Software/Applikationen

- BTRFS
  - RAID Rebuild
- Nextcloud (OLD)
  - Basis-Installation
  - Faster Previews (Imaginary)
  - AIO-Installation
- LVM
  - Thin-Pool
- Proxmox
  - Remote-Migrate
- Mailcow
  - mailcow.conf Tuneables
  - reset rspamd config/stats/full
  - Verbessertes/Eigenes Spamverhalten
- Prometheus
  - Node Exporter
- Nextcloud AIO
- VyOS

- Build own iSO

BTRFS

RAID Rebuild

OS Disk

> Replace /dev/sda and /dev/sdb if required !!!

Wipe New Disk `wipefs -a /dev/sdb`

Copy Partition Table from other Bootable-Disk to New Bootable-Disk `sfdisk -d /dev/sda | sfdisk /dev/sdb`

NOT DONE YET, now replace the "broken" device with the new one using btrfs tools and scrub after that one time maybe.

Nextcloud (OLD)

Basis-Installation

> for debian 12!

Enviroment

```
PHP_VERSION="8.2"
PG_PASSWORD=$(openssl rand -base64 12)
REDIS_PASSWORD=$(openssl rand -base64 12)
DOMAIN="cloud.example.com"
CERTMAIL="ssl@example.com"
NC_ADMINPWD=$(openssl rand -base64 12)
NC_RELEASE="latest-27"
NC_DATA="/mnt/nc_data"
NC_TMPDIR="$NC_DATA/tmp"
OCC="sudo -u www-data php /var/www/nextcloud/occ"
```

