SOP: OFFLINE AfyaSTAT INSTALLATION USING PIE-HOLE

This is a technical user guide how to setup an offline AfyaSTAT instance using the pie hole technology. This facilitates local DNS server management and local DHCP services without the need for internet connectivity. The procedure outlined here are specific to Ubuntu 20.04LTS platform.

Objective: To provide simplified guidance to users on how to install and setup offline AfyaSTAT on Ubuntu environment.

Target audience: SI, Dev, M&E, HIS

Last Update: 7th Jan 2022

Requirements

The following dependencies are required for successful AfyaSTAT and will be automatically installed by the script.

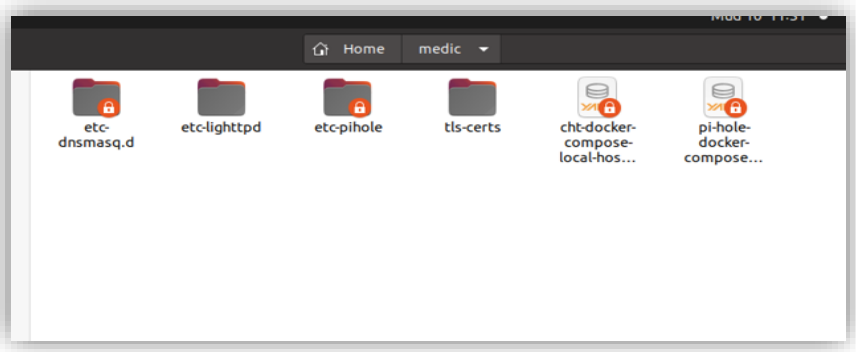
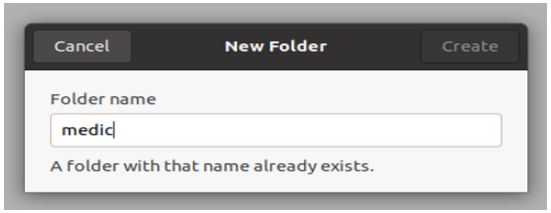
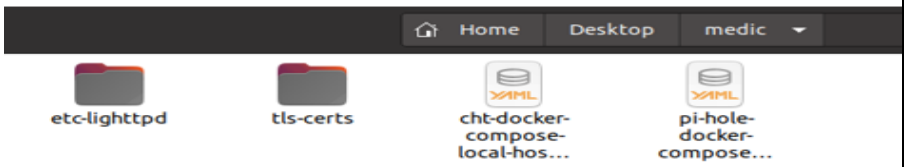
i. Functional instance of Ubuntu 20.04
ii. A working instance of AfyaSTAT
iii.

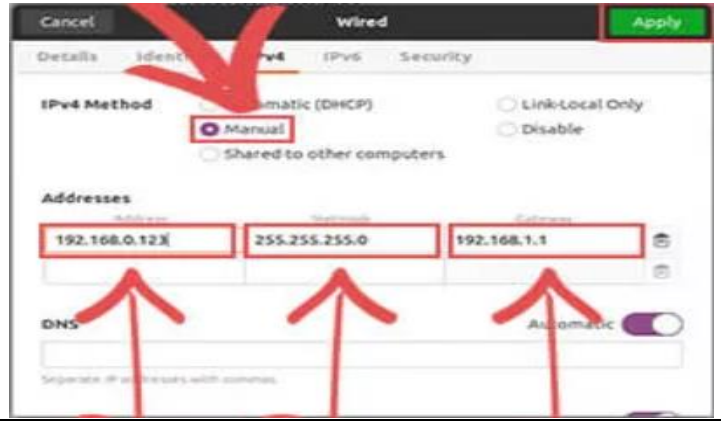
Other Requirements:

- A router with the IP
- A WiFi access point (AP) (using the one in the router)
- Ensure Docker and Docker compose is on the server

General Instruction:

Unless otherwise specified, all commands are run on the Ubuntu server as the root user. All commands should be run from the same location of /root.

