on all Windows NT inheritors
OS	Operating System; run time
PAM	Pluggable authentication module
PBIS	PowerBroker Identity Services
PHP	Personal Home Page Tools
RAID	Redundant Array of inexpensive or Independent disk-Drives
RDP	Remote Desktop Protocol; from Microsoft
RFC	Request for comment; internet standard
ROM	Read only memory; storage for fixed values
sic!	so, exactly so (even if unbelievable) or wrong in the (cited) source already
SSL	Secure Sockets Layer; former name of TSL
SVN	Subversion; versioning (revision control) system for files
TLS	Transport Layer Security; RFC6176
UEFI	Unified Extensible Firmware Interface; OS-BIOS interface
UUID	Universally Unique Identifier
W10	MS Windows 10
xRDP	Linux' RDP (particulate) adaptation on the (X) server side

References

[1] Ubuntu Server guide, 2016 for 16.04
help.ubuntu.com/lts/serverguide/serverguide.pdf

[2] slapd.access - access configuration for slapd, the stand-alone LDAP
this is referred to in [1] but describes configuration files and procedures
non- existent in the actual openLDAP version for Ubuntu 16.04
manpages.ubuntu.com/manpages/xenial/en/man5/slapd.access.5.html

[3] Bug: openLDAP read/admin fails in several ways (from beginning / freshly installed)d
bugs.launchpad.net/ubuntu/+source/apparmor/+bug/1392018

[4] serverfault.com/questions/737889/ldap-modify-insufficient-access-50-for-cn-config-as-h-ldapi-y-external
ldap_modify: Insufficient access (50) for cn=config

[5] Insufficient Access Rights when applying LDIF
stackoverflow.com/questions/30404788/error-50-insufficient-access-rights-when-applying-ldif-openldap

[6] How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04
www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04

[7] Ubuntu, Description of /etc/group - the list of Ubuntu (non LDAP) groups
manpages.ubuntu.com/manpages/wily/man5/group.5.html

[8] Ubuntu, Description of /etc/passwd - the list of Ubuntu (non LDAP) users
manpages.ubuntu.com/manpages/wily/man5/passwd.5.html

[9] Ubuntu, Description of /etc/shadow - optional extension to /etc/passwd
manpages.ubuntu.com/manpages/wily/man5/shadow.5.html

[10] Mendel Cooper, Advanced Bash-Scripting Guide, 2014 (abs-guide.pdf)
tldp.org/LDP/abs/abs-guide.pdf

[11] Apache 2 and HTTP Authentication with PAM
icephoenix.us/linuxunix/apache-and-http-authentication-with-pam/

[12] Linux Server, Das umfassende Handbuch, 3. Auflage 2014
ISBN 978-3-8362-3511-2, Galileo Press, Bonn

[13] Jack Wallen, How to join a Ubuntu machine to a Windows domain, 2010
linux.com/learn/how-join-ubuntu-machine-windows-domain

[14] Beyond Trust, PowerBroker Identity Services, Samba Integration Guide, Feb. 2026

[26] Albrecht Weinert, Ubuntu for remote services, Detours, March 2017,
Supplement to this paper [29]: a-weinert.de/pub/ubuntu4remoteDetours.pdf

[27] Albrecht Weinert, Enable log-out from Apache 2.4, Report, March 2017,
Addendum to this paper [29]: a-weinert.de/pub/enableApache24logout.pdf

[28] Albrecht Weinert, Make a Linux server Active Directory member, Report, February 2017,
Addendum to this paper [29]: a-weinert.de/pub/makeUbuntuServerADmember.pdf

[29] Albrecht Weinert, Ubuntu for remote services, Report, November 2016,
This paper (the last actual version): a-weinert.de/pub/ubuntu4remoteServices.pdf

[30] Albrecht Weinert, Ubuntu for docker, Report, June 2017,
Addendum to this paper [29]: a-weinert.de/pub/ubuntu4docker.pdf

[31] Albrecht Weinert, Raspberry for remote services, Report, May 2017,
a-weinert.de/pub/raspberry4remoteServices.pdf

Table of Content

Dr. Albrecht Weinert is computer science professor at
Bochum University of Applied Sciences or Hochschule Bochum.
He is founder and director of
MEVA-Lab – Laboratory for versatile distributed applications –
as well as of the service provider weinert - automation.
albrecht@a-weinert.de

Rev. 38 25.08.2017