SDN Launch Control

Network Architecture and Software Installation Guide

Authored by Keegan White (keeganwhite@taurinetech.com)

Overview

SDN Launch Control (Launch Control for short) is a management software that allows you to set up equipment and utilise the Software Defined Networking paradigm in various ways. The core functionality of the software beyond device set up is the management of networks utilising Artificial Intelligence (AI) and OpenFlow (OF).

This guide aims to assist you in understanding the network architecture and equipment required to set up a Software Defined-Network (SDN) and aid you in installing and running Launch Control on your server.

There is also an equivalent video demo for installing the Launch Control software and setting up a network.

Minimum Requirements

Architecture and Recommended Equipment

- A router/firewall with a Dynamic Host Configuration Protocol (DHCP) server running.
- A Raspberry Pi with at least three ethernet ports running Ubuntu **Server** 22.04 LTS operating system. We strongly suggest you run our custom image. Find out more about this below.
    - *NOTE*: the 24 LTS software is **not** a viable option at this time.
    - Ensure you have SSH access enabled for the device.
    - We highly recommend a Raspberry Pi 4 Model B with 4gb or 8gb of RAM. This document will use a Pi throughout as an example.
    - Ensure you have at least a 32 gigabyte SD Card.
    - You can add more ports to the Pi using USB to Ethernet adapters.
- A server running Ubuntu 22/24.04 LTS operating system to run Launch Control and other software on.
- An Access Point (AP).

Outcomes and Objectives

This guide will assist you in configuring and building your own SDN network. The final outcome of this will be the network seen in *Figure 1* and a production ready instance of the Launch Control software running.

The red connections in *Figure 1* represent traditional connections connected to your original network with your router/firewall. This original network is depicted by the red circle. The blue connections are connections that are controlled by the flowtable on the SDN switch that you will set up using Launch Control in this guide. This results in the blue circles which are OF-enabled

networks. This means that any user connected to the AP in the blue circle will have a connection that is fully controllable through the Launch Control UI and SDN.

The server in the red circle will be the one running Launch Control and the other tools we will set up in this guide. Finally the switch in the blue circle is your target device we will configure with Launch Control.