Install

```
apt update -y
apt full-upgrade -y

apt-mark hold apache2 apache2-bin
apt install -y --no-install-recommends php php-
{fpm,ctype,curl,dom,fileinfo,gd,mbstring,xml,xmlreader,xmlwriter,zip,sqlite3,mysql,pgsql,intl,ldap,ftp,imap,bcm
ath,gmp,exif,apcu,memcached,redis,imagick,phar} libmagickcore-6.q16-6-extra ffmpeg nginx aria2 curl wget
unzip bzip2 p7zip p7zip-full postgresql redis certbot python3-certbot-nginx

# PHP Tuning
sed -i 's/^apc.enable_cli.*/apc.enable_cli=1/' /etc/php/$PHP_VERSION/cli/conf.d/20-apcu.ini
sed -i "s/.*clear_env.*/clear_env=no/" /etc/php/$PHP_VERSION/fpm/pool.d/www.conf
sed -i 's/.*max_input_time.*/max_input_time=3600/' /etc/php/$PHP_VERSION/cli/php.ini
sed -i 's/.*max_input_time.*/max_input_time=3600/' /etc/php/$PHP_VERSION/fpm/php.ini
sed -i 's/.*max_execution_time.*/max_execution_time=3600/' /etc/php/$PHP_VERSION/cli/php.ini
sed -i 's/.*max_execution_time.*/max_execution_time=3600/' /etc/php/$PHP_VERSION/fpm/php.ini
sed -i "s/.*memory_limit.*/memory_limit=512M/" /etc/php/$PHP_VERSION/fpm/php.ini
```

```
sed -i "s/.*memory_limit.*/memory_limit=512M/" /etc/php/$PHP_VERSION/cli/php.ini
sed -i "s/.*opcache.enable=.*/opcache.enable=1/" /etc/php/$PHP_VERSION/fpm/php.ini
sed -i "s/.*opcache.enable=.*/opcache.enable=1/" /etc/php/$PHP_VERSION/cli/php.ini
sed -i "s/.*opcache.enable_cli=.*/opcache.enable_cli=1/" /etc/php/$PHP_VERSION/fpm/php.ini
sed -i "s/.*opcache.enable_cli=.*/opcache.enable_cli=1/" /etc/php/$PHP_VERSION/cli/php.ini
sed -i "s/.*post_max_size.*/post_max_size=200G/" /etc/php/$PHP_VERSION/fpm/php.ini
sed -i "s/.*post_max_size.*/post_max_size=200G/" /etc/php/$PHP_VERSION/cli/php.ini
sed -i "s/.*upload_max_filesize.*/upload_max_filesize=200G/" /etc/php/$PHP_VERSION/fpm/php.ini
sed -i "s/.*upload_max_filesize.*/upload_max_filesize=200G/" /etc/php/$PHP_VERSION/cli/php.ini
sed -i "s|.*upload_tmp_dir.*|upload_tmp_dir=$NC_TMPDIR|" /etc/php/$PHP_VERSION/fpm/php.ini
sed -i "s|.*upload_tmp_dir.*|upload_tmp_dir=$NC_TMPDIR|" /etc/php/$PHP_VERSION/cli/php.ini
sed -i "s/.*opcache.interned_strings_buffer.*/opcache.interned_strings_buffer=64/"
/etc/php/$PHP_VERSION/fpm/php.ini
sed -i "s/.*opcache.interned_strings_buffer.*/opcache.interned_strings_buffer=64/"
/etc/php/$PHP_VERSION/cli/php.ini

service php8.2-fpm restart

# PostgreSQL
sudo -u postgres psql <<EOF
CREATE DATABASE nextcloud;
CREATE USER nextcloud WITH PASSWORD '${PG_PASSWORD}';
GRANT ALL PRIVILEGES ON DATABASE nextcloud TO nextcloud;
ALTER DATABASE nextcloud OWNER TO nextcloud;
CREATE SCHEMA IF NOT EXISTS public;
GRANT USAGE ON SCHEMA public TO nextcloud;
EOF

cat << EOF
--- [PostgreSQL] ---
User: nextcloud
Database: nextcloud
Password: ${PG_PASSWORD}
# Bitte Speichern, auch wenn es warscheinlich nicht mehr benötigt wird.
EOF

# Redis
sed -i 's/.*port 6379.*/port 0/' /etc/redis/redis.conf
sed -i "s/.*unixsocket .*/unixsocket \/run\/redis\/redis-server.sock/" /etc/redis/redis.conf
sed -i 's/.*unixsocketperm .*/unixsocketperm 770/' /etc/redis/redis.conf
```

```
sed -i 's/.*maxclients .*/maxclients 10240/' /etc/redis/redis.conf
sed -i "s/.*requirepass foobared.*/requirepass $(echo "$REDIS_PASSWORD" | sed -e 's/[\/&]/\\&/g')/"
/etc/redis/redis.conf
usermod -aG redis www-data
service redis-server restart

cat << EOF
--- [REDIS] ---
Unixsocket: /run/redis/redis-server.sock
Password: ${REDIS_PASSWORD}
# Bitte Speichern, auch wenn es warscheinlich nicht mehr benötigt wird.
EOF

# NGINX
certbot certonly --nginx --non-interactive --agree-tos --email $CERTMAIL -d $DOMAIN

rm -f /etc/nginx/sites-enabled/default
cat << 'EOF' > /etc/nginx/sites-enabled/nextcloud
upstream php-handler {
    server 127.0.0.1:9000;
    server unix:/run/php/php8.2-fpm.sock;
}

# Set the `immutable` cache control options only for assets with a cache busting `v` argument
map $arg_v $asset_immutable {
    "" "";
    default ", immutable";
}

server {
    listen 80;
    listen [::]:80;
    server_name $DOMAIN;

    # Prevent nginx HTTP Server Detection
    server_tokens off;

    # Enforce HTTPS
    return 301 https://$server_name$request_uri;
}
```

```
server {
    listen 443      ssl http2;
    listen [::]:443 ssl http2;
    server_name $DOMAIN;

    # Path to the root of your installation
    root /var/www/nextcloud;

    # Use Mozilla's guidelines for SSL/TLS settings
    ssl_certificate     /etc/letsencrypt/live/$DOMAIN/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/$DOMAIN/privkey.pem;

