Server Software

Server software configuration and installation procedures such as Apache, and Postfix

- Apache Option FollowSymLinks not allowed here Error
- Migrating Self-Signed SSL Certificate to LetsEncrypt Certificate
- LAMP Stack Upgrade Issues
- Standard Installation Procedures for LAMP Stack on CentOS 7
- Slow Loading on Ownloud 8
- Postfix and Dovecot Configuration
- Install RethinkDB on CentOS 7
- Turtl API Server and Client Installation CentOS 7

Apache Option FollowSymLinks not allowed here Error

Apache htaccess `Option FollowSymLinks not allowed here` error:

```
find /home -name ".htaccess" -type f -exec sed -i '/FollowSymLinks/SymLinksIfOwnerMatch/g' {} ";"
```

Migrating Self-Signed SSL Certificate to LetsEncrypt Certificate

Download Let's Encrypt Client

1. `sudo -s`
2. `git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt`

Update Apache Configuration

Let's Encrypt does not detect multiple virtual host in a single file, so if you have multiple virtual hosts in a single file, you need to separate it and update the configuration for SSL only. Then redirect all plain-text traffic to SSL using a single virtual host.

Create a new virtual host in `/etc/httpd/conf.d/redirect_ssl.conf` to redirect plain-text traffic to SSL, replace all `<domain>` to your TLD, such as `example.com`:

1. `<VirtualHost *:80>`
2. `ServerName <domain>`
3. `ServerAlias *.<domain>`
4. `RewriteEngine on`
5. `RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R=301,L]`
6. `</VirtualHost>`

Setup SSL Certificates

1. `cd /opt/letsencrypt`
2. `./letsencrypt-auto --apache -d <domain> -d www.<domain> -d <subdomain>.<domain>`

Replacing `<domain>` with your domain, subsequent subdomains can be specified with `-d` option.

Restart Apache and Test

1. `systemctl restart httpd`

(Optional) Renewing SSL Certificates

Let's Encrypt issue **90 days** validity certificates, but you can however, renew it earlier in case errors occurred.

To renew the certificates, simply use the following command:

1. `/opt/letsencrypt/letsencrypt-auto renew`

If you have just created a new certificate, Let's Encrypt will never issue you a new one, it will only issue a new certificate for your domains if the validity period is **less than 30 days**, so, you can create a cronjob to try and renew the certificate every day, week or month, in case anything goes wrong with your certificate.

To setup cronjob to automatically renew certificate, enter command `crontab -e` to create a new cronjob and add the following line:

1. `0 3 * * 1 /opt/letsencrypt/letsencrypt-auto renew >> /var/log/le-renew.log`

The cronjob above will run on **every monday** at **3 A.M.**, it will append any output from `/opt/letsencrypt/letsencrypt-auto` to `/var/log/le-renew.log`. Please refer to the reference for more info on Linux cronjobs.

References

1. Digital Ocean Article
2. Let's Encrypt Article
3. Cronjob Format

LAMP Stack Upgrade Issues

"Table Doesn't Exists" After MySQL/MariaDB Upgrade

Paste MySQL data directory to upgraded data directory, containing `ibdata1`, `ib_logfile0` and `ib_logfile1`, in `lampp`, it's `/opt/lampp/var/mysql`:

1. `sudo cp /opt/lampp_backup/var/mysql /opt/lampp/var/mysql`
2. `sudo chown -R mysql: mysql /opt/lampp/mysql`

Standard Installation Procedures for LAMP Stack on CentOS 7

1. System Upgrade

1. `yum -y update`

2. Install Required Software

1. `yum -y install git policycoreutils-python httpd mariadb mariadb-server php-mysql php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap curl curl-devel`

3. Setup MySQL Server

1. `mysql_secure_installation`

4. Start and Enable All Services

1. `systemctl enable httpd`
2. `systemctl enable mariadb`
3. `systemctl start httpd`
4. `systemctl start mariadb`

5. Open Firewall Ports

1. `firewall-cmd --permanent --zone=public --add-service=http`
2. `firewall-cmd --permanent --zone=public --add-service=https`
3. `firewall-cmd --permanent --zone=public --add-port=<ssh_port>/tcp`
4. `firewall-cmd --reload`

6. Change SSH Port

1. `vim /etc/ssh/sshd_config #and append 'Port <ssh_port>'`
2. `semanage port -a -t ssh_port_t -p tcp <ssh_port>`
3. `systemctl restart sshd`

7. Enable Shutdown Button

Edit `/etc/systemd/logind.conf` and uncomment the following 2 lines:

1. `PowerKeyIgnoreInhibited=no`
2. `HandlePowerKey=poweroff`

8. Reboot System

1. `reboot`

(HP MicroServer Only)

Edit `/etc/default/grub` and append `clocksource=hpet nolapic` to the end of `GRUB_CMDLINE_LINUX` variable.

