Configure local https on ubuntu

T0 set up a local HTTPS server on Ubuntu, use a tool like `mkcert` for simplicity or `openssl` for more control, and then configure your web server (e.g., Apache) to use the generated certificates. The recommended method is to install and use `mkcert`, which generates a trusted, local Certificate Authority (CA) to create self-signed certificates for `localhost` or custom hostnames without browser warnings. Alternatively, you can generate a self-signed certificate using `openssl`, which requires more manual steps to create the key, certificate, and to trust it in your browser.

Method 1: Using `mkcert` (Recommended)

This video demonstrates how to set up HTTPS with mkcert and enable HTTPS for localhost:

Unconventional Coding

YouTube • 31 Dec 2022

1. **Install `mkcert`**: Install the tool to manage local certificate authorities and certificates.
   - `sudo apt install mkcert`
2. **Install the local CA**: Run `mkcert` to install a local CA that your system will trust.
   - `mkcert -install`
3. **Create a certificate**: Generate a certificate and key for your local site (e.g., `localhost`) or domain. Point to note here in case of domain, ip address should be mapped to it before this command.
   - `mkcert localhost 127.0.0.1`
   - `sudo mkcert drupalwithdata.offline.com`

Configure your server: Configure your specific application or web server to use the generated `localhost.pem` (certificate) and `localhost-key.pem` (key) files.

Method 2: Using openssl

1. **Install `openssl`**: Ensure `openssl` is installed. It is usually pre-installed on Ubuntu.
2. **Generate a private key and a certificate signing request (CSR)**:
   - `openssl req -new -newkey rsa:2048 -nodes -keyout localhost.key -out localhost.csr`

3. **Generate the self-signed certificate**: Use the CSR and private key to create the certificate.
   - `openssl x509 -req -days 365 -in localhost.csr -signkey localhost.key -out localhost.crt`
4. **Configure your web server**:
   - **For Apache**:
     - Enable the SSL module: `sudo a2enmod ssl`.
     - Configure your site's virtual host to point to your certificate (`localhost.crt`) and private key (`localhost.key`) files.
     - Restart Apache: `sudo systemctl restart apache2`.
   - **For other servers**: Configure your server's settings to use the `.crt` and `.key` files.
5. **Trust the certificate**:
   - This is an optional but recommended step to avoid browser warnings.
   - Install certificate utilities: `sudo apt-get install libnss3-tools`.
   - Import and trust the certificate into the browser's database (using `pk12util` for Firefox or `certutil` for other applications).

Final steps for both methods

- After generating the certificates and configuring your server, you may still see a browser warning because the certificate is "self-signed" and not issued by a public Certificate Authority.
- You will need to accept the security risk to proceed to your local serve

---

Revision #4
Created 29 October 2025 02:43:33 by AI API
Updated 14 December 2025 14:17:18 by AI Channel