    # Prevent nginx HTTP Server Detection
    server_tokens off;

    # HSTS settings
    add_header Strict-Transport-Security "max-age=15768000; includeSubDomains; preload" always;

    # set max upload size and increase upload timeout:
    client_max_body_size 0;
    client_body_timeout 3600s;
    client_body_temp_path $NC_TMPDIR;
    fastcgi_buffers 64 4K;

    # Enable gzip but do not remove ETag headers
    gzip on;
    gzip_vary on;
    gzip_comp_level 4;
    gzip_min_length 256;
    gzip_proxied expired no-cache no-store private no_last_modified no_etag auth;
    gzip_types application/atom+xml text/javascript application/javascript application/json application/ld+json
application/manifest+json application/rss+xml application/vnd.geo+json application/vnd.ms-fontobject
application/wasm application/x-font-ttf application/x-web-app-manifest+json application/xhtml+xml
application/xml font/opentype image/bmp image/svg+xml image/x-icon text/cache-manifest text/css text/plain
text/vcard text/vnd.rim.location.xloc text/vtt text/x-component text/x-cross-domain-policy;

    # The settings allows you to optimize the HTTP2 bandwidth.
    # See https://blog.cloudflare.com/delivering-http-2-upload-speed-improvements/
    # for tuning hints
```

```
    client_body_buffer_size 512k;

    # HTTP response headers borrowed from Nextcloud `.htaccess`
    add_header Referrer-Policy                   "no-referrer"       always;
    add_header X-Content-Type-Options            "nosniff"           always;
    add_header X-Frame-Options                   "SAMEORIGIN"        always;
    add_header X-Permitted-Cross-Domain-Policies "none"              always;
    add_header X-Robots-Tag                      "noindex, nofollow" always;
    add_header X-XSS-Protection                  "1; mode=block"     always;

    # Remove X-Powered-By, which is an information leak
    fastcgi_hide_header X-Powered-By;

    # Set .mjs and .wasm MIME types
    include mime.types;
    types {
        text/javascript mjs;
 application/wasm wasm;
    }

    # Specify how to handle directories -- specifying `/index.php$request_uri`
    index index.php index.html /index.php$request_uri;

    # Rule borrowed from `.htaccess` to handle Microsoft DAV clients
    location = / {
        if ( $http_user_agent ~ ^DavClnt ) {
            return 302 /remote.php/webdav/$is_args$args;
        }
    }

    location = /robots.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    # Make a regex exception for `/.well-known` so that clients can still
    # access it despite the existence of the regex rule
    # `location ~ /(\.|autotest|...)` which would otherwise handle requests
    # for `/.well-known`.
```

```
    location ^~ /.well-known {
        # The rules in this block are an adaptation of the rules
        # in `.htaccess` that concern `/.well-known`.

        location = /.well-known/carddav { return 301 /remote.php/dav/; }
        location = /.well-known/caldav  { return 301 /remote.php/dav/; }

        location /.well-known/acme-challenge    { try_files $uri $uri/ =404; }
        location /.well-known/pki-validation    { try_files $uri $uri/ =404; }

        # Let Nextcloud's API for `/.well-known` URIs handle all other
        # requests by passing them to the front-end controller.
        return 301 /index.php$request_uri;
    }

    # Rules borrowed from `.htaccess` to hide certain paths from clients
    location ~ ^/(?:build|tests|config|lib|3rdparty|templates|data)(?:$|/)  { return 404; }
    location ~ ^/(?:\.|autotest|occ|issue|indie|db_|console)                { return 404; }

    # Ensure this block, which passes PHP files to the PHP process, is above the blocks
    # which handle static assets (as seen below). If this block is not declared first,
    # then Nginx will encounter an infinite rewriting loop when it prepends `/index.php`
    # to the URI, resulting in a HTTP 500 error response.
    location ~ \.php(?:$|/) {
        # Required for legacy support
        rewrite ^/(?!index|remote|public|cron|core\/ajax\/update|status|ocs\/v[12]|updater\/.+|ocs-
provider\/.+|.+\/richdocumentscode(_arm64)?\/proxy) /index.php$request_uri;

        fastcgi_split_path_info ^(.+?\.php)(/.*)$;
        set $path_info $fastcgi_path_info;

        try_files $fastcgi_script_name =404;

        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $path_info;
        fastcgi_param HTTPS on;

