Usage instructions:

1. Launch the product via 1-click. **Please wait until** the instance passes **all** status checks and is running. You can connect using your Amazon private key and '**ubuntu**' login via SSH client.

- Optional: To update software, use: **sudo apt update** and **sudo apt upgrade**

2. This Instance has been preconfigured with Docker for faster deployment. Change into the proper directory to get started.

cd librenms/docker-librenms

3. Start Docker:

sudo docker-compose up -d

4. Check if the Docker containers are running and whether LibreNMS is up and accessible:

sudo docker-compose ps

5. After configuring you can access LibreNMS web GUI by navigating to:

http://your_instance_public_ip:8000

6. Use "**admin**" and "**CC5150!!!**" for initial username & password login.

Optional: Test Polling

1. **Delete all test devices in the LibreNMS Web GUI under “All Devices”**

2. **Remove Test containers.**

   sudo docker rm -f snmp-test

3. **To Start a fresh SNMP Test Device Container**

This container simulates a real SNMP device on your LibreNMS network.

```
sudo docker run -d \
  --name snmp-test \
  --network docker-librenms_librenms_net \
  -p 161:161/udp \
  polinux/snmpd
```

- This exposes an SNMP v2c device with community **public** on the container's hostname (here: snmp-test).

---

2. Add the Test Device to LibreNMS via Web UI

1. **Open LibreNMS Web UI**:
   Visit http://<your-server-ip>:8000 in your browser.
2. **Login** as admin.
3. Go to **Devices > Add Device**.
4. **Enter details:**
   - **Hostname**: snmp-test
   - **SNMP Version**: v2c
   - **Community**: public
   - (Leave other fields default; optional: set location/description)
5. **Click Add Device.**

---

3. Test SNMP Connectivity (Optional, from LibreNMS container)

```
sudo docker exec -it librenms bash
snmpwalk -v2c -c public snmp-test sysDescr.0
```

Should return the Linux string for the snmp-test container.

```
exit
```

---

4. Manually Trigger Discovery & Poller

Inside the LibreNMS container:

```
sudo docker exec -it librenms bash
cd /opt/librenms
```

Find your device ID (should be 1 or 2 if this is the first/second device)

```
php artisan tinker
```

Then in tinker:
\DB::table('devices')->select('device_id', 'hostname')->get();
 # Note the device_id for snmp-test, say it's 3

exit

Run discovery (registers all SNMP info)

php discovery.php -h <device_id> -vvv
Ex: **php discovery.php -h 3 -vvv**

Run poller (collects data)
php lnms device:poll <device_id> -vvv
Ex: **php lnms device:poll 3 -vvv**

- Replace 3 with your actual device ID.

---

5. Confirm Polling in the Web UI

- Go to **Devices > All Devices**.
- Click on **snmp-test**.
- Check **Last Polled** timestamp and status.
- You should see SNMP and system data populated.

AWS Data

- Data Encryption Configuration: This solution does not encrypt data within the running instance.
- User Credentials are stored: /root/.ssh/authorized_keys & /home/ubuntu/.ssh/authorized_keys
- Monitor the health:
    - Navigate to your Amazon EC2 console and verify that you're in the correct region.
    - Choose Instance and select your launched instance.
    - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.

Extra Information: (Optional)

Allocate Elastic IP

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html