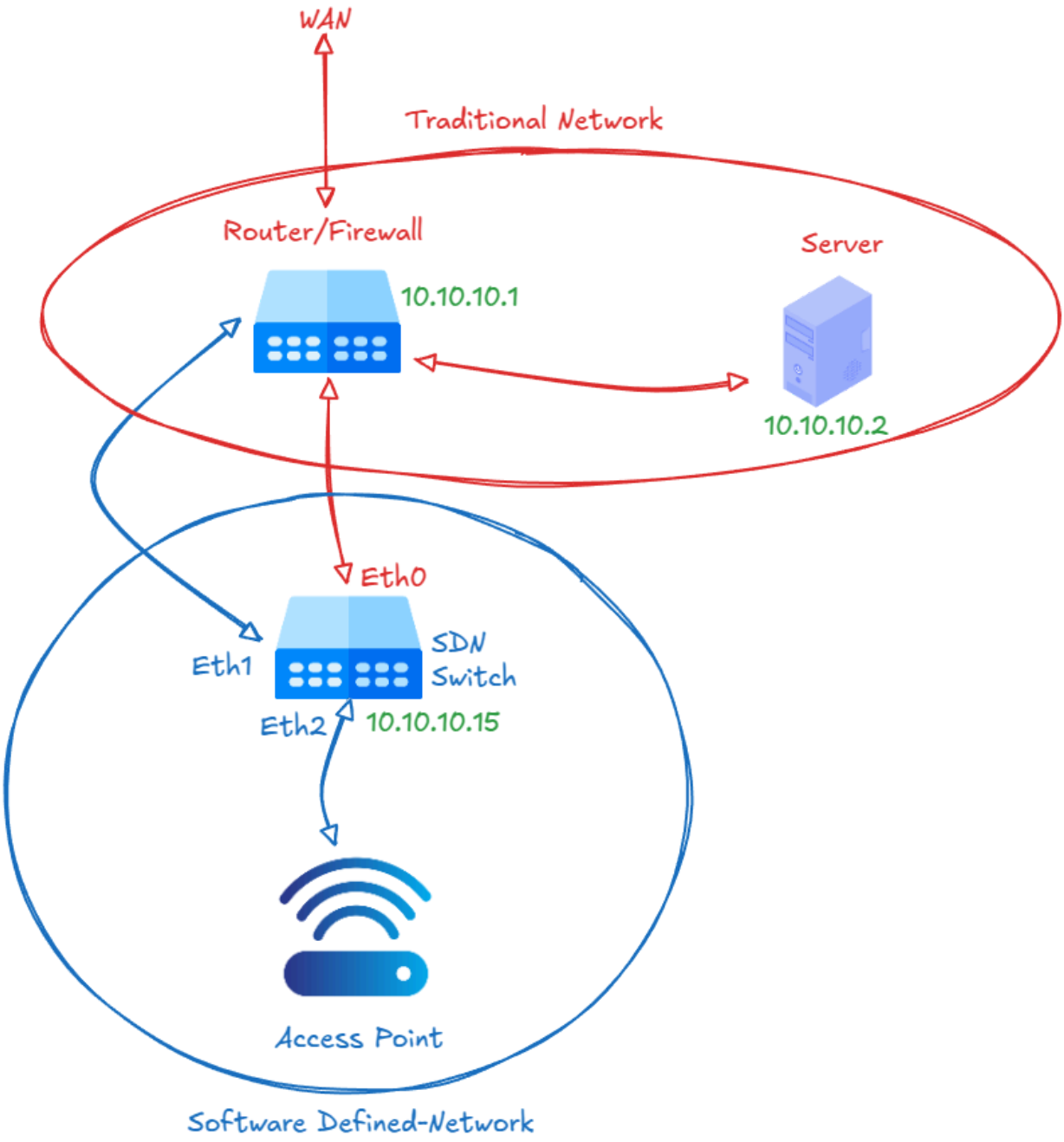

Figure 1: Configured network

Base Installation and Set Up Guide

The following guide will be divided into two parts. We will start by setting up a Raspberry Pi and Launch Control on our server. Once this is done we will connect our devices to our network and set up the network with SDN Launch Control.

1. *Configuring the Raspberry Pi*

Follow the instructions in this Github Repo to download our custom Raspberry Pi image. Once this has been downloaded you can check the integrity of the file by matching the sha256 hash of the file to the one in the Github repository.

Write this image to a SD card using Raspberry Pi Imager by selecting the '*Use custom*' option when selecting your operating system and then selecting our Pi image. Ensure you enable SSH access using password authentication and set a user name and password in the advanced options before writing the image to the SD card.

Please note Imager could show a status of '*Writing… 99%*' for quite a while but wait and it should move onto the verification step. Once the writing process is done and you have booted the Pi with the SD card it can take a few minutes for the device to configure itself and for Ubuntu to set up. You can monitor this via a monitor or wait ~10 minutes to be safe before trying to set up the device with Launch Control. Before continuing to step 2, run:

```
None
sudo apt update -y && sudo apt upgrade -y
```

The rest of this guide assumes you are using a Raspberry Pi 4B with our custom image and an SD card with at least 32GB of storage space.

2. *Installing SDN Launch Control*

Your server should be running Ubuntu 22 or 24 LTS for server or desktop. You will need Docker installed on your machine to run SDN Launch Control.

Important: Docker Permissions

Before proceeding, ensure that Docker can be run without sudo. If you get permission denied errors when running Docker commands, you need to add your user to the docker group. Find detailed instructions on the official site on how to set this up here.

Downloading the Code

Clone the Github repository, navigate into the resulting directory and check out the beta release:

```
None
git clone https://github.com/Taurine-Technology/sdn-launch-control.git
cd sdn-launch-control
git checkout v1.0.0-beta
```

Backend

Navigate into the backend directory and create an `.env` file:

```
None
cd backend
cp control_center/.env.example control_center/.env
```

You can use the `.env.example` for guidance but you will create a `.env` file that will look like this with your own values:

```
None
# .env
# Celery
CELERY_BROKER_URL=redis://redis:6379/1

# Channels
CHANNEL_REDIS_HOST=redis
CHANNEL_REDIS_PORT=6379

# Database
DB_HOST=pgdatabase
DB_NAME=postgres
DB_USER=postgres
DB_PASS=postgres

# default user login
DJANGO_SUPERUSER_USERNAME=admin
DJANGO_SUPERUSER_EMAIL=admin@example.com
DJANGO_SUPERUSER_PASSWORD=admin
```

```
# telegram
TELEGRAM_API_KEY=123
```

Your telegram API key is very important if you want to set up a Telegram notification system. Creating a telegram bot and getting an API key is free and easy to do. Follow the instructions on the telegram website here.