	Step 1: Obtain installation resources: Obtain correct AfyaSTAT installation resources and copy to **Home folder/directory**. Confirm the following resources: i. /cht-docker-compose-local-host.yml ii. /etc-dnsmasq.d iii. /etc-pihole iv. /etc-lighttpd v. /medic-srv vi. /pi-hole-docker-compose.yml vii. /tls-certs The Installation folder should be name as **AfyaStat** while the folder containing the forms should be named as **AfyaSTAT forms** respectively.	 Content of the AfyaSTAT resources folder
	Step 2: Create folder to contain the resources Create a folder either on the Desktop or in Home directory. Give it an appropriate  name (e.g medic). Copy all the files downloaded into this folder as shown here ➔: The files include - Etc-lighttpd - Tls-certs - Cht-docker-compose-local-host - Pi-hole-docker-compose	

<table>
<tr>
<td></td>
<td>Step: 3: Asign Static IP address to the host server
Assign a static IP address to the server using the desktop interface; Refer to this guide on how to set it up on Ubuntu 20.04. The illustration shows sample IP settings during configuration.</td>
<td> 
</td>
</tr>
<tr>
<td></td>
<td>Step 4: Ensure port :53 is free

- Ensure port :53 is free on the server to allow for DNS binding. Follow this guide on how to free up port :53
- The illustration shows the two parameters to be set i.e <code>DNS</code> and <code>NDSStubListener</code></td>
<td><pre>[Resolve]
DNS=1.1.1.1
#FallbackDNS=
#Domains=
#LLMNR=no
#MulticastDNS=no
#DNSSEC=no
#DNSOverTLS=no
#Cache=no
DNSStubListener=no
#ReadEtcHosts=yes</pre></td>
</tr>
</table>

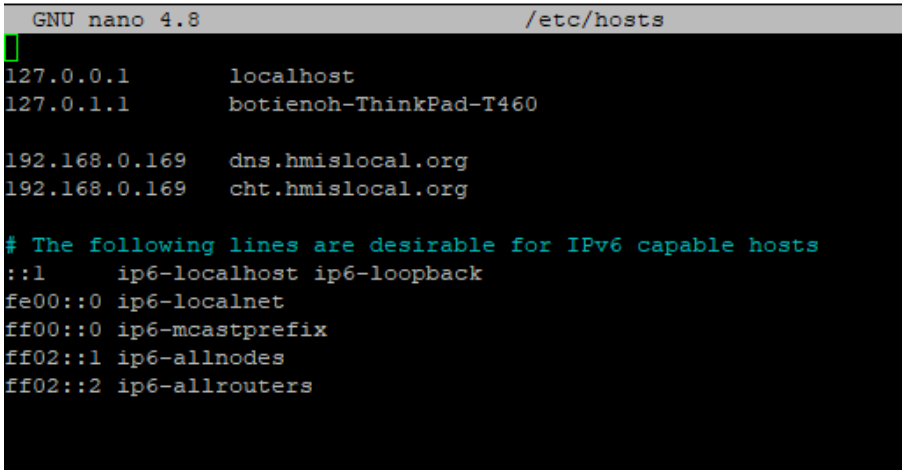
Step 5: Edit the hosts file and add new entries

- You need to add two pie-hole entries in the host file. This is besides the exisiting DNS settings already set in the host file.
- In the terminal window, open the hosts file in edit mode as follows:
  `sudo nano /etc/hosts` **[ENTER]**

- The entries will have the static address of the server. In this case my server has a static ip address of :192.168.100.6 as shown below:

```
192.168.100.6      dns.hmislocal.org
192.168.100.6      cht.hmislocal.org
```

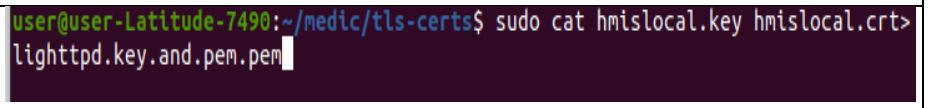
Note:
192.168.100.6 -This is the static IP address of the server you assigned. This varies based on the network.

