Wordpress on Debian

This page describes the installation of Wordpress under Debian 7 and 8 (Wheezy and Jessie), in an lxc container.

*This page is mostly outdated: for Stretch (Debian 9), go to Wordpress on Debian 9 (Stretch)*

Import a site from another machine

The shell script below assumes many things:

- names of the sites, directories on the old machine match the ones to be used on the new machine
- the ssh keys are properly forwarded, and you have your public key in the /root/.ssh/authorized_keys file of the old machine

So: YMMV (do not forget to use your brain).

```
SITE_NAME=my.site.org  ## Put the complete name (should match a valid, public DNS entry)
DB_NAME=$(hostname)  ## Please correct if the DB name does not match the host name
OLD_MACHINE=176.9.96.186  ## The name or IP of the machine to import the existing site from (default: the
    public IP address of CT108, aka WebPanel/ISPManager)
OLD_WP_CONTENT_DIR=/var/www/$DB_NAME/data/www/$SITE_NAME/wp-content  ## The directory of wp-content on the
    old machine (default: webpanel standard location)
NEW_WP_CONTENT_DIR=/var/lib/wordpress/wp-content  ## The directory of wp-content on the new installation, do
    not change unless you know what you are doing
apt-get install wordpress nginx mysql-server php5-fpm pwgen
DB_USER_PASSWORD=$(pwgen 8 1)  ## Generate a random password
mysql << EOF
CREATE DATABASE $DB_NAME;
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
ON $DB_NAME.*
TO $DB_NAME@localhost
IDENTIFIED BY '$DB_USER_PASSWORD';
FLUSH PRIVILEGES;
EOF

cat > /etc/wordpress/config-$SITE_NAME.php << EOF
<?php
    define('DB_NAME', '$DB_NAME');
    define('DB_USER', '$DB_NAME');
    define('DB_PASSWORD', '$DB_USER_PASSWORD');
    define('DB_HOST', 'localhost');
    define('WP_CONTENT_DIR', '/var/lib/wordpress/wp-content');
/** IN URI INDEXOF THE CATEGORY*/
    define('NUMBER_SLASH', '2');
    define('NUMBER_SLASH2', '3');
    define('CAT26', 'courses');
    define( 'DB_CHARSET', 'utf8' );
    define('FS_METHOD', 'direct');
?>
EOF

cat > /etc/nginx/sites-available/$DB_NAME << EOF
    server {
        listen 80;
        root /usr/share/wordpress;
        index index.php index.html index.htm;
        server_name $SITE_NAME;
        access_log /var/log/nginx/$DB_NAME.access.log;
        error_log /var/log/nginx/$DB_NAME.error.log;

        location = /favicon.ico {
            log_not_found off;
            access_log off;
```

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

location /wp-content/ {
    root /var/lib/wordpress;
    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires 30d;
        log_not_found off;
    }
}

location / {
    try_files $uri $uri/ /index.php?$args;
}

location ~ .php$ {
    #NOTE: You should have "cgi.fix_pathinfo = 0;" in php.ini
    include /etc/nginx/fastcgi.conf;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
    #fastcgi_intercept_errors on;
}
}

EOF

## Relax the permissions on wp-content, allowing the admin panel to install/update themes and plugins
chown www-data -R /var/lib/wordpress/wp-content

## Add directory for updates
mkdir -p /usr/share/wordpress/wp-content/wppa-depot
chown www-data -R /usr/share/wordpress/wp-content/wppa-depot
ln -s ../sites-available/${DB_NAME} /etc/nginx/sites-enabled/  ## Activates the nginx config
rm /etc/nginx/sites-enabled/default  ## Remove the default nginx setting: not used
nginx -t  ## Test the nginx config
service nginx restart

## Copy and import an existing sql database
## It must have been generated from the "old" machine with:
## OLD_DB_NAME=foobar
## mysql -u root -p ${OLD_DB_NAME} < /tmp/${DB_NAME}.sql
## Copy the files from the "old" machine
rsync -r root@${OLD_MACHINE}:${OLD_WP_CONTENT_DIR}/uploads/* ${NEW_WP_CONTENT_DIR}/uploads/
rsync --exclude=index.php --exclude=akismet -r root@${OLD_MACHINE}:${OLD_WP_CONTENT_DIR}/plugins/* ${NEW_WP_CONTENT_DIR}/plugins
rsync --exclude=index.php --exclude=twentythirteen --exclude=twentytwelve -r root@${OLD_MACHINE}:${OLD_WP_CONTENT_DIR}/themes/* ${NEW_WP_CONTENT_DIR}/themes

Utilities

Fix encoding

Common problem importing from a Mysql dump generated with unknown method. We can somehow fix these directly in the database.
CREATE admin user

Connect to the DB with mysql, then:
Change domain name

Scenario: the site at old.name.org is to be changed to new.name.org.

**Step 1: Wordpress config**

```bash
mv /etc/wordpress/conf-old.name.org.php /etc/wordpress/conf-new.name.org.php
```

**Step 2: Wordpress DB**

The connection parameters can be found in {{/etc/wordpress/conf-new.name.org.php}}:

```bash
mysql -u user_name -h localhost --password=secret db_name << EOF
UPDATE wp_options SET option_value = 'http://new.name.org' WHERE option_name = 'home' OR option_name = 'siteurl';
EOF
```

**Step 3: nginx config**

Nginx can instruct browsers to use the new domain name, add this snippet in /etc/nginx/sites-available/wordpress_site_config_file

```bash
server {
    server_name old.name.org;
    rewrite ^ $scheme://new.name.org$request_uri permanent;
}
```

And, change the {{server_name}} attribute in the main server block to: new.name.org.

**Step 4: Nginx frontend (web2)**

Add new.name.org (and eventually {{*.new.name.org}}) to the server_name attribute in the relevant file in web2:/etc/nginx/sites-enabled/.