        fastcgi_param modHeadersAvailable true;         # Avoid sending the security headers twice
        fastcgi_param front_controller_active true;     # Enable pretty urls
```

```
        fastcgi_pass php-handler;

        fastcgi_intercept_errors on;
        fastcgi_request_buffering off;

        fastcgi_max_temp_file_size 0;
    }

    # Serve static files
    location ~ \.(?:css|js|mjs|svg|gif|png|jpg|ico|wasm|tflite|map|ogg|flac)$ {
        try_files $uri /index.php$request_uri;
        # HTTP response headers borrowed from Nextcloud `.htaccess`
        add_header Cache-Control                    "public, max-age=15778463$asset_immutable";
        add_header Referrer-Policy                  "no-referrer"       always;
        add_header X-Content-Type-Options           "nosniff"           always;
        add_header X-Frame-Options                  "SAMEORIGIN"        always;
        add_header X-Permitted-Cross-Domain-Policies "none"             always;
        add_header X-Robots-Tag                     "noindex, nofollow" always;
        add_header X-XSS-Protection                 "1; mode=block"     always;
        access_log off;     # Optional: Don't log access to assets
    }

    location ~ \.woff2?$ {
        try_files $uri /index.php$request_uri;
        expires 7d;         # Cache-Control policy borrowed from `.htaccess`
        access_log off;     # Optional: Don't log access to assets
    }

    # Rule borrowed from `.htaccess`
    location /remote {
        return 301 /remote.php$request_uri;
    }

    location / {
        try_files $uri $uri/ /index.php$request_uri;
    }
}
EOF

sed -i "s|\$DOMAIN|$DOMAIN|g" /etc/nginx/sites-enabled/nextcloud
```

```
sed -i "s|\$NC_TMPDIR|$NC_TMPDIR|g" /etc/nginx/sites-enabled/nextcloud

service nginx restart

# NEXTCLOUD Installation
cd /var/www
wget https://download.nextcloud.com/server/releases/$NC_RELEASE.zip
unzip $NC_RELEASE.zip
rm -f $NC_RELEASE.zip
cd
mkdir -p /var/log/nextcloud $NC_TMPDIR
touch /var/log/nextcloud/default.log /var/log/nextcloud/audit.log /var/log/nextcloud/flow.log
chown -R www-data:www-data nextcloud/ $NC_DATA /var/log/nextcloud $NC_TMPDIR

systemctl restart nginx php8.2-fpm redis-server postgresql
rm -f /var/www/nextcloud/config/config.php
$OCC maintenance:install --database "pgsql" --database-name "nextcloud" --database-user "nextcloud" --
database-pass "$PG_PASSWORD" --admin-user "admin" --admin-pass "$NC_ADMINPWD" --data-dir "$NC_DATA"

## OCC Settings
$OCC config:system:set log_type --value=file
$OCC config:system:set loglevel --value=0 #DEBUG
$OCC config:system:set logfile --value=/var/log/nextcloud/default.log
$OCC config:app:set admin_audit logfile --value=/var/log/nextcloud/audit.log
$OCC config:app:set workflowengine logfile --value=/var/log/nextcloud/flow.log
$OCC config:system:set trusted_domains 0 --value="$DOMAIN"
$OCC config:system:set overwrite.cli.url --value "https://$DOMAIN"
$OCC config:system:set default_phone_region --value "de_DE"
$OCC config:system:set redis host --value=/var/run/redis/redis-server.sock
$OCC config:system:set redis port --value=0
$OCC config:system:set redis dbindex --value=0
$OCC config:system:set redis password --value=$REDIS_PASSWORD
$OCC config:system:set redis timeout --value=1.5
$OCC config:system:set memcache.locking --value="\OC\Memcache\Redis"
$OCC config:system:set memcache.distributed --value="\OC\Memcache\Redis"
$OCC config:system:set memcache.local --value="\OC\Memcache\Redis"
$OCC config:app:set files max_chunk_size --value 20971520
$OCC config:system:set tempdirectory --value $NC_TMPDIR
$OCC config:system:set maintenance_window_start --type=integer --value=1
$OCC db:add-missing-indices
```

```
cat << EOF
--- [Nextcloud] ---
User: admin
Password: ${NC_ADMINPWD}
# Bitte Speichern!
EOF

## CRON
cat << EOF > /etc/cron.d/nextcloud
PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin
LD_LIBRARY_PATH=/usr/local/lib
@weekly root certbot --nginx renew
*/5 * * * * root chown -R www-data:www-data /var/www/nextcloud $NC_DATA /var/log/nextcloud $NC_TMPDIR
*/10 * * * * www-data php -f /var/www/nextcloud/occ files:scan --all --unscanned
* * * * * www-data php -f /var/www/nextcloud/cron.php
EOF
systemctl enable --now cron

## ENVIROMENT
cat << EOF > /etc/profile.d/100-nextcloud.sh
OCC="sudo -u www-data php /var/www/nextcloud/occ"
EOF

# Firewall und Sicherheit!
apt install ufw fail2ban -y
ufw limit 22/tcp
ufw allow 80/tcp
ufw allow 443/tcp
echo "y" | ufw enable
```