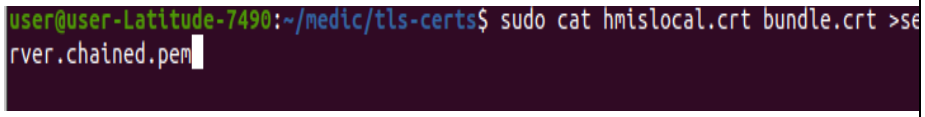


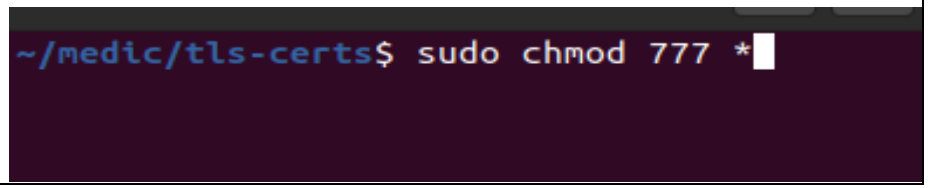
Step 6: Extract the resources

- From the downloaded resources, locate the zipped folder named "**hmislocal.zip**" and extract the content into "**tls-cert**" folder (the folder is initially blank). The files should appear as follows:

- In the terminal window, change directory into the tls-certs and run the following commands to prepare the extracted files for use. Only do this after extraction process above.







```
sudo cat hmislocal.key hmislocal.crt >
lighttpd.key.and.pem.pem [ENTER]

sudo cat hmislocal.crt bundle.crt >
server.chained.pem [ENTER]

sudo chmod 777 * [ENTER]
```

Step 7: Set Admin Password for Pi-Hole web access

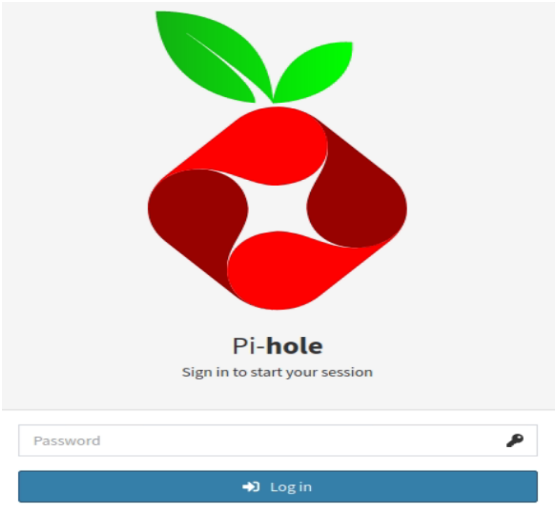
- Locate and open the file **pi-hole-docker-compose.yml**. Uncommment the **WEBPASSWORD** line and set appropriate password for the Pi-Hole as shown.
- Save and close the file when done.

```
version: "3"

services:
  pihole:
    container_name: pihole
    image: pihole/pihole:latest
    network_mode: "host"
    restart: unless-stopped
    environment:
      TZ: 'America/Chicago'
      WEBPASSWORD: 'test1234'
      WEB_PORT: 8081
    volumes:
      - './etc-pihole/:/etc/pihole/'
      - './etc-dnsmasq.d/:/etc/dnsmasq.d/'
      - './tls-certs/:/etc/tls-certs/'
      - './etc-lighttpd/external.conf:/etc/lighttpd/external.conf'
    cap_add:
      - NET_ADMIN
```

Step 7: Start the container

- Start the Pi-Hole container by issuing this command.
  `docker-compose -f pi-hole-docker-compose.yml up --detach`
- Access pi-hole web application via https://dns.hmislocal.org:8443
- Log in using the password set in step 6 above.



