Abandoned projects

obsolet unfinished

- Description of the machine
- Backup to USB Drive
- PWM Password Management
    - Introduction
    - Installation
    - Configuration
- LDAP Account Manager
    - Introduction
    - Installation on Turnkey Debian
    - Installation on Ubuntu Server 22.04
    - Configuration of the LDAP Account Manager
- Bacula
    - Introduction Bacula
    - Installation server components
    - Configuration
    - Evaluation
- NEVIS
    - Installation of NEVIS

Description of the machine

Hardware

Gigabyte embedded AMD Epyc

OS

MX Linux 23.2 Libretto fluxbox based on Debian Linux 12 bookworm

Configuration

Installation

Straight forward using these settings:

- Keyboard German
- encrypted disk with password
- filesystem btrfs
- disk with swap partition

Added Software

- VNC Server
  The Desktop can be remotely controlled by either VNC or IPMI (HTML5, MegaRAC)

  Install VNC on Manjaro

  Installation of VNC Server on MX Linux
- SSH server
- Bitwarden
- conky (not really necessary)

Tweaks

join the sync queue of Firefox for the favorites

misc

default file manager Thunar

Add pci=noaer to the default Kernel parameter in the grub configuration

USB Drive mounting

This is not yet finally. When one of those drives is not available, the system will hang at system start.

I created two mount points:

- /mnt/lacie
- /mnt/armorlock

https://linuxconfig.org/automatically-mount-usb-external-drive-with-autofs

I added these two lines to /etc/fstab:

```
UUID=A9B7-5D47 /mnt/armorlock exfat
auto,nofail,rw,relatime,fmask=0022,dmask=0022,iocharset=utf8,errors=remount-ro 0 0
UUID=FFDF-F997 /mnt/lacie exfat
auto,nofail,rw,relatime,fmask=0022,dmask=0022,iocharset=utf8,errors=remount-ro 0 0
```

Then I executed these commands:

```
systemctl daemon-reload
mount -a
```

Description of networking

This machine uses altogether 4 network interfaces:

- hidden network interface for IPMI

- Internal Intel I210 Gigabit (only used during installation)
- Internal Intel I210 Gigabit (unused)
- USB to Ethernet Adapter 2.5 GBit/s

Backup to USB Drive

Introduction

For same applications it might be necessary to use an external drive without adding this drive to a ZPool. E.g. if you want to copy from or to an external device. TrueNAS did not play well with this USB Bavkup solution. So finally I created another Hardware with a Linux client OS (MXLinux) to get the job dons.

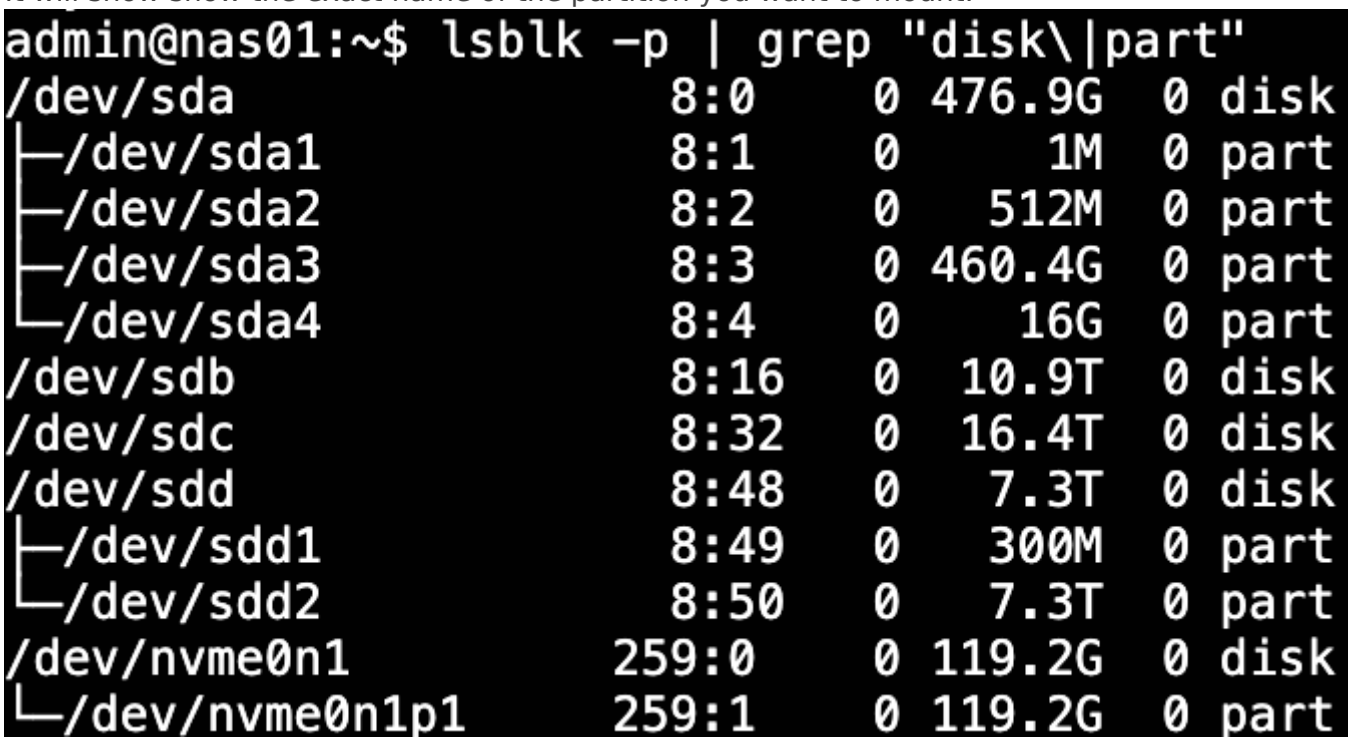
Setup / mounting

TrueNAS will not mount a drive automatically when plugged into an USB Port. This has to be done manually. In this example I will use an external drive from LaCie.

1. Plug in the drive to any USB Port
2. Figure out the name of the device. It can be seen in Storage --> Disks. It is usually the drive without pool.
3. Enter

   ```
   lsblk -p | grep "disk\|part"
   ```

   It will show show the exact name of the partition you want to mount.



In this case it is sdd2.

