Installation

The following outlines the instructions for installing the analysis server on Ubuntu linux.

Database

- Install a new server with Ubuntu or use an existing installation
- Install the PostgreSQL database server

```
sudo apt-get install postgresql postgresql-contrib
```

- Configure the PostgreSQL server e.g.. set a new **postgres** user password

```
sudo -u postgres psql postgres
\password postgres
```

- Enter the new **postgres** user password and repeat as required. Type Control+D or \q to exit the posgreSQL prompt
- Create a new database

```
sudo -u postgres createdb sml
```

- Change the terminal location to the **database** directory and ensure the **schema.sql** file is located in the directory
- Import the database schema into the **sml** database. The following command will import the schema onto the local PostgresSQL server, using the **postgres** user. The **psql** command will prompt for the password

```
sudo psql -U postgres -W -f schema.sql -h 127.0.0.1 -d sm
```

Other Database commands

- Login to postgres and prompt for password

```
psql -U postgres -h 127.0.0.1 -W
```

- List databases via psql

```
\list
```

- Drop database

```
drop database "sml"
```

- Create database

```
create database "sml"
```

Golang

In order to compile the server source code, the golang toolset must be installed. If only release versions are to be used then this step is not necessary.

The easiest method for installing the current version of the golang tool chain is to use the **godeb** project. More information regarding the **godeb** project can be found at the following URL:

https://github.com/niemeyer/godeb

A pre-built **godeb** binaries can be found at the following URL:

https://godeb.s3.amazonaws.com/godeb-amd64.tar.gz

https://godeb.s3.amazonaws.com/godeb-386.tar.gz

Using the **list** godeb command (as shown below) will display the various versions available.

```
./godeb list
1.6.2
1.6.2
1.6.1
1.6
1.5.4
```

The latest version can be installed using the following command:

```
godeb install 1.6.2
```

Logging

The server logs to it's own specific file, which requires the following steps:

- Make a directory for the log

  ```
  sudo mkdir /var/log/sml-server
  ```

- Change the directory owner to the user that will run the server

  ```
  sudo chown YOURUSER /var/log/sml-server
  ```

- Change the directory group to the user that will run the server

```
sudo chgrp YOURUSER /var/log/sml-server
```

HTTPS Certificate

The client uses HTTPS (TLS) to communicate with the server. An organisation specific certificate should be created.

- Generate a new RSA private key file. The example creates a 2048 bit key

```
openssl genrsa -out server.key 2048
```

- Generate a new certificate in X509 format. The certificate generation process requires the user to enter various organisation details. The following shows the command and some DEMO values: ``` openssl req -new -x509 -key server.key -out server.pem -days 3650

You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:COUNTRY State or Province Name (full name) [Some-State]:STATE Locality Name (eg, city) []:CITY Organization Name (eg, company) [Internet Widgits Pty Ltd]:COMPANY

Organizational Unit Name (eg, section) []:SECTION Common Name (e.g. server FQDN or YOUR name) []:COMPANY Email Address []:USER@ORG

```
## Systemd Service

To enable the SysMon analysis server to run on boot up, a Sys

- Create a directory in **/opt**
```

sudo mkdir /opt/sml

```
- Change the directory owner to the user we want to run the a
```

sudo chown USERNAME /opt/sml

```
- Change the directory group to the user we want to run the a
```

sudo chgrp USERNAME /opt/sml

```
- Copy the required files to the directory as listed below:
```

sml-server.config sml-setbind.sh sml server.key server.pem

```
- Set the appropriate configuration options in the config fil
- If the server is to be run on the lower TCP ports such as 8
```

#!/bin/sh chmod +x /opt/sml/sml sudo setcap cap_net_bind_service+ep /opt/sml/sml

```
- Next define the Systemd service config by creating a new se
```

sudo nano /etc/systemd/system/sml.service

```
- Copy the following to the service file or use the file loca
```

[Unit] Description=SML

[Service] ExecStart=/opt/arl/arl -c /opt/sml/sml.config

[Install] WantedBy=multi-user.target

```
- To test the service, run the following command:
```

systemctl start sml.service

```
- To check the service output, run the following command:
```

systemctl status sml.service

```
- If there are problems, then the service can be stopped by r
```

systemctl stop sml.service

```
- Once the configuration is correct and the server is running
```

systemctl enable sml.service ```