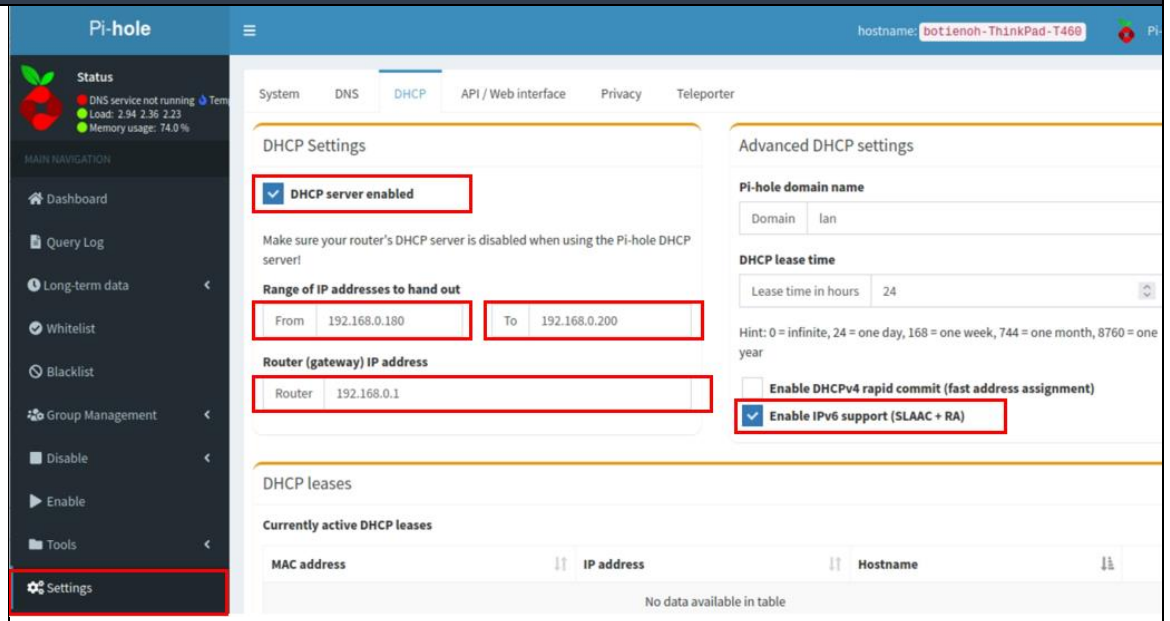
PI-HOLE CONFIGURAION

Step 8: DHCP Settings

- On the left most menu, go to "**Settings**" → "**DHCP**" and turn on DHCP, ensuring "**range**" and "**router**" are set correctly for your LAN. For example, the IP range could be from 192.168.100.101 - 192.168.100.151 and router 192.168.100.1 in my case.

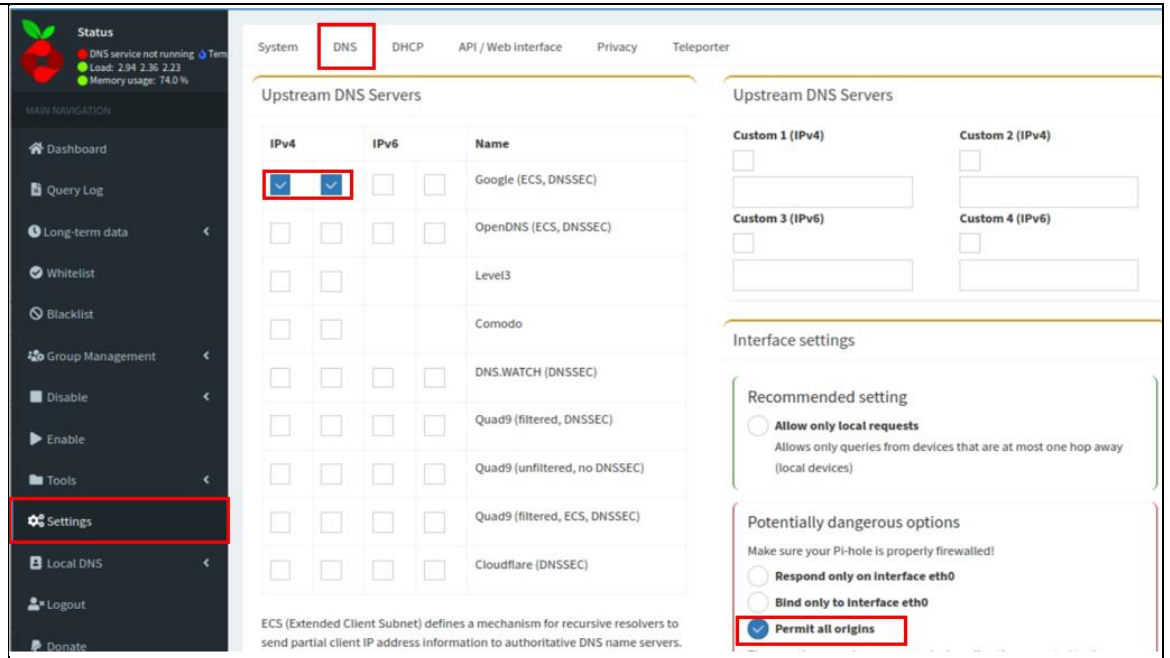
- Tick the checkbox to Enable IPv6 support