Automated Setup

Run the automated setup script from the `backend` directory:

```
None
chmod +x setup.sh
./setup.sh
```

The script will:

- Check all prerequisites and Docker permissions
- Validate your `.env` file
- Build the Docker image (this will take some time)
- Start services in the correct order with proper timing
- Provide status updates and helpful information

You will then be able to navigate to your API backend and manage your database at `<your-ip-address>:8000/admin` and log in using the variables you set in your `.env` file.

Frontend Setup

From the root of the cloned repo navigate into the `ui` directory and create a `.env.local` file :

```
None
cd ui/ui/
cp .example .env.local
```

Your file should look like this:

```
None
#.env.local
NEXT_PUBLIC_WS_OPENFLOW=ws://192.168.1.100:8000/ws/openflow_metrics/
```

```
NEXT_PUBLIC_WS_DEVICESTATS=ws://192.168.1.100:8000/ws/device_stats/
NEXT_PUBLIC_WS_CLASIFICATIONS=ws://192.168.1.100:8000/ws/flow_updates/
NEXT_PUBLIC_API_BASE_URL=http://192.168.1.100:8000/api/v1
```

Important Notes:

- Replace `192.168.1.100` with your actual server IP address
- Do NOT include a trailing forward slash at the end of `NEXT_PUBLIC_API_BASE_URL`
- Ensure your backend is running before starting the frontend

Automated Setup

Run the automated setup script from the `ui/ui` folder:

```
None
chmod +x setup.sh
./setup.sh
```

The script will:

- Check Docker permissions and prerequisites
- Validate your .env.local file
- Test connectivity to your backend
- Build the Docker image (this may take several minutes)
- Start the frontend application
- Provide status updates and access information

You will then be able to navigate to your API backend and manage your database at `<your-ip-address>:3000/`. Login with your username and password you set in the backend environment variable file. Once you have logged in you can change your language settings by clicking on the profile icon in the top right corner, navigating to settings and choosing your desired language.

3. Setting Up Your Networking Equipment

Below, *Figure 3.1*, is an example configuration of the network shown in *Figure 1*. The yellow ethernet cables represent OF controlled connections while the black cable is connecting the Raspberry Pi device to the traditional network. Similarly to *Figure 1*, the blue circle is the OF-enabled network and the red is the traditional network. The server at IP address 10.10.10.2 is running the Launch Control Software. The Raspberry Pi is replaceable by any device running an Ubuntu 22 Server Operating system but to follow this guide we recommend you use a Pi running our custom image as stated above.