Slow Loading on Ownloud 8

Change `/var/www/html/owncloud/config/config.php` database host to `127.0.0.1` instead of `localhost`

Postfix and Dovecot Configuration

Installation

1. `hostnamectl set-hostname mail.<domain>.<tld>`
2. `yum -y install postfix dovecot`

Postfix Configuration

1. Append the following to `/etc/postfix/main.cf`:
    1. `myhostname = mail.<domain>.<tld>`
    2. `mydomain = <domain>.<tld>`
    3. `myorigin = $mydomain`
    4. `home_mailbox = mail/`
    5. `mynetworks = 127.0.0.0/8 <domain IP>`
    6. `inet_interfaces = all`
    7. `mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain`
    8. `smtpd_sasl_type = dovecot`
    9. `smtpd_sasl_path = private/auth`
    10. `smtpd_sasl_local_domain =`
    11. `smtpd_sasl_security_options = noanonymous`
    12. `broken_sasl_auth_clients = yes`
    13. `smtpd_sasl_auth_enable = yes`
    14. `smtpd_recipient_restrictions = permit_sasl_authenticated,permit_mynetworks, reject_unauth_destination`
    15. `smtp_tls_security_level = may`
    16. `smtpd_tls_security_level = may`
    17. `smtp_tls_note_starttls_offer = yes`
    18. `smtpd_tls_loglevel = 1`
    19. `smtpd_tls_key_file = /etc/letsencrypt/live/<domain>.<tld>/privkey.pem`
    20. `smtpd_tls_cert_file = /etc/letsencrypt/live/<domain>.<tld>/fullchain.pem`
    21. `smtpd_tls_received_header = yes`
    22. `smtpd_tls_session_cache_timeout = 3600s`
    23. `smtpd_use_tls=yes`

24. `tls_random_source = dev:/dev/urandom`
25. `virtual_alias_domains = <domain>.<tld>`
26. `virtual_alias_maps = hash:/etc/postfix/virtual`

2. Find and uncomment the following lines in `/etc/postfix/main.cf`:
    1. `#inet_interfaces = localhost`
    2. `#mydestination = $myhostname, localhost.$mydomain, localhost`
3. Append the following lines to `/etc/postfix/master.cf`:
    1. `submission inet n - n - - smtpd`
    2. `-o syslog_name=postfix/submission`
    3. `-o smtpd_sasl_auth_enable=yes`
    4. `-o smtpd_recipient_restrictions=permit_sasl_authenticated,reject_unauth_destination`
    5. `-o milter_macro_daemon_name=ORIGINATING`
    6. `smtps inet n - n - - smtpd`
    7. `-o syslog_name=postfix/smtps`
    8. `-o smtpd_sasl_auth_enable=yes`
    9. `-o smtpd_recipient_restrictions=permit_sasl_authenticated,reject_unauth_destination`
    10. `-o milter_macro_daemon_name=ORIGINATING`
4. Make sure that the following is present in `/etc/postfix/main.cf`:
    1. `alias_maps = hash:/etc/aliases`
5. Edit and add the desired email address to `/etc/postfix/virtual` such as the following:
    1. `info@<domain>.<tld> admin`
    2. `webmaster@<domain>.<tld> admin`
6. Create a map database: `postmap /etc/postfix/virtual`

Dovecot Configuration

1. Find and modify the following lines in `/etc/dovecot/conf.d/10-master.conf`:
    1. `# Postfix smtp-auth`
    2. `unix_listener /var/spool/postfix/private/auth {`
    3. `mode = 0660`
    4. `user = postfix`
    5. `group = postfix`
    6. `}`
2. Find and modify the following lines in `/etc/dovecot/conf.d/10-auth.conf`:
    1. `auth_mechanisms = plain login`
3. Find and modify the following lines in `/etc/dovecot/conf.d/10-mail.conf`:
    1. `mail_location = maildir:~/mail`
4. Find and modify the following lines in `/etc/dovecot/conf.d/20-pop3.conf`:
    1. `pop3_uidl_format = %08Xu%08Xv`
5. Find and modify the following lines in `/etc/dovecot/conf.d/10-ssl.conf`:
    1. `ssl_cert = </etc/letsencrypt/live/<domain>.<tld>/fullchain.pem`
    2. `ssl_key = </etc/letsencrypt/live/<domain>.<tld>/privkey.pem`

Restart and Enable Services

1. `systemctl restart postfix`
2. `systemctl enable postfix`
3. `systemctl restart dovecot`
4. `systemctl enable dovecot`

Open Firewall Ports

1. `firewall-cmd --permanent --add-service=smtp`
2. `firewall-cmd --permanent --add-port=587/tcp`
3. `firewall-cmd --permanent --add-port=465/tcp`
4. `firewall-cmd --permanent --add-port=110/tcp`
5. `firewall-cmd --permanent --add-service=pop3s`
6. `firewall-cmd --permanent --add-port=143/tcp`
7. `firewall-cmd --permanent --add-service=imaps`
8. `firewall-cmd --reload`