- Click "Save" on the bottom right



Step 9: DNS Settings

- On the left most menu, go to “Settings” → “DNS” → “Interface listening behavior” and set it to “Listen on all interfaces, permit all origins”. Click “Save” on the bottom right as shown ➔:



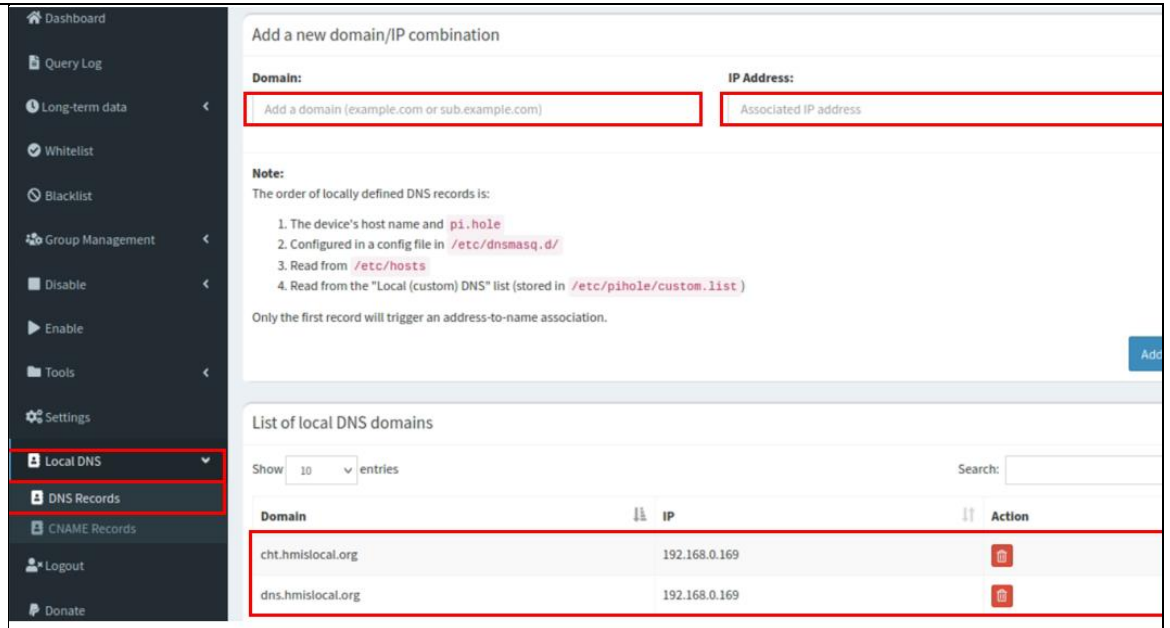
<table>
<tr>
<td></td>
<td>

Step 10: DNS Settings

On the left most menu, go to “Local DNS” → “DNS Records” and add two entries for your CHT instance and Pi-hole instance. These need to match the CN in your certificate.