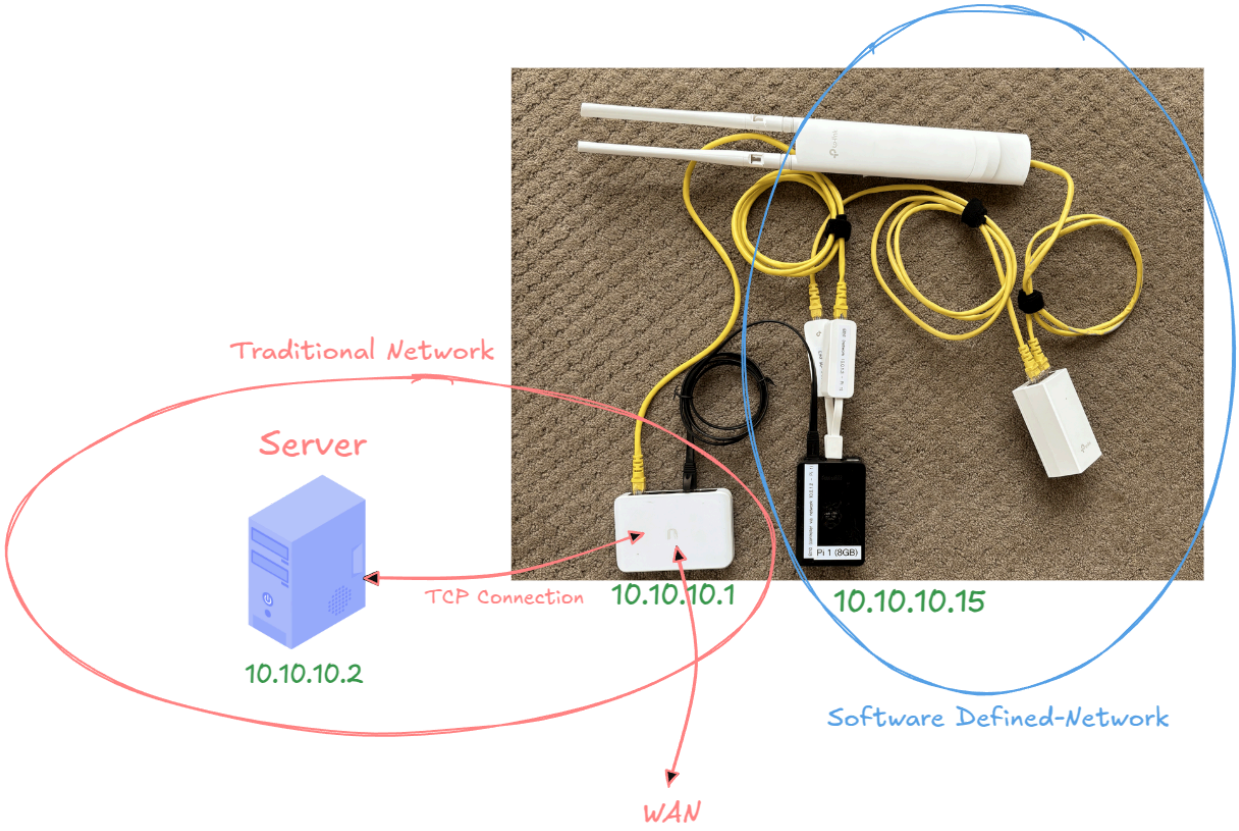

Figure 3.1: Example Network with Device Overlay

The ports used for the connection to the AP and back to the network are very important for the sections below. Notice that the top right USB port is connected to the network and the one below is connected to the AP, refer to *Figure 3.2*. Once your device has booted, connect your ethernet adapters starting with port 1. It is important to do this once the device has powered on when working with Ubuntu 22.

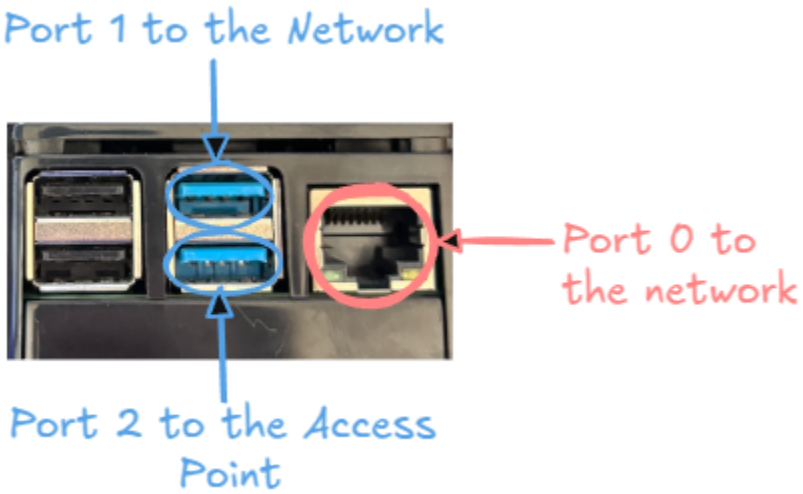

Figure 3.2: Raspberry Pi Ports

If you run into an issue with ports not showing up when configuring the device you can simply unplug and re-insert the adapters and they should show up. A reboot of the device can also solve this issue.

4. Setting Up Open vSwitch on your Target Device

The target device configured in the screenshots that follow is a Raspberry Pi with the IP address *10.10.10.15*. The first step to configuring a switch is to install OVS on it. This can be done on the 'Switches' page, navigate to it from the sidebar menu: Network > Switches. Once on the page click on the '+ Add Switch' button. This will open a dialogue where you fill in your switch's details as shown in *Figure 4.1* below.

