

Submitted By

KINGDOM OF LESOTHO

**Technical User Manual**

FOR VAC HELPLINE SYSTEM

Ministry of Social Development

Table of Contents

[1. Introduction 3](#_Toc87535348)

[2. Case Management System 3](#_Toc87535349)

[2.1 Install a web server 3](#_Toc87535350)

[Configuring Nginx 4](#_Toc87535351)

[2.2 Install MySQL 5](#_Toc87535352)

[2.3 Install PHP 5](#_Toc87535353)

[2.4 Install the CMS 6](#_Toc87535354)

[3. Install and configure Asterisk 6](#_Toc87535355)

[Install Asterisk 6](#_Toc87535356)

[Install SSL certificates 7](#_Toc87535357)

[Configuring Asterisk 8](#_Toc87535358)

[Conclusion 9](#_Toc87535359)

[Technical Manual Sign-Off 10](#_Toc87535360)

# Introduction

The solution is made up of two main components and a number of dependencies as described here. The two main parts are: Case Management Systems (CMS) and Call Module powered by Asterisk.

Asterisk runs best on Linux based operating systems and Centos 8 is recommended for this solution. All descriptions related to this version shall be based:

* Centos 8
* Nginx
* MySql

# Case Management System

This is a web solution developed on PHP, html and Javascript. It runs on a web (apache/nginx/httpd) server and uses MySql or PostgresQl databases.

To setup the solution, install the following and their dependencies. It is advisable to run the installations as a non-root user and have firewall running.

## 2.1 Install a web server

In our case, we will use Nginx web server

In order to install Nginx, we’ll use the dnf package manager, which is the new default package manager on CentOS 8.

Install the nginx package with:

sudo dnf install nginx

Enable and start the server:

sudo systemctl enable nginx

sudo systemctl start nginx

Check on the browser with your domain IP or using 127.0.0.1 if on current computer. The service runs on port 80 by default. You should be able to get a Nginx welcome page as below if the server is correctly and successfully installed.



### Configuring Nginx

Nginx can host several web application in different drectories and different domains. Since we have one application for our case, we will install our application on Nginx root folder. The default Nginx root directory is

/usr/share/nginx/html

Most users prefer changing the root directory to:

/var/www/html

Nginx configuration files are found in /etc/nginx/ directory and the default configuration file is /etc/nginx/nginx.conf. The root directory can be changed on the configuration file.

Another important Nginx directory is /etc/nginx/conf.d/, this directory contains server block configuration files, where you can define the websites that are hosted within Nginx. A typical approach is to have each website in a separate file that is named after the website’s domain name, such as helpline.conf.

For our case, we will have the application server block within the etc/nginx/nginx.conf configuration file as below: Update the configuration file with the below lines, some may already exist.

server {

listen 80;

listen [::]:80;

root /var/www/html;

index index.html index.htm index.nginx-debian.html;

server\_name uchl2.mglsd.go.ug www.uchl2.mglsd.go.ug;

location / {

try\_files $uri $uri/ =404;

}

}

To make sure that there are no syntax errors in any of your Nginx files, run:

sudo nginx -t

Once your configuration test passes, restart Nginx to enable your changes:

sudo systemctl restart nginx

With this and [CMS installation,](#_Install_the_CMS) the application will be available on the local IP and the domain on condition that the domain and the local server IP have been mapped to a public IP.

Also, for php to work with nginx, install php-fpm with the following command:

sudo systemctl enable php-fpm

sudo systemctl start php-fpm

Configure php-fpm listen to php-fpm.sock in /etc/php-fpm.d/www.conf. The default path for the sock file is /var/run/php-fpm/php-fpm.sock

## 2.2 Install MySQL

Run the following command to install the mysql-server package and a number of its dependencies:

sudo dnf install mysql-server

With that, MySQL is installed on the server but it isn’t yet operational. The package you just installed configures MySQL to run as a systemd service named mysqld.service. In order to use MySQL, there is need to start.

sudo systemctl start mysqld.service

Then set MySQL to start whenever the server boots up

sudo systemctl enable mysqld

To secure MySQL, run the following commands and follow the prompts.

sudo mysql\_secure\_installation

This will enable one to set a password for the root user, however, it is recommended not to run the system with MySQL root user

## 2.3 Install PHP

The installed web server can execute files which do not need compilations such as html files, php compiler will need to be installed for php files to be executed on the server.

To install the php and php-mysqlnd packages using the dnf package manager, run:

sudo dnf install php php-mysqlnd

Once the installation is complete, restart the Nginx server in order to enable the PHP module.

## 2.4 Install the CMS

This is done by simply pulling the code from the code repository, either [the solution github](https://github.com/Bitz-IT-Consulting/helpline) repository or a local directory.