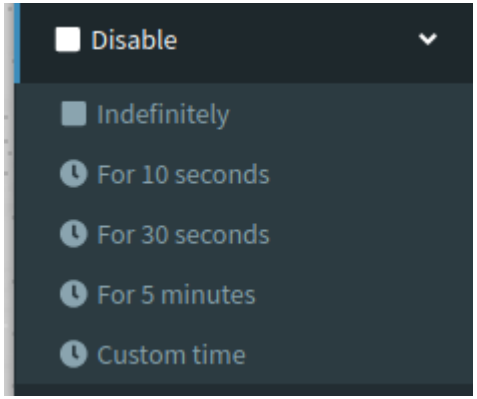
Note:
IP address depends on the static IP address of the server where you host the DNS server. In this example we use 192.168.100.6

```
cht.hmislocal.org  192.168.100.6
dns.hmislocal.org  192.168.100.6
```

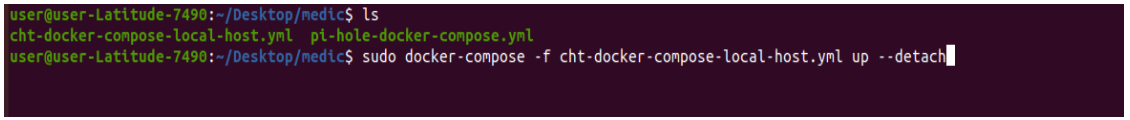
</td>
<td> 
</td>
</tr>
<tr>
<td></td>
<td>

Step 11: Disable the DNS Filtering

- On the left most menu, go to “Disable” and choose “Indefinitely” so there is no DNS filtering

</td>
<td> 
</td>
</tr>
</table>

INSTALLATION OF MEDIC - OS		
	Step 12: Check / Export the COUCHDB ADMIN PASSWORD - Check if COUCHDB ADMIN Password is already set. `echo $DOCKER_COUCHDB_ADMIN_PASSWORD` - Where the password is not set, you can export it as follows:	`sudo export OCKER_COUCHDB_ADMIN_PASSWORD=cb6f4d4b-73cc-4c42-97cb-0db5a631190a`

	Step 13: Install the medic-os by executing the .yml file
	- Remove previous instances of medic-os and haproxy as follows:
	`sudo docker stop haproxy`
	`sudo docker stop medic-os`
	`sudo docker rm medic-os`
	`sudo docker rm haproxy`
	`sudo docker volume rm medic-data`
	`sudo docker system prune -y`
	`sudo docker prune -a –volumes`
	- Open the terminal **where cht-docker-compose-local-host.yml** is located as shown below and Start the cht instance by running
	`sudo docker-compose -f cht-docker-compose-local-host.yml up –detach`
	(This installs medic-os)
		

Step 14: Verify the medic-os is running

- Execute the following command to verify if medic-os is running
  `docker ps -a`
- Check on the output for medic-os instance.

```
botienoh@botienoh-ThinkPad-T460:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED
STATUS                  PORTS     NAMES
bec4fecf548b   medicmobile/medic-os:cht-3.9.0-rc.1   "/bin/bash -l /boot/…"   4 hours ago
Up 4 hours                        medic-os
47fe51b1cc19   medicmobile/haproxy:rc-1.17          "/entrypoint.sh -f /…"   4 hours ago
Up 4 hours                        haproxy
a00e34bcb6e2   pihole/pihole:latest                 "/s6-init"               4 hours ago
Up 4 hours (healthy)              pihole
botienoh@botienoh-ThinkPad-T460:~$
```

Step 15: Configurations for medic-os license

- Enter the medic-os container with this command:
  `sudo docker exec -it medic-os /bin/bash`
- Press **[ENTER]**
- Open the nginx config file in edit mode as follows:

```
sudo nano srv/settings/medic-core/nginx/nginx.conf file
```

Edit the nginx.conf file by changing the following two certificate lines

```
ssl_certificate           /srv/settings/medic-core/nginx/private/default.crt;
ssl_certificate_key       /srv/settings/medic-core/nginx/private/default.key;
```