4. Enter

```
blkid /dev/sdd2
```

It will show you the UUID of the partition you want to mount

```
root@nas04[/home/admin]# blkid /dev/sdd2
/dev/sdd2: LABEL_FATBOOT="EFI" LABEL="EFI" UUID="B7D1-A689" BLOCK_SIZE="512" TYPE="vfat"
PARTUUID="5545caa6-b0c3-4558-b222-aac5fb9c0026"
```

5. Create a mountpoint

```
mkdir /mnt/LaCie
```

6. add to fstab

```
UUID=B7D1-A689 /mnt/LaCie vfat
rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-
1,shortname=mixed,errors=remount-ro 0 0
```

7. Mount the device

```
mount -a
```

Explanation

It seems to be awkward to make so many steps to mount an USB device. However, TrueNAS does no auto mount. So a permanent mount must be added manually to the fstab. And furthermore, TrueNAS seems to change the name of the partition frequently, so the UUID of the partition has to be used.

Create an rsync job

Create the file /root/rsync_exclude.txt with this content:

```
ix-applications
replika
.~tmp~
```

```
*/._*
*/.DocumentRevisions-V100/
*/.DS_Store
*/.fseventsd/
*/.Spotlight-V100/
*/.TemporaryItems/
*/.Trashes/
.@*
.*
@Recycle
*.@__thumb
sync.ffs_lock
```

All these files/directories will not be copied to the target drive. These items are created by MacOS and will automaticall re-created, when these objects in the backup are used by MacOS.

The command for the rsync job looks like this:

```
rsync -av --delete --log-file="/var/log/rsyncd.LaCie.log" --no-perms --no-owner --no-group --exclude-from "/root/rsync_exclude.txt" /mnt/N4pool/ /mnt/LaCie/backup
```

If you want to run it over the network:

create a passwordless ssh connection

Enable ssh login with a public key

```
rsync -av --delete --log-file="/var/log/rsyncd.LaCie.log" --no-perms --no-owner --no-group --exclude-from "/root/rsync_exclude.txt" rsync@nas04.simmy.ch:/mnt/N4pool/ /mnt/lacie/backup
```

Add to cron

```
sudo crontab -e
```

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
```

```
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 1 * * * /root/nas04_backup.sh
0 3 * * * /root/hcloud_backup.sh
0 1 * * * /root/dyndns.sh
0 4 * * * /root/hcloud2_backup.sh
```

Job descriptions

Currently therea are four jobs executed:

Job	**description**
nas04_backup.sh	Backup all data in unencrypted from nas04 to external USB Drive
hcloud_backup.sh	old unencrypted backup of hCloud
hcloud2_backup.sh	backup from hCloud on nas02 to encrypted exteranal usb drive
dyndns.sh	Update for hosting.de DynDNS service

hcloud2_backup.sh

The data is located on nas02.simmy.ch in an encrypted dataset. rsyncd is installed and configured on nas02.simmy.ch. This docker container is not listening on port 22, but on port 30026. Therefore it was necessary to modify the rsync job:

```
rsync -av --delete --log-file="/var/log/rsyncd.armorlock2.log" --no-perms --no-owner --no-group --exclude-from "/root/rsync_exclude.txt" rsync://nas02.simmy.ch:30026/hcloud_simmy /mnt/armorlock/backup/hcloud2
```

Useful links

rsync documentation

Manually mount a USB drive in the Linux terminal

How To Use Rsync to Sync Local and Remote Directories

How to use UUID to mount a volume

PWM Password Management

Introduction

Introduction

PWM is an open source password self-service application for LDAP directories

It includes:

- LDAP Directory Support
- Change Password module for Self-Service
- Account Activation / First time password assignment
- Password reset
- User registration

Multiple Deployment Options

- Java WAR file (bring your own application server, tested with Apache Tomcat)
- Java single JAR file (bring your own Java VM)
- Docker container

Multiple SSO options

- Basic Authentication
- HTTP header username injection
- Central Authentication Service (CAS)
- OAuth client

REST Server APIs for most functionality

- Password set
- Forgotten password
- Password policy reading
- User attribute updates
- Password policy verification

Installation / Architecture

The service is installed on portainer.simmy.ch. It is simply the default installation of the docker container.

Useful links

Install experience

https://github.com/pwm-project/pwm

https://groups.google.com/g/pwm-general?pli=1

https://www.pwm-project.org/pwm/public/reference/

Downloads

Installation

Introduction

There three different ways for the installation. I choose the docker deployment.

Requirements

- Debian Linux Server
- pre-installed docker

Installation

The PWM docker image includes Java and Tomcat. It listens using https on port 8443, and has a volume exposed as /config. You will need to map the /config volume to some type of persistent docker volume for PWM to retain configuration.

Download the newest version

Goto https://github.com/pwm-project/pwm/releases

find and download the most recent .tar file. In my case it was pwm-docker-image-2.0.6.tar.

Load the docker image

Load your docker image with image name of default pwm/pwm-webapp:

```
docker load --input=pwm-docker-image-v2.0.0.tar
```

Create file structure

I worked in the root path.

```
mkdir pwm-config
```

This subdirectoy will become very useful, hence there will be all fiels for configuration and debugging puposes.

Create docker image

Create docker image named mypwm, map to the server's 8443 port, and set the config volume to use the server's local file system /home/user/pwm-config folder (this will be the PWM application path for the container):

```
docker create --name mypwm -p '8443:8443' --mount 'type=bind,source=/root/pwm-config,destination=/config' pwm/pwm-webapp
```

Start the mypwm container:

```
docker start mypwm
```

Configuration

Introduction

After the installation it is necessary to configure several parameters and options to ensure the system works properly.

How To change values

The PWM can run in two different modes:

- config read only
- config editable

New registration are only working in the read only mode. If there is the need to change any configuration setting, the PWM config has to be set to editable. To do this, got to the directory /root/pwm-config, edit the file PwmConfiguration.xml and change the following property:

```
    <property key="configIsEditable">false</property>
```

Fortunately this is the first property of the file.

The key has to be changed from false to true. Save the file and exit the editor. After that open the webpage https://portal.simmy.org and you will find on the upper right corner a new menu, that enables you to edit any configuration setting. When you finished editing, safe. All changes are written to the file PwmConfiguration.xml. Reopen the file PwmConfiguration.xml and change the property Key configIsEditable to true.

> In theory all settings can be changed directly in of the file PwmConfiguration.xml. The changes are applied immediately to the application. However, this is not recommended.

Password policy

It appears that the solution here (in case anyone else ever runs into this) is to change Settings...Password Settings...Password Policy Source to "Local".

https://groups.google.com/g/pwm-general/c/dQN9irsCZ2w/m/ESp9RLfdCAAJ

Valid E-Mail address

The original settings did not allow to enter E-mails with "_"s. So I had to change the corresponding regex that checks the entered E-Mail address for valid characters. I simply added the "_" to the list of allowed characters.