# Install and configure Asterisk

## Install Asterisk

The current system runs and operates optimally with Asterisk 16 and is recommended to install from source. The following steps describe the process:

Download [Asterisk 16 source files](https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz) from to your server using the following command:

wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz

Other Asterisk dependencies we will install are:

1. Lipri - The libpri library allows Asterisk to communicate with ISDN connections. We will need because of using DAHDI with ISDN interface hardware such as E1. [Download Link](https://downloads.asterisk.org/pub/telephony/libpri/libpri-current.tar.gz)
2. Dahdi - The DAHDI library allows Asterisk to communicate with analog and digital telephones and telephone lines, including connections to the Public Switched Telephone Network, or PSTN. [Download Link](https://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz)

After downloads, untar the source files using the following commands:

tar -zxvf libpri-current.tar.gz

tar -zxvf dahdi-linux-complete-current.tar.gz

tar -zxvf asterisk-16-current.tar.gz

Build and install dahdi using the following commands:

cd dahdi-linux-complete-current

make

make install

make config

Build and install LibPRI using the following commands:

cd libpri-current

make

make install

Now compile and install asterisk with the following commands:

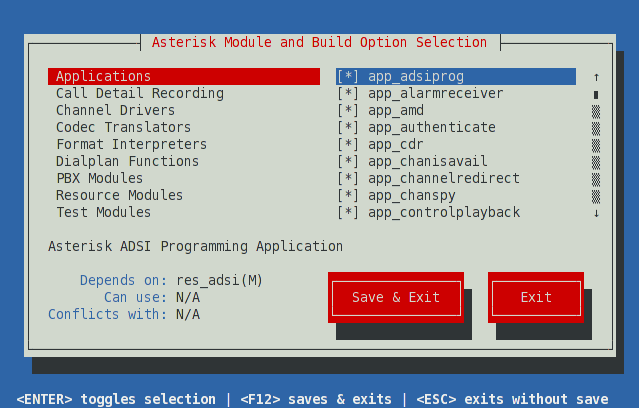
cd asterisk-16-current

./configure

Resolve any dependency issue that may cause the ./configure to fail then use the following command to select the desired asterisk menu items.

make menuselect

This command will have a screen like this:



To compile Asterisk, simply type make at the Linux command line.

make

Then install Asterisk using the following command:

make install

At this point, Asterisk installation is complete and can start or check the service status using the commands below:

/etc/init.d/asterisk start

/etc/init.d/asterisk status

## Install SSL certificates

Asterisk provides a utility script, ast\_tls\_cert in the contrib/scripts source directory. We will use it to make a self-signed certificate authority and a server certificate for Asterisk, signed by our new authority.

Still on the Asterisk installation source directory, run the following commands:

sudo mkdir /etc/asterisk/keys

sudo contrib/scripts/ast\_tls\_cert -C helpline. socialdevelopment.gov.ls -O "Lesotho Helpline" -b 2048 -d /etc/asterisk/keys

Follow the prompts keenly not to skip any necessary information. A successfully certificate generation will create the following files on the certificate directory mkdir /etc/asterisk/keys.

asterisk.crt

asterisk.csr

asterisk.key

asterisk.pem

ca.cfg

ca.crt

ca.key

tmp.cfg

## Configuring Asterisk

To meet the system requirements for optimal operations, follow the following configuration steps with the respective parameters

To communicate with WebSocket clients, Asterisk uses its built-in HTTP server. Configure /etc/asterisk/http.conf as follows:

[general]

enabled=yes

bindaddr=0.0.0.0

bindport=8088

tlsenable=yes

tlsbindaddr=0.0.0.0:8089

tlscertfile=/etc/asterisk/keys/asterisk.crt

tlsprivatekey=/etc/asterisk/keys/asterisk.key

Configure PJSIP on /etc/asterisk/pjsip.conf as follows:

[webrtc\_client]

type=aor

max\_contacts=5

remove\_existing=yes

[webrtc\_client]

type=auth

auth\_type=userpass

username=webrtc\_client

password=webrtc\_client ; This is a completely insecure password! Do NOT expose this

; system to the Internet without utilizing a better password.

[webrtc\_client]

type=endpoint

aors=webrtc\_client

auth=webrtc\_client

dtls\_auto\_generate\_cert=yes

webrtc=yes

; Setting webrtc=yes is a shortcut for setting the following options:

; use\_avpf=yes

; media\_encryption=dtls

; dtls\_verify=fingerprint

; dtls\_setup=actpass

; ice\_support=yes

; media\_use\_received\_transport=yes

; rtcp\_mux=yes

context=default

disallow=all

allow=opus,ulaw

Additionally, for asterisk configurations, make a backup of the following files in /etc/asterisk/ directory and copy the same files from CRM directory /helpline/configs/ readily configured into the asterisk configuration directory /etc/asterisk/

manager.conf

pjsip.conf

extensions.conf

confbridge.conf

After the configurations, ensure you restart asterisk for the configurations to take effect.

# Conclusion

This technical document followed step by step shall end up in a working solution for a call center as provided by Bitz IT Consulting. It is good to keep in mind the dynamics of the open source community and updates in the installation procedures for upgraded versions of the dependencies where upgrades are not avoidable.

# Technical Manual Sign-Off

|  |  |
| --- | --- |
| By signing this document, I acknowledge that I have received stated deliverables to the agreed quality levels. | |
| **Ministry of Gender Labor and Social Development**  **Permanent Secretary**  **Name:** Aggrey David Kibenge  **Email:** [ps@mglsd.go.ug](mailto:ps@mglsd.go.ug) | **Signature:** |
| **Date:** |
| **Ministry of Gender Labor and Social Development**  **Ag. Principal Information Technology Officer**  **Name:** Emmanuel Yeka  **Email:** [emmanuel**.**yeka**@**mglsd**.**go**.**ug](mailto:emmanuel.yeka@mglsd.go.ug) | **Signature:** |
| **Date:** |
| **Bitz IT Consulting Ltd: Director**  **Name:** James Nganga  **Email:** [james.nganga@bitz-itc.com](mailto:james.nganga@bitz-itc.com) | **Signature:** |
|  | **Date:** |