To look like this:

```
ssl_certificate           /etc/tls-certs/server.chained.pem;
ssl_certificate_key       /etc/tls-certs/hmislocal.key;
```

Also edit the following section from **haproxy** to **localhost**

```
upstream couchdb {
    server haproxy:5984;
}
```

```
upstream couchdb {
 server localhost:5984;

}
```

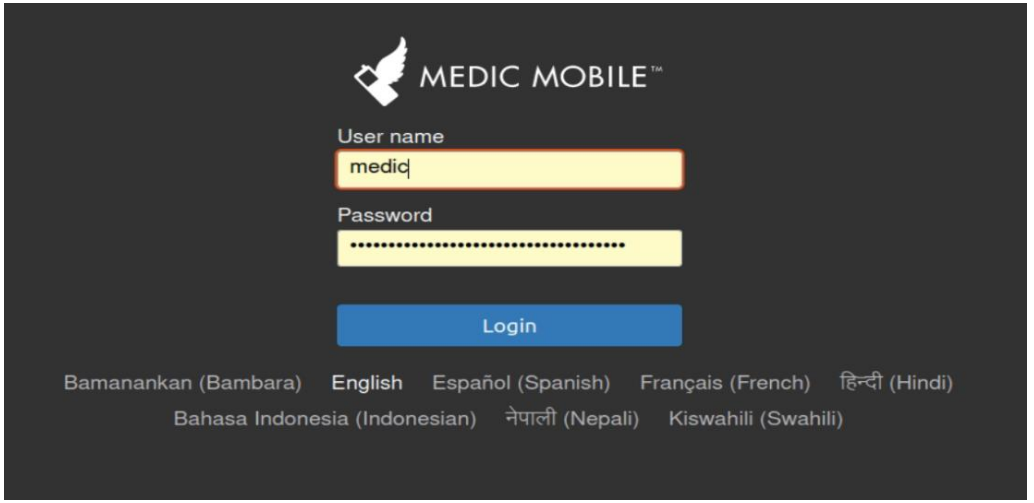
- Exit and write-out the nginx.config file by pressing Ctrl + X, then Press “Y” and then press [ENTER]
- Restart the web srver with the follwinf command: **`/boot/svc-restart medic-core nginx`**

Step 16: Launch afyaSTAT

- Launch afyaSTAT instance through the following URL:

  https://cht.hmislocal.org

- If all the settings are alright and the SSL certificate is valid, you should be able to successfully launch afySTAT page as shown.



	Step 17: Disable DHCP from the router - The Pi-Hole service once properly setup up will take over assignment of dynamic IP addresses to all the connecting devises. Therefore, it is advisable that you turn off the DHCP service of your router. - Depending on the type of your network access point, locate the DHCP settings and uncheck the pox or chose “disable” and restart the unit.	

Step 18: Connecting devices

Now that your CHT instance is available via DNS, DHCP and TLS, any device on the network can connect to it without Internet. Disconnect your LAN so all devices are fully offline with no Internet connection.

On Android device

- Connect to the AP on your LAN
- After installing launch the APK, choose “**Custom**” to select which CHT instance to use
- Enter https://cht.hmislocal.org and tap “Save”.

Note:

Some configurations of Android may not like being connected to an Access Point with no Internet access.

On a Desktop Browsers

- Connect to the AP on your LAN or via Ethernet
- In a browser https://cht.hmislocal.org

Warning: The CHT does not support the Safari browser on macOS

TROUBLESHOOTING

TROUBLESHOOTING OFFLINE AFYASTAT INSTALLATION

Try the following if your instance of Afyastat fails to start. You can identify the issue by checking the medic-os logs. Here are possible errors

i. **Error 404: Nginx Cannot Start**

This might be issues with the browser version.

- Go back to the folder that contains the setup files (created in Step 2 above).
- Open the **cht-docker-compose-local-host.yml file and edit the image line from .2 to .1 as shown below.**

Save and close the file.

- Remove the **medic-os** and **haproxy** containers redo the setup from step 13 above.
- Try accessing afyaSTAT once more from the browser.

THE END