Running the Secure Search Server as a Linux Daemon

1 Overview

This document provides step-by-step instructions to run the Secure Search Server as a Linux daemon using `systemd`. It covers creating a service file, reloading `systemd`, starting/enabling the service, and monitoring/managing the service.

2 Generating SSL Certificates and Keys

For secure communication, you need an SSL certificate and a private key. Use OpenSSL to generate these files.

2.1 Generating a Self-Signed Certificate

Run the following command to generate a self-signed SSL certificate and private key:

```
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes
```

This command:

- Generates a 4096-bit RSA private key and saves it as `key.pem`.
- Creates a self-signed certificate valid for 365 days, saved as `cert.pem`.
- Uses the `-nodes` option to prevent encryption of the private key.

2.2 Verifying the Certificate

After generating the certificate, verify it using:

```
openssl x509 -in cert.pem -text -noout
```

2.3 Using the Certificate in the Secure Search Server

Place `key.pem` and `cert.pem` in a secure directory and update the server configuration to use them for SSL.

3 Creating the Systemd Service File

Create a new service file (e.g., `/etc/systemd/system/secure-search.service`) with the following content. Adjust the paths and user according to your environment.

```
[Unit]
Description=Secure Search Service
After=network.target

[Service]
Type=simple
User=your_username
WorkingDirectory=/path/to/your/server/directory
ExecStart=/usr/bin/python3 /path/to/your/server.py
Restart=on-failure
Environment="PYTHONUNBUFFERED=1"

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

Notes:

- Replace `your_username` with the Linux user under which the server should run.
- Replace `/path/to/your/server/directory` with the directory where your server files reside.
- Replace `/path/to/your/server.py` with the correct path to your `server.py` file.

4 Reloading Systemd

After creating or modifying the service file, reload the `systemd` manager configuration:

```
sudo systemctl daemon-reload
```

5 Starting and Enabling the Service

5.1 Start the Service

To start the service immediately, execute:

```
sudo systemctl start secure-search
```

5.2 Enable the Service at Boot

To ensure the service starts automatically at boot time, run:

```
sudo systemctl enable secure-search
```

6 Monitoring and Managing the Service

6.1 Checking Service Status

To check the status of the service:

```
sudo systemctl status secure-search
```

6.2 Viewing Logs

To view logs for the service, use `journalctl`:

```
sudo journalctl -u secure-search
```

6.3 Stopping or Restarting the Service

To stop the service:

```
sudo systemctl stop secure-search
```

To restart the service:

```
sudo systemctl restart secure-search
```

7 Conclusion

By following these instructions, you can run your Secure Search Server as a background daemon on a Linux system using `systemd`. For further customization (e.g., adding additional environment variables or resource limits), modify the service file as needed.