```
Regex:^[a-zA-Z0-9 .,'@]*$


Regex:^[a-zA-Z0-9_ .,'@]*$
```

Bug at user registration

For some reasons the Token that is sent out by pwm gets changed by some web handlers or the E-mail software itself. I could at least partially solve it by overriding some defaults directly in the file PwmConfiguration.xml:

```
        <setting key="pwm.appProperty.overrides" modifyTime="2024-02-21T16:26:32Z"
syntax="STRING_ARRAY" syntaxVersion="0">
            <label>Settings    ^g    Application    ^g    Application    ^g    App Property Overrides</label>
            <value>security.http.permittedUrlPathCharacters=^[a-zA-Z0-9-_=]*$</value>
        </setting>
```

If there is still an error message, just press enter.

LDAP Account Manager

Introduction

What is the LDAP account manager?

LDAP Account Manager (LAM) is a web frontend for managing entries (e.g. users, groups, DHCP settings) stored in an LDAP directory. LAM was designed to make LDAP management as easy as possible for the user. It abstracts from the technical details of LDAP and allows persons without technical background to manage LDAP entries. If needed, power users may still directly edit LDAP entries via the integrated LDAP browser.

LDAP Account Manager

Features

The most important account types which are supported by LAM are Samba, Unix, Zarafa and PPolicy. The user can define profiles for all account types to set default values. Account information can be exported as PDF files. There is also the possibility to create users via file upload. It also includes the tree view of PhpLDAPadmin to access the raw LDAP attributes. LAM is translated to 16 languages.

Supported account types:

- Unix
- Samba 3,4
- Kolab
- Address book entries
- Asterisk (incl. voicemail and Asterisk extensions)
- Mail routing
- IMAP mailboxes (non-LDAP, via IMAP protocol)

- Hosts
- FreeRadius
- Authorized services
- SSH keys
- File system quota (in LDAP (systemQuotas) and via external script)
- DHCP entries
- NIS netgroups

Installation on Turnkey Debian

Installation

```
apt -y install ldap-account-manager
```

The account manager is available on http://lamp.simmy.ch/lam.

Useful links

https://www.unixmen.com/setup-samba-domain-controller-with-openldap-backend-in-ubuntu-13-04/

https://www.ldap-account-manager.org/lamcms/howto

https://computingforgeeks.com/install-and-configure-ldap-account-manager-on-ubuntu/

https://www.ldap-account-manager.org/lamcms/documentation

Installation on Ubuntu Server 22.04

Install Apache Webserver and PHP

```
apt -y install apache2 php php-cgi libapache2-mod-php php-mbstring php-common php-pear
```

Then enable php-cgi PHP extension:

```
a2enconf php*-cgi
systemctl reload apache2
```

Install LDAP Account Manager

```
apt -y install ldap-account-manager
```

The account manager is available on http://lam.simmy.ch/lam.

Useful links

https://www.unixmen.com/setup-samba-domain-controller-with-openldap-backend-in-ubuntu-13-04/

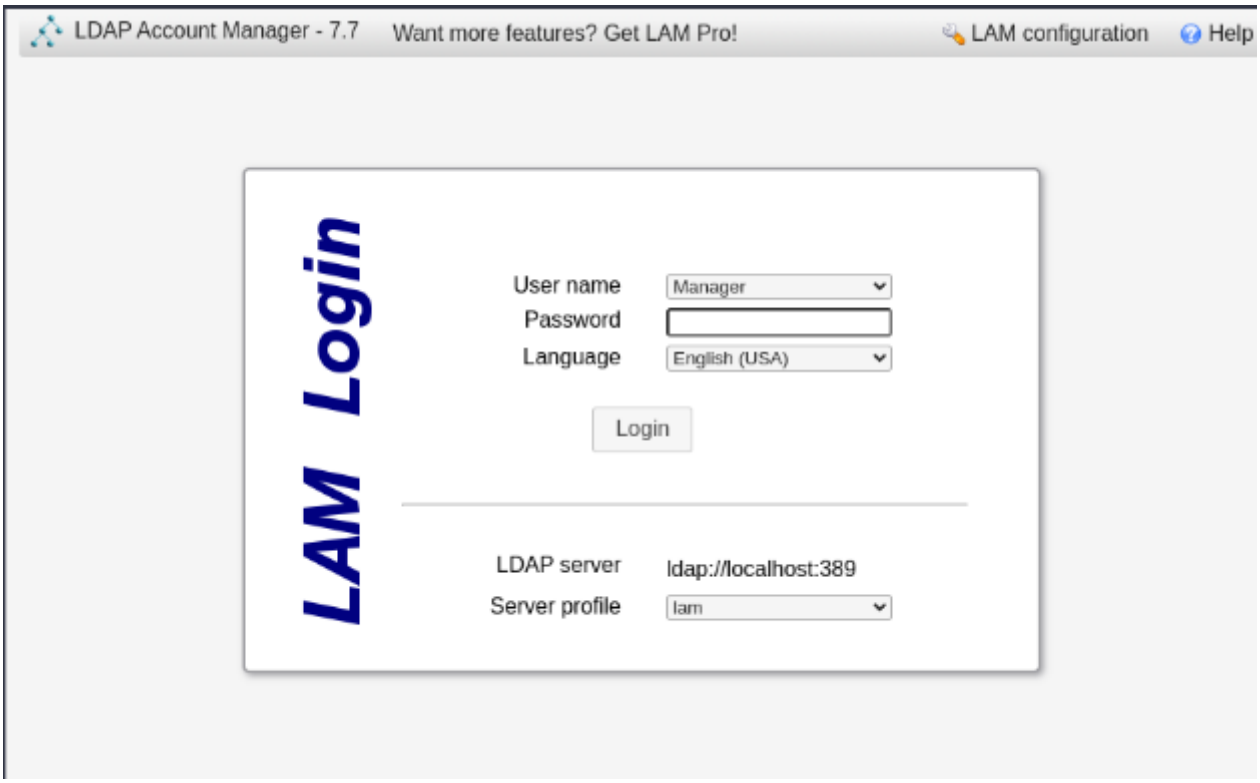
https://www.ldap-account-manager.org/lamcms/howto

https://computingforgeeks.com/install-and-configure-ldap-account-manager-on-ubuntu/

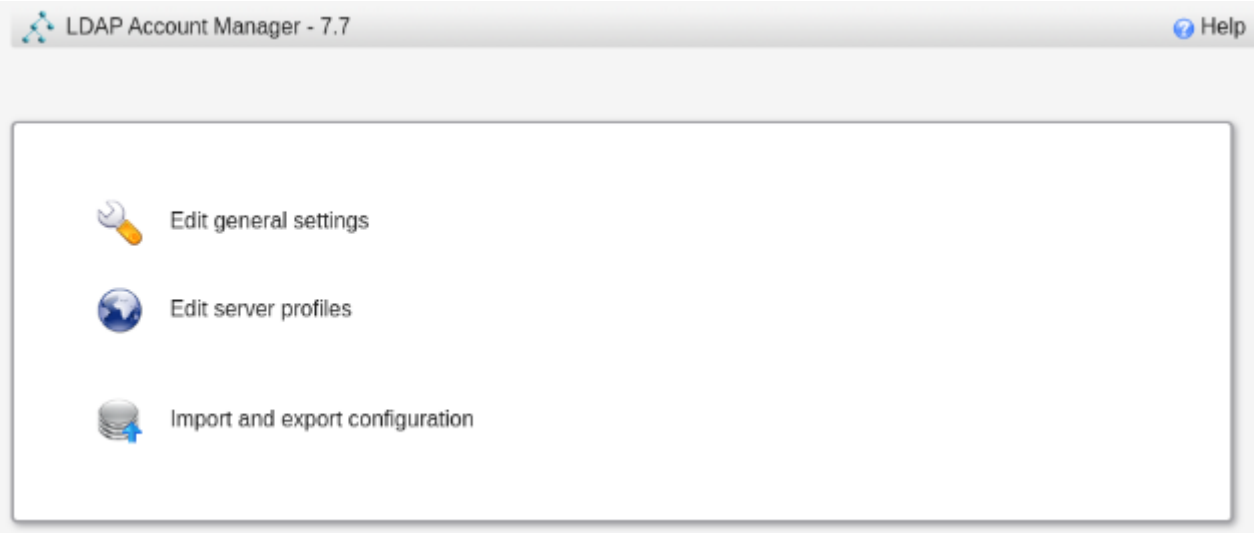
https://www.ldap-account-manager.org/lamcms/documentation

Configuration of the LDAP Account Manager

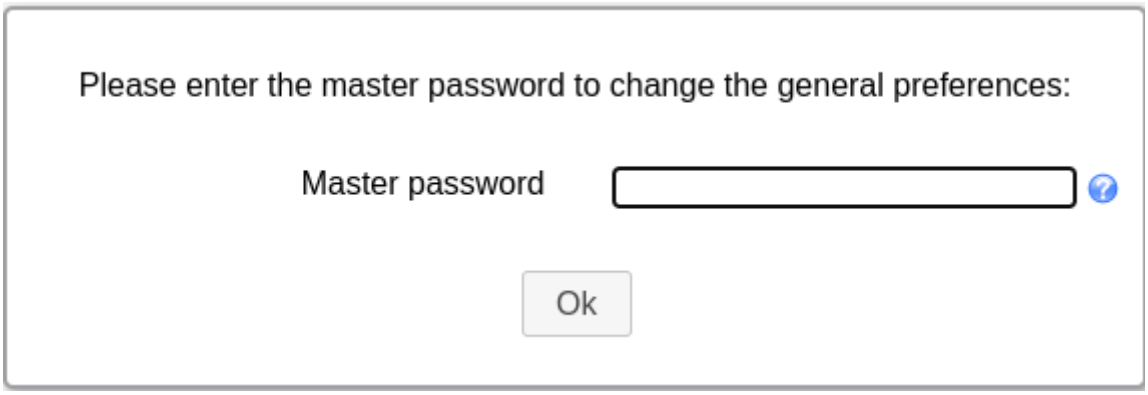
Change master password



Click on LAM configuration on the upper right corner.



"Edit general settings"