Faster Previews (Imaginary)

> Tested with debian 12

```
# Imaginary
apt update -y
apt install libvips libvips-dev libopenslide-dev golang pkg-config original-awk grep bc -y

go install github.com/h2non/imaginary@latest
mv go/bin/imaginary /usr/local/bin/imaginary
chmod +x /usr/local/bin/imaginary
rm -rf go/

cat << EOF > /etc/systemd/system/imaginary.service
[Unit]
Description=Imaginary Service
After=network.target

[Service]
Type=simple
Environment="MALLOC_ARENA_MAX=2"
ExecStart=/usr/local/bin/imaginary -p 9001 -concurrency 20
Restart=on-failure
RestartSec=10s

[Install]
WantedBy=multi-user.target
EOF
systemctl daemon-reload
systemctl enable --now imaginary

# use Imaginary for Preview Generation
HALF_MEMORY_GB=$(echo "$(grep MemTotal /proc/meminfo | awk '{print int($2 / 1024 / 1024 / 2)}')")
INSTANCE_ID=$(grep "'instanceid'" /var/www/nextcloud/config/config.php | awk -F "=> '" '{print $2}' | sed
"s/',//")
```

```
NC_DATA=$(grep "'datadirectory'" /var/www/nextcloud/config/config.php | awk -F "=> '" '{print $2}' | sed "s/',//")
PREVIEW_DIR="$NC_DATA/appdata_$INSTANCE_ID/preview"
OCC="sudo -u www-data php /var/www/nextcloud/occ"
$OCC config:system:set enable_previews --value=true
$OCC config:system:set enabledPreviewProviders 0 --value="OC\\Preview\\Imaginary"
$OCC config:system:set preview_imaginary_url --value="http://127.0.0.1:9001"

systemctl stop nginx php8.2-fpm redis-server postgresql imaginary

# use RAM instead of storage for previews
rm -rf $PREVIEW_DIR
echo "nextcloud-previews $PREVIEW_DIR tmpfs x-mount.mkdir,uid=33,gid=33,size=${HALF_MEMORY_GB}G 0 0" >> /etc/fstab
systemctl daemon-reload
mount -a

systemctl start nginx php8.2-fpm redis-server postgresql imaginary
$OCC files:scan-app-data

# Cronjobs
cat << EOF > /etc/cron.d/nextcloud-previews
PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin
LD_LIBRARY_PATH=/usr/local/lib
@reboot www-data sleep 10; php -f /var/www/nextcloud/occ files:scan-app-data
EOF
systemctl enable --now cron
```