Configure DNS

1. Add an `A` record for the mail server:
    1. `name = mail.<domain>.<tld>`
    2. `IP = <mail server IP>`
2. Add an `MX` record:
    1. `Hostname = mail.<domain>.<tld>`
    2. `Priority = 5`
3. Add the following `TXT` records:
    1. `Name = @`
    2. `Text = "v=spf1 ip4:<domain IP> ~all"`
    3. `Name = _dmarc.<domain>.<tld>`
    4. `Text = v=DMARC1; p=none`
4. Add `PTR` record for `<domain>.<tld>`
5. Finally, test your email at `https://www.mail-tester.com/`

Notes on Using Let's Encrypt for SSL

Make sure that `Encryption` is set to `STARTTLS` when configuring mail clients

References

1. Krizna Article
2. Ubuntu Postfix Alias Configuration

Install RethinkDB on CentOS 7

Installing RethinkDB

```
sudo wget http://download.rethinkdb.com/centos/7/`uname -m`/rethinkdb.repo -O
/etc/yum.repos.d/rethinkdb.repo
sudo yum install rethinkdb
```

Create Service File

Create the service file, `/usr/lib/systemd/system/rethinkdb@.service` with the following content:

```
[Unit]
Description=RethinkDB database server for instance '%i'

[Service]
User=rethinkdb
Group=rethinkdb
ExecStart=/usr/bin/rethinkdb serve --config-file /etc/rethinkdb/instances.d/%i.conf
KillMode=process
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

Make sure that it has a permission of `644`: `chmod 644 /usr/lib/systemd/system/rethinkdb@.service`

Creating a Rethink DB Instance

1. Create the RethinkDB data directory: `rethinkdb create -d /path/to/your/rethinkdb/directory`
2. Set the ownership to RethinkDB user: `sudo chown -R rethinkdb. rethinkdb /path/to/your/rethinkdb/directory`
3. Copy RethinkDB sample config file: `sudo cp /etc/rethinkdb/default. conf. sample /etc/rethinkdb/instances. d/instance1. conf`
4. Edit `/etc/rethinkdb/instances. d/instance1. conf`, the line with `directory=` must be changed to point to your Rethink DB data directory.

Start RethinkDB Instance

in this case would be `instance1`:

```
sudo systemctl enable rethinkdb@<name_instance>
sudo systemctl start rethinkdb@<name_instance>
```

References

RethinkDB Startup Doc

Turtl API Server and Client Installation CentOS 7

Turtl API

Clone and Configure Turtl API

1. Create a user for turtl API: `sudo useradd turtl`
2. Switch user to `turtl`: `sudo su turtl`
3. Change directory to `turtl`'s home: `cd ~`
4. Install Clozure CL
5. Install RethinkDB and create an instance for Turtl API
6. Install `libuv`: `sudo yum install libuv`
7. Clone Turtl repo: `git clone https://github.com/turtl/api.git`
8. Copy Turtl API config: `cp config/config.default.lisp config/lisp`
9. Edit and configure `config/config.lisp`, make sure to update the following parameters:

```
(defvar *local-upload* "<local upload directory>"
(defvar *local-upload-url* "<upload url>"
```

Setup Up Service

Create a service file at `/usr/lib/systemd/system/turtl.service` with the following entry:

```
[Unit]
Description=Turtl API Server

[Service]
User=turtl
Group=turtl
ExecStart=/usr/local/bin/ccl64 --load /home/turtl/api/start.lisp
KillMode=process
```

```
[Install]
WantedBy=multi-user.target
```

Start and enable the service:

```
sudo systemctl start turtl
sudo systemctl enable turtl
```

(OPTIONAL) Configure Reverse Proxy in Apache

Create `httpd` virtual host configuration `/etc/httpd/conf.d/turtl.conf` with the following content, make sure to change `<turtl domain>` to your own domain name:

```
<VirtualHost *:80>
    #Server name configuration
    ServerName <turtl domain>
    ServerAdmin webmaster@<turtl domain>

    #Proxy configuration
    ProxyPreserveHost on
    ProxyRequests off
    ProxyPass / https://wiki.twcloud.tech:8181/
    ProxyPassReverse / https://wiki.twcloud.tech:8181/

    #Logging configuration
    ErrorLog /var/log/httpd/turtl.err
    LogLevel warn
</VirtualHost>
```

(OPTIONAL) Restrict User Registration

Add the following lines in your Turtl API Virtual Hosts configuration:

```
    #Restrict Registration
    <LocationMatch  "^/users[/]?$">
        AuthType Basic
```

```
        AuthName "Restricted"
        AuthUserFile /home/turtl/.htpasswd
        Require valid-user
    </LocationMatch>
```

Then generate a `.htpasswd` password file in `/home/turtl`: `sudo htpasswd -c /etc/apache2/.htpasswd <whatever username>`. Make sure that it's in the right permission and owner: `chmod 640 /home/turtl/.htpasswd && chown turtl:apache /home/turtl/.htpasswd`

Installing JS Client

1. Clone `turtl/js` repo to webserver webroot: `sudo mkdir /var/www/turtl && cd /var/www/turtl && sudo git clone https://github.com/turtl/js.git .`
2. Install NodeJS dependencies: `npm install`
3. Copy default config: `cp config/config.js.default config.js`
4. Edit `config/config.js`
5. Update owner and group: `chown -R apache:apache .`
6. Generate assets: `make`