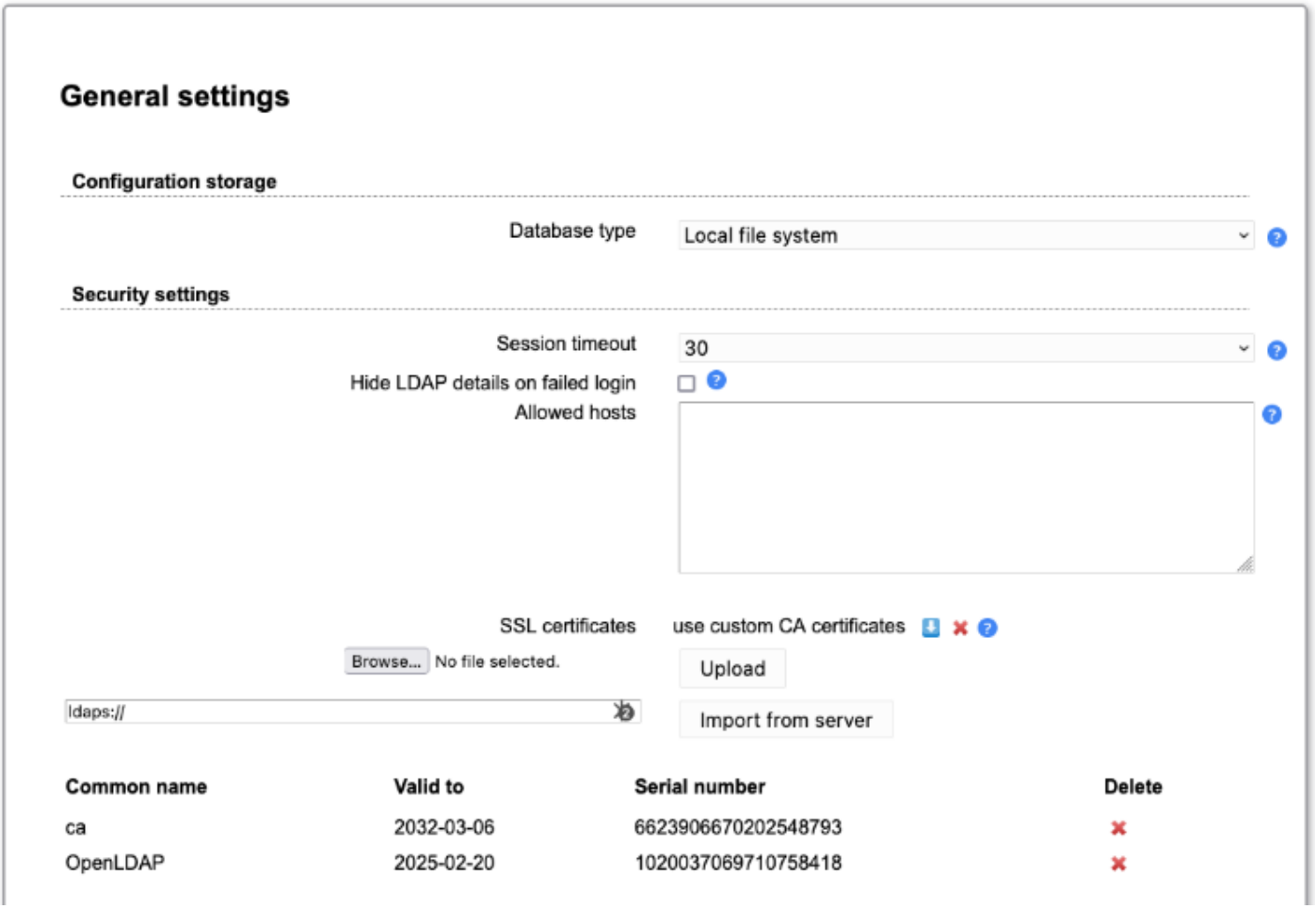


The Master password is "lam".

Scroll down to "Change master password" and enter your desired password two times.

> The password will be saved in cleartext in a configuration file of LAM

Add certificates



The communication with the the OpenLDAP server over SSL didn't work. So finally I added two certificates. The CA, which I simply uploaded (Choose file --> "Upload") and the certificate of the Domain Controller (enter ldaps://openldap.simmy.ch --> "Import from server").

Scroll down and click "Ok". Restart the apache server:

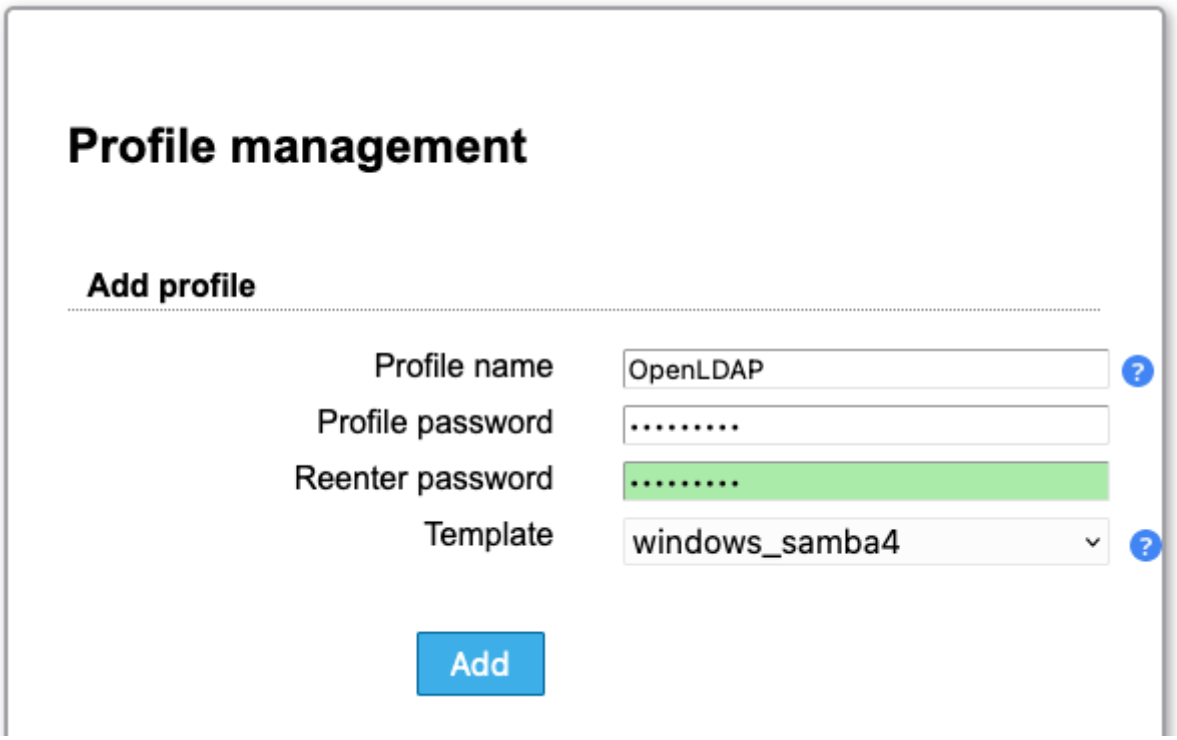
```
systemctl restart apache2
```

Create a profile for OpenLDAP

Click on LAM configuration on the upper right corner.

Click on "Edit server profiles".