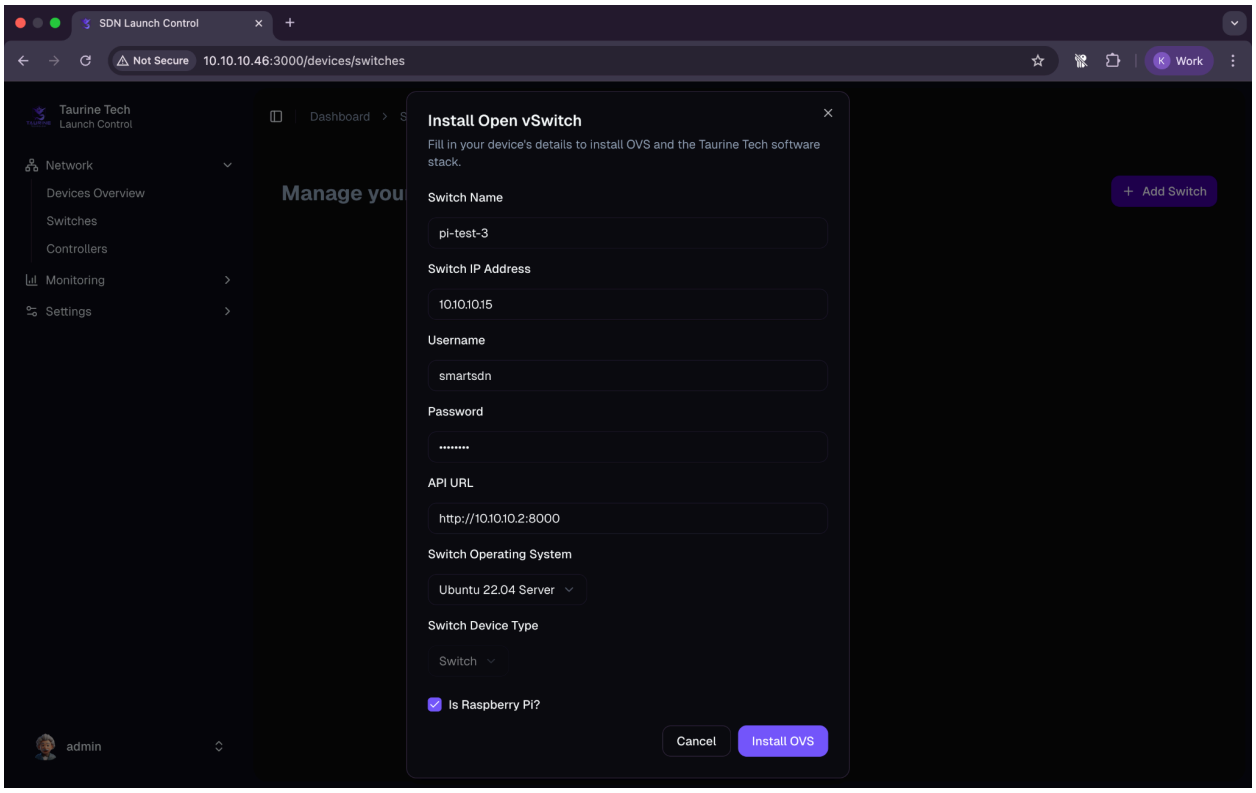

Figure 4.1: OVS Installation Dialogue

The important fields to note are the LAN IP Address field which will be specific to your network. This is the IP address of the switch assigned to *eth0*, refer to *Figure 1* and *Figure 3.1*, by your DHCP server. The username and password are your Ubuntu server username and password. The API URL is the IP address of your server, refer to *Figure 1*, assigned by your DHCP server. This must be prefixed by 'http://' and the port number of your API should be added to the end, the default port is '*8000*' so in *Figure 4.1* the API URL is 'http://10.10.10.2:8000'. Lastly ensure you mark your device as a Pi or ARM device if it is, see the last field in *Figure 4.1.*

This installation process can vary depending on your target device's processing power. For example a Raspberry Pi 4 can take up to 30 minutes. However, if you are using our custom image OVS is pre-installed and configured. This will greatly speed up the installation process, you can see example installation times in *Table 4.1*. Do not interrupt this process.

Device	Image	Installation Time
Raspberry Pi 3 Model B 1GB	Custom Ubuntu 22 Image	7 minutes and 15 seconds
Raspberry Pi 4 Model B 4GB	Custom Ubuntu 22 Image	3 minutes and 6 seconds
Raspberry Pi 4 Model B 8GB	Custom Ubuntu 22 Image	2 minutes and 45 seconds

Table 4.1: Installation Times

Once the installation has finished you will be able to navigate to the switch's page by clicking on the settings icon on the 'Switches' page next to the switch you set up, as seen in Figure 4.2.