AIO-Installation

LVM

LVM

Thin-Pool

Tasks

Anlegen

```
NAME="local-NVME"
BLOCK="/dev/vdb"
wipefs -a $BLOCK
pvcreate $BLOCK
vgcreate $NAME $BLOCK
lvcreate --type thin-pool --chunksize 64K -l 100%FREE -n vm-data $NAME
```

Troubleshooting

LVM Pool aktivierung dauert ewig

set `thin_check_options = [ "-q", "--skip-mappings" ]` in file `/etc/lvm/lvm.conf`

afterwards run: `update-initramfs -u -k all`

Proxmox

Proxmox

Remote-Migrate

```
pct remote-migrate 100 100 apitoken='PVEAPIToken=root@pam!migrate=8afa7395-1499-4ce3-90c0-XXXXX',host=192.168.178.XXX,fingerprint=XXX --target-storage local-HDD --target-bridge vmbr0 --restart
```

Mailcow

mailcow.conf Tuneables

```
# Direkter Login Button für jeden Nutzer als Admin um sich bei SOGo direkt als diesen anzumelden
sed -i 's/^ALLOW_ADMIN_EMAIL_LOGIN=.*/ALLOW_ADMIN_EMAIL_LOGIN=y/' mailcow.conf
```

Änderungen anwenden:

```
docker compose up -d
```

reset rspamd config/stats/full

```
docker compose down
docker volume rm -f mailcowdockerized_rspamd-vol-1
docker compose up -d
docker compose exec redis-mailcow sh -c 'redis-cli --scan --pattern BAYES_* | xargs redis-cli del'
docker compose exec redis-mailcow sh -c 'redis-cli --scan --pattern RS* | xargs redis-cli del'
docker compose exec redis-mailcow sh -c 'redis-cli --scan --pattern rn_* | xargs redis-cli del'
## Clear Fuzzy data
# Wir müssen zuerst die redis-cli aufrufen:
docker compose exec redis-mailcow redis-cli
# In redis-cli geben wir nun ein:
127.0.0.1:6379> EVAL "for i, name in ipairs(redis.call('KEYS', ARGV[1])) do redis.call('DEL', name); end" 0 fuzzy*
```

Verbessertes/Eigenes Spamverhalten

Normalerweise wendet Mailcow ein entweder oder + oder prinzip bei jeder E-Mail an die reinkommt.

Also die E-Mail landet entweder in der Inbox ODER im Spam ODER wird abgelehnt. Dieses verhalten halte ich persönlich für zu umständlich und doof.

Im folgenden wird mailcow also so umgestellt, dass es entweder zustellt ODER Ablehnt/Quarantänisiert aber die "Move to Spamfolder" Funktionalität weiterhin erhalten bleibt damit das lernen von SPAM einfacher ist.

Sieve Filter anpassen

Der Filter der E-Mails mit dem header "X-Spam-Flag" in den Junk Ordner verschiebt wird hiermit gelöscht.

```
sed -i '/if header :contains "X-Spam-Flag" "YES" {/,/}/d' data/conf/dovecot/global_sieve_after
```

Rspamd verhalten anpassen

add_header aktion deaktivieren

notwendig damit der Spam header nicht mehr gesetzt wird für emails welche spam sein könnten.

```
grep -q '^add_header' data/conf/rspamd/local.d/actions.conf && sed -i 's/^add_header.*/add_header = null;/' data/conf/rspamd/local.d/actions.conf || echo 'add_header = null;' >> data/conf/rspamd/local.d/actions.conf
```

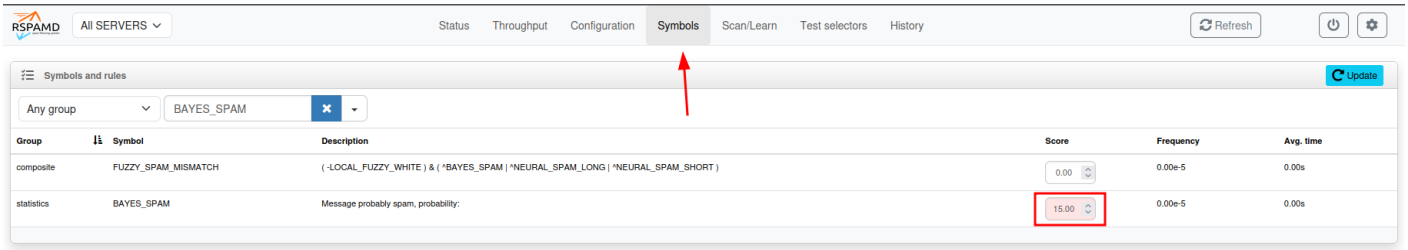
Referenz: https://rspamd.com/doc/faq.html#how-can-i-disable-some-rspamd-action

Bayes SPAM aggressiver einstellen

Wird dies nicht gemacht ist es sehr warscheinlich dass E-Mails welche schonmal als SPAM Gelernt wurden trotzdem zugestellt werden. Das passiert weil der Score zu gering durch "BAYES_SPAM"

beeinträchtigt wird.