Click on "Manage server profiles".

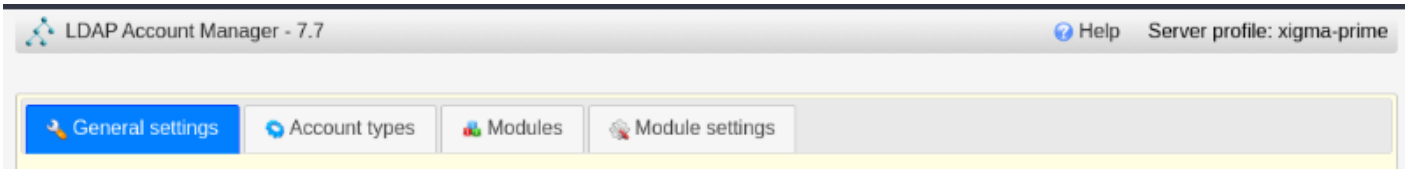


Enter these options:

1. Profile name --> *OpenLDAP*
2. Profile password --> *your password here*
3. Reenter password --> *your password here*
4. Template --> choose Template "*unix*" for OpenLDAP
5. Add

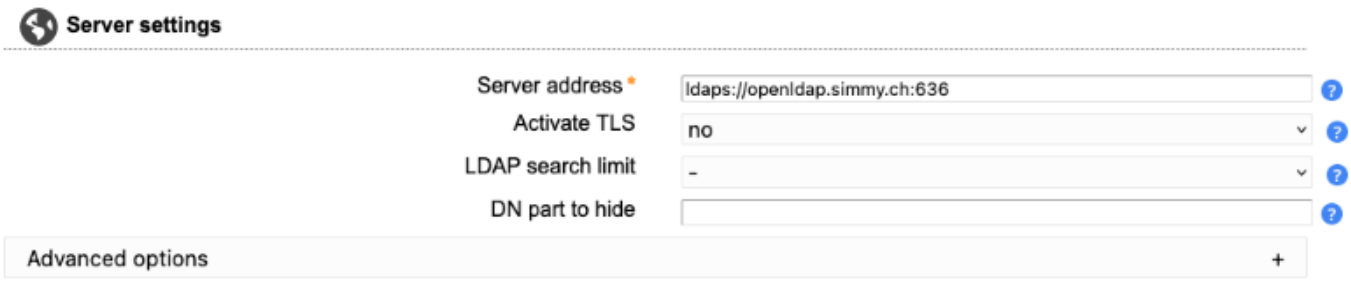
> The password will be saved in cleartext in a configuration file of LAM

Configuration of the profile for OpenLDAP



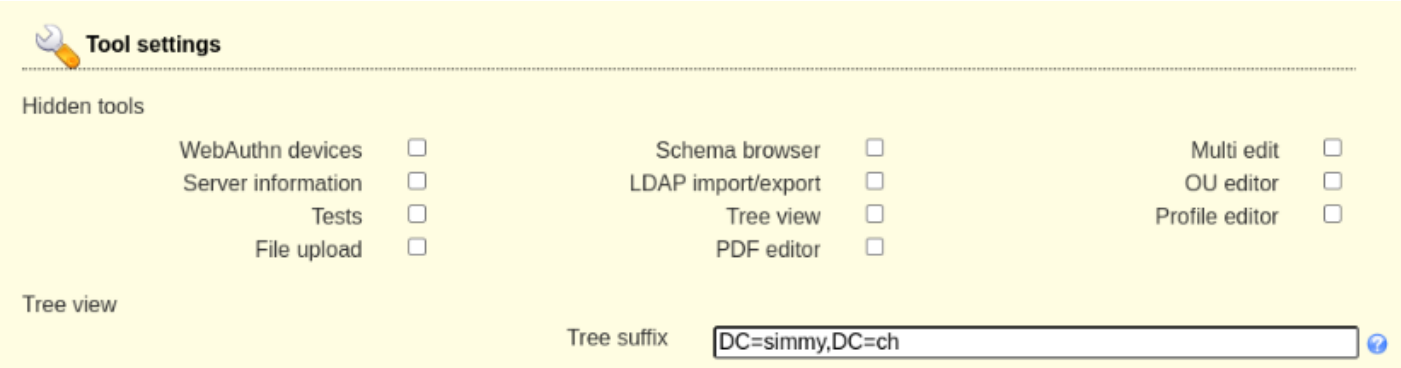
General settings

Server settings



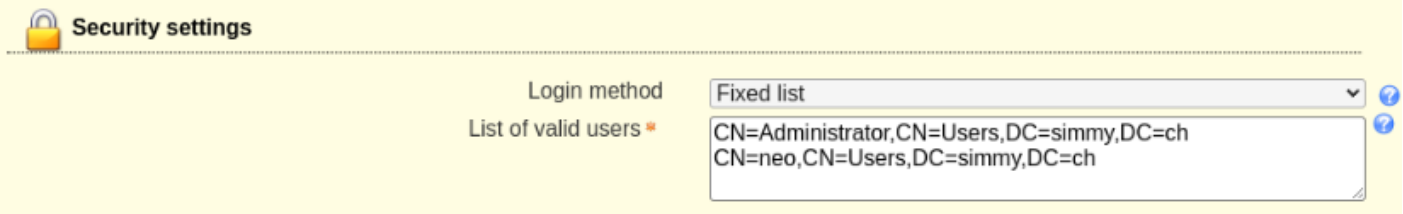
Server address --> ldap://openldap.simmy.ch:389

Tool settings



Tree suffix: DC=simmy,DC=ch

Security settings



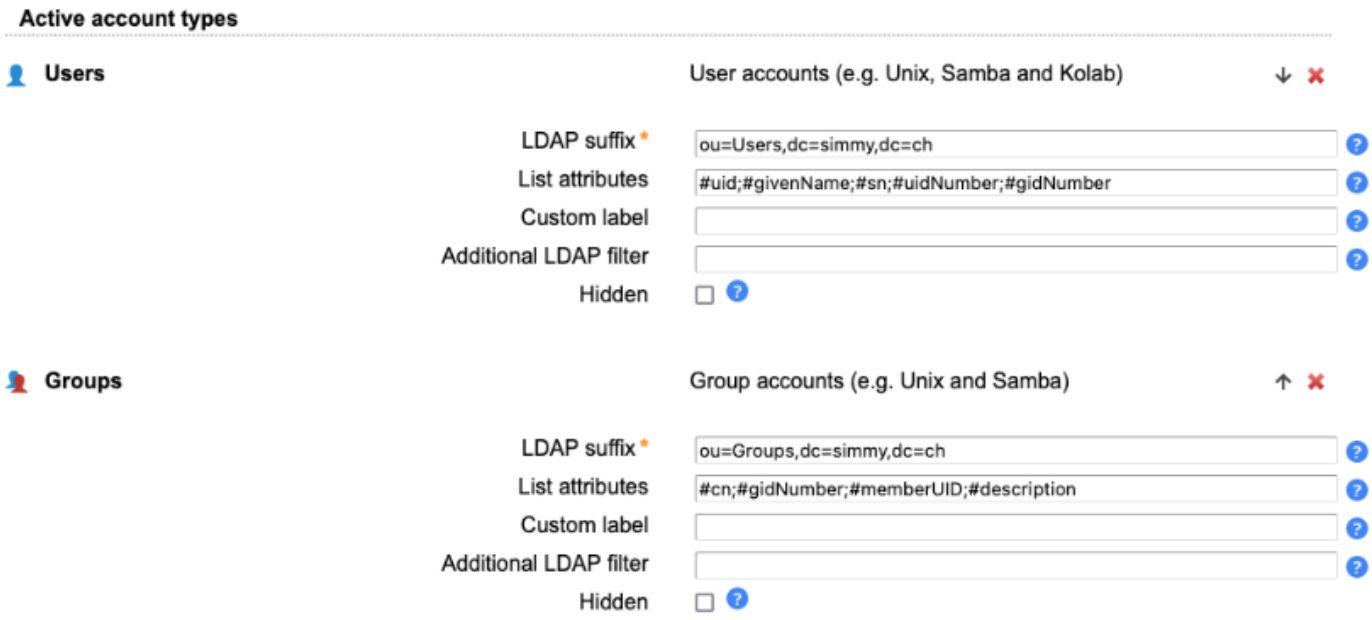
Login method: Fixed list

List of valid users:

```
cn=admin,dc=simmy,dc=ch
cn=binduser,ou=Users,dc=simmy,dc=ch
cn=Holger Schindler,ou=Users,dc=simmy,dc=ch
```

Account types

> Create the OU groups before doing this:



These two LDAP suffixes have to be set:

- *CN=Users,DC=simmy,DC=ch*
- *OU=Groups,DC=simmy,DC=ch*

Modules

Nothing to change here.

Module settings

Nothing to change here.

Final

"Save" and login to your profile "OpenLDAP. You will have to enter the password of the Administrator.

Useful links

https://www.unixmen.com/setup-samba-domain-controller-with-openldap-backend-in-ubuntu-13-04/

https://www.ldap-account-manager.org/lamcms/howto

https://computingforgeeks.com/install-and-configure-ldap-account-manager-on-ubuntu/

https://www.ldap-account-manager.org/lamcms/documentation

https://www.ldap-account-manager.org/static/doc/manual.pdf

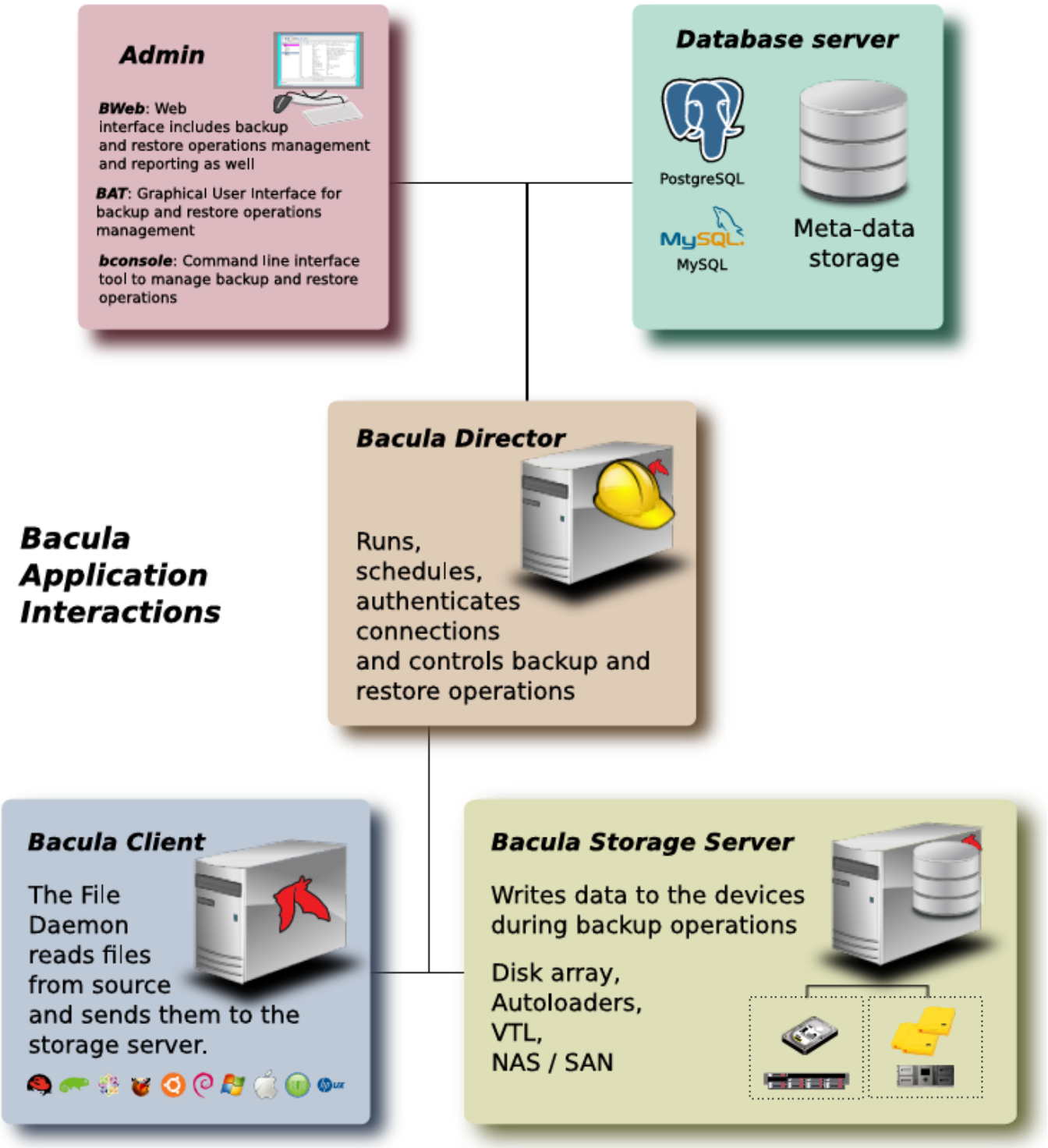
Bacula

Introduction Bacula

Introduction

Bacula is an open-source, enterprise-level computer backup system for heterogeneous networks. It is designed to automate backup tasks that had often required intervention from a systems administrator or computer operator.

Architecture



Components

Bacula Director

Server component. Supervisor.

Bacula Console/Admin

Interface for the Director. There are text versions and GUIs:

- BWeb
- BAT
- bConsole

Bacula Client

File daemon installed on the client.

Bacula Storage

Interface to the storage components. Here a share on the NAS.

Catalog

SQL database.

Bacula Monitor

Monitor program. Works with GTK+ (GNOME, KDE, FreeDesktop.org system tray standard).

Installation server components

Introduction

In general, you should get the binary packages from your download area on www.bacula.org. You can either download what you need or setup a repository pointing to the download area that will allow you to use your installer program such as apt to ensure that all the dependencies a met.

This will install these components:

- Database Server PostgreSQL
- Bacula Director
- Bacula Storage Server

Setup the repository

