

LAP stack installation

Packages

```
dnf install -y epel-release
dnf install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
dnf install -y https://rpms.remirepo.net/enterprise/remi-release-8.rpm
dnf config-manager --set-enabled remi
dnf module install -y php:remi-7.4
dnf install -y \
    httpd \
    php \
    php-bcmath \
    php-cli \
    php-gd \
    php-gmp \
    php-intl \
    php-json \
    php-mbstring \
    php-mcrypt \
    php-opcache \
    php-pdo \
    php-pear \
    php-pgsql \
    php-xml \
    php-xmlrpc \
    php-zstd
```

httpd config

```
systemctl stop httpd

# welcome page
echo "# welcome disabled" > /etc/httpd/conf.d/welcome.conf

# error_log > error.log
rm /var/log/httpd/error_log -f
sed -i 's|^ErrorLog "logs/error_log"|ErrorLog "/var/log/httpd/error.log"|' /etc/httpd/conf/

cat << HARDEN_END > /etc/httpd/conf.d/harden.conf
# harden web server ... see https://devops.ionos.com/tutorials/how-to-harden-the-apache-web

# Hide The Apache Version
ServerSignature Off
ServerTokens Prod

# Outside The Document Root
<Directory />
    Options -Indexes
    AllowOverride None
</Directory>

# Secure Apache From Clickjacking Attacks
Header append X-FRAME-OPTIONS "SAMEORIGIN"

# Disable ETag
FileETag None

# Secure Apache From XSS Attacks
<IfModule mod_headers.c>
    Header set X-XSS-Protection "1; mode=block"
</IfModule>

# Protect Cookies With HTTPOnly Flag
<IfModule mod_headers.c>
    Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
</IfModule>
HARDEN_END
```

Strip down modules loaded: disable all, then enable those needed

```

# disabled all
sed -i 's|^LoadModule |#LoadModule |' /etc/httpd/conf.modules.d/*

for m in $( echo "
    access_compat_module
    alias_module
    auth_basic_module
    authn_file_module
    authz_core_module
    authz_host_module
    authz_user_module
    autoindex_module
    core_module
    deflate_module
    dir_module
    expires_module
    headers_module
    http2_module
    http_module
    info_module
    log_config_module
    logio_module
    mime_module
    rewrite_module
    setenvif_module
    so_module
    status_module
    systemd_module
    unixd_module
    php7_module
    mpm_event_module
    proxy_fcgi_module
    proxy_module
" | egrep -w "^[[:blank:]]*[a-z_]*_module" | sort -u ) ; do
    if [ ! -z "$m" ]; then
        echo "* enabling $m"
        sed -i "s|#LoadModule $m |LoadModule $m |" /etc/httpd/conf.modules.d/*
    fi
done

```

Example VHOST (hier rpac)

```
install -o root -g apache -m 0750 -d /srv/www/rpac
echo "<?php phpinfo(); ?>" > /srv/www/rpac/index.php
rm -f /etc/httpd/conf.d/tokenizer.conf
cat << VHOST_END > /etc/httpd/conf.d/rpac.conf
<VirtualHost *:80>
    ServerName $(hostname -f)
    ServerAlias rpac.app.xres.rz1.xtrav.de
    DocumentRoot /srv/www/rpac
    ErrorLog /var/log/httpd/error.log
    CustomLog /var/log/httpd/rpac.log combined

    <Directory /srv/www/rpac>
        Options -Indexes -FollowSymLinks -ExecCGI -Includes
        AllowOverride All
        Require all granted
    </Directory>

</VirtualHost>
VHOST_END
```

verify vhost config

```
# check startup errors and syntax config
httpd -M
httpd -t
```

start webserver

```
systemctl start httpd
```