Dafür in der Rspamd UI den "BAYES_SPAM" parameter anpassen:



Den wert auf z.B. "15" setzen.

Dienste Neustarten

```
docker compose restart dovecot-mailcow
docker compose restart rspamd-mailcow
```

Prometheus

Node Exporter

Installation

```
apt install prometheus-node-exporter
```

Basis Config

Implementierung von Self-Signed SSL with Snakeoil cert and basic user auth.

```
mkdir -p /etc/prometheus-node-exporter
password=`openssl rand -base64 32`
passwordHashed=`echo ${password} | htpasswd -inBC 10 "" | tr -d ':\n'`
echo "Clear password to keep for Prometheus Server: ${password}"

adduser prometheus ssl-cert
cat << "EOF" > /etc/default/prometheus-node-exporter
# Set the command-line arguments to pass to the server.
# Due to shell escaping, to pass backslashes for regexes, you need to double
# them (\\d for \d). If running under systemd, you need to double them again
# (\\\\d to mean \d), and escape newlines too.
ARGS="--web.config.file="/etc/prometheus-node-exporter/configuration.yml""
EOF

cat << EOF > /etc/prometheus-node-exporter/configuration.yml
tls_server_config:
  cert_file: /etc/ssl/certs/ssl-cert-snakeoil.pem
  key_file: /etc/ssl/private/ssl-cert-snakeoil.key
basic_auth_users:
  prometheus: ${passwordHashed}

EOF
```

restart service: `systemctl restart prometheus-node-exporter`

Nextcloud AIO

Installation

> The whole Document is tested with a fresh Debian 12 Installation only!

Basis

> Current State is from: 2025-01-16

```
apt update -y
apt upgrade -y
```

Webserver

```
apt install nginx python3-certbot-nginx -y
```

PHP

As per survy README

```
apt-get update
apt-get install lsb-release ca-certificates curl -y
curl -sSLo /tmp/debsuryorg-archive-keyring.deb https://packages.sury.org/debsuryorg-archive-keyring.deb
dpkg -i /tmp/debsuryorg-archive-keyring.deb
sh -c 'echo "deb [signed-by=/usr/share/keyrings/deb.sury.org-php.gpg] https://packages.sury.org/php/
$(lsb_release -sc) main" > /etc/apt/sources.list.d/php.list'
apt-get update
```

Required and DB Connector as well as Recommeded as per Nextcloud Documentation.

```
apt install php8.3-{ctype,curl,dom,gd,mbstring,posix,simplexml,xmlreader,xmlwriter,zip,pgsql,intl,fpm} -y
```

Konfiguration

```
_php_config_files=(
  "/etc/php/8.3/fpm/php.ini"
  "/etc/php/8.3/cli/php.ini"
)

for file in ${_php_config_files[@]}; do
sed -i 's/^max_execution_time =.*/max_execution_time = 3600/' $file
sed -i 's/^memory_limit =.*/memory_limit = 512M/' $file
sed -i 's/^.*opcache.enable.*=.*/opcache.enable=1/' $file
```

PSQL

VyOS

Build own iSO

```
apt-get update && apt-get install -y \
dialog apt-utils locales bash bash-completion \
vim vim-autopep8 nano git curl sudo mc pbuilder \
devscripts equivs lsb-release libtool libapt-pkg-dev \
flake8 pkg-config debhelper gosu po4a openssh-client jq \
socat build-essential python3-pystache squashfs-tools \
genisoimage fakechroot pipx python3-git python3-pip \
python3-flake8 python3-autopep8 python3-tomli \
python3-tomli-w yq debootstrap live-build gdisk sbsigntool \
dosfstools swtpm debhelper libffi-dev libpcre3-dev unzip qemu-utils kpartx
```