```
apt-get install apt-transport-https
wget https://bacula.org/downloads/Bacula-4096-Distribution-Verification-key.asc
apt-key add Bacula-4096-Distribution-Verification-key.asc
```

Add to your /etc/apt/sources.list file the following entries:

```
# Bacula
deb https://www.bacula.org/packages/65f518dfc0382/debs/13.0.4 bookworm main
```

Installation

```
apt-get update
apt-get install dbconfig-common postgresql
```

```
apt-get install bacula-postgresql
```

I choose the name of the application as password for the Postgre Database. There is small untilitiy installed with the director: bconsole.

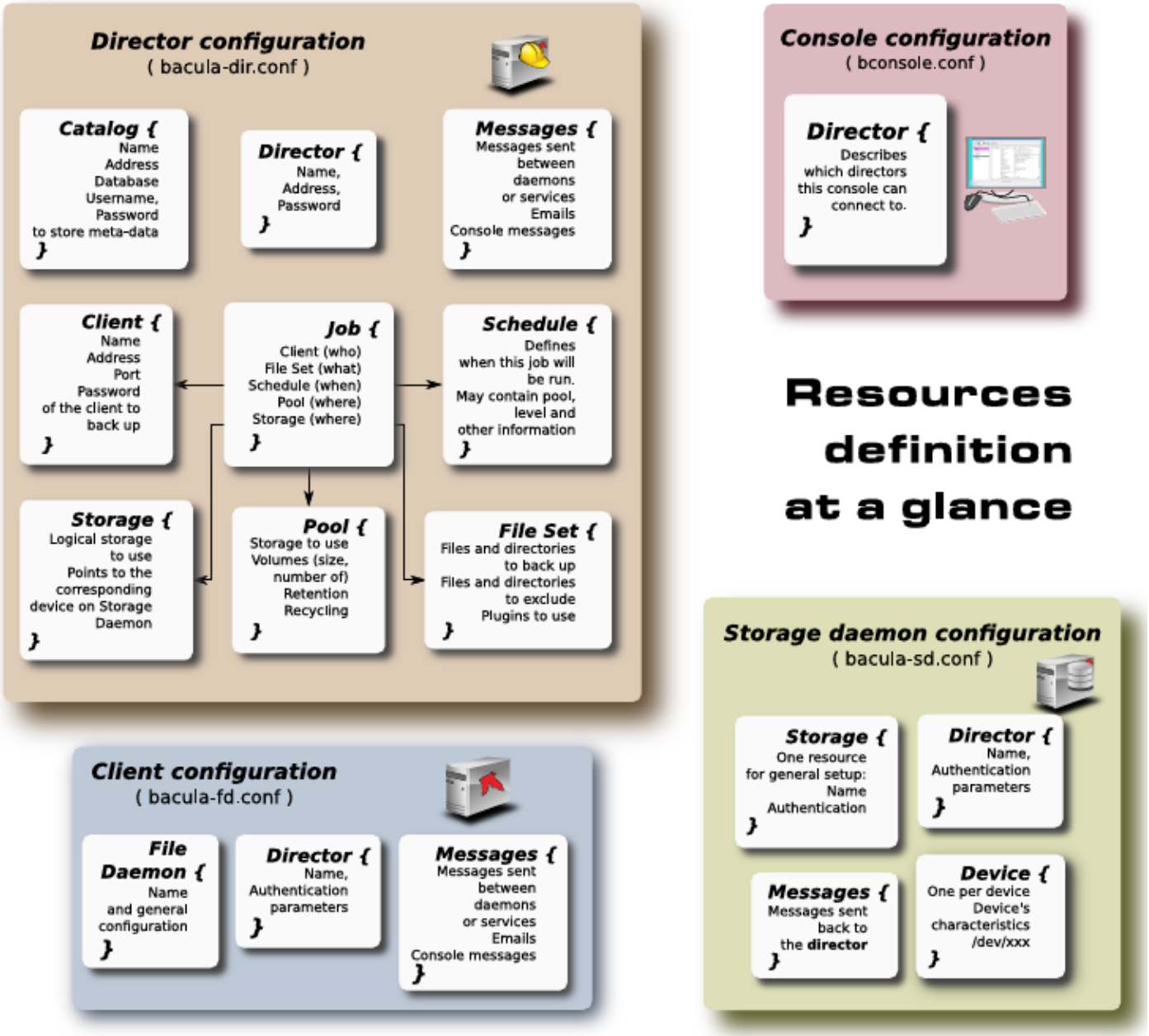
Useful links

bacula binaries

Installation guide

Configuration

The configuration files are located in /opt/bacula/etc/.



Evaluation

Despite the fact that the architecture looks very promising, I finally abandoned the project. There are three reasons:

1. The client app for Archlinux is broken
2. It's difficult to install on Debian
3. The interesting parts of the project are taken by a commercial company (https://www.baculasystems.com/company/)

NEVIS

Installation of NEVIS

Introduction

I decided to install NEVIS inside a kubernetes cluster.

Installation in Kubernetes Cluster

Installation of kubernetes

Fedora installation of kubernetes

```
sudo dnf install kubernetes kubernetes-kubeadm kubernetes-client
```

Open firewall ports 6443, 10250

```
sudo systemctl enable kubelet.service
sudo systemctl enable containerd
sudo systemctl start containerd
sudo swapoff -a
sudo dnf install iproute-tc

sudo cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF



sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
sudo cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
```

```
net.bridge.bridge-nf-call-iptables  = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward                 = 1
EOF



# setting DNS correcly
sudo mkdir -p /etc/systemd/resolved.conf.d/
sudo cat <<EOF | sudo tee /etc/systemd/resolved.conf.d/stub-listener.conf
[Resolve]
DNSStubListener=no
EOF

sudo sysctl --system

sudo systemctl enable --now kubelet

sudo kubeadm init

# set KUBELET_KUBEADM_ARGS
sudo tee -a /etc/kubernetes/kubelet.conf <<EOF
KUBELET_LOG_LEVEL=5
KUBELET_KUBEADM_ARGS="--v=4 --logtostderr=true"
EOF
```

Kubelet configuration

using-kubernetes-kubelet

Accessing the cluster as normal user

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
# Allow the control plane machine to also run pods for applications. Otherwise more than one machine is needed in the cluster.
kubectl taint nodes --all node-role.kubernetes.io/control-plane-

# Install flannel into the cluster to provide cluster networking. There are many other networking solutions besides flannel. Flannel is straightforward and suitable for this guide.
kubectl apply -f https://github.com/coreos/flannel/raw/master/Documentation/kube-flannel.yml
```

Useful commands

```
sudo systemctl restart kubelet
sudo systemctl status kubelet
sudo journalctl -u kubelet
ss -tlnp | grep 6443
kubectl config use-context
kubectl config view
kubectl cluster-info
kubectl get pods --all-namespaces
kubectl get svc -A
kubectl get events --namespace=kube-system
kubectl get nodes -o wide
```

Additional .conf files:

The kubernetes-kubeadm rpm installs an overriding `kubelet` unit file at:

```
/usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf
```

We strongly recommend to **not** modify either file as any changes could be lost during an update.

As documented by the Kubernetes team (https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/kubelet-integration/#the-kubelet-drop-in-file-for-systemd), create the following directory for user managed, system-level systemd `kubelet` overrides:

```
$ sudo mkdir -p /etc/systemd/system/kubelet.service.d/
```

Then create a unit file (`.conf` extension required) and copy the file to the directory listed above. Settings in this file will override settings from either or both of the default systemd files.

misc

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.1.35:6443 --token dapwn1.21bvsun7tw95b6j7 \
--discovery-token-ca-cert-hash
sha256:bc878aa0a8db726627f0be2a9bfbec584bde1156114e1af61aa727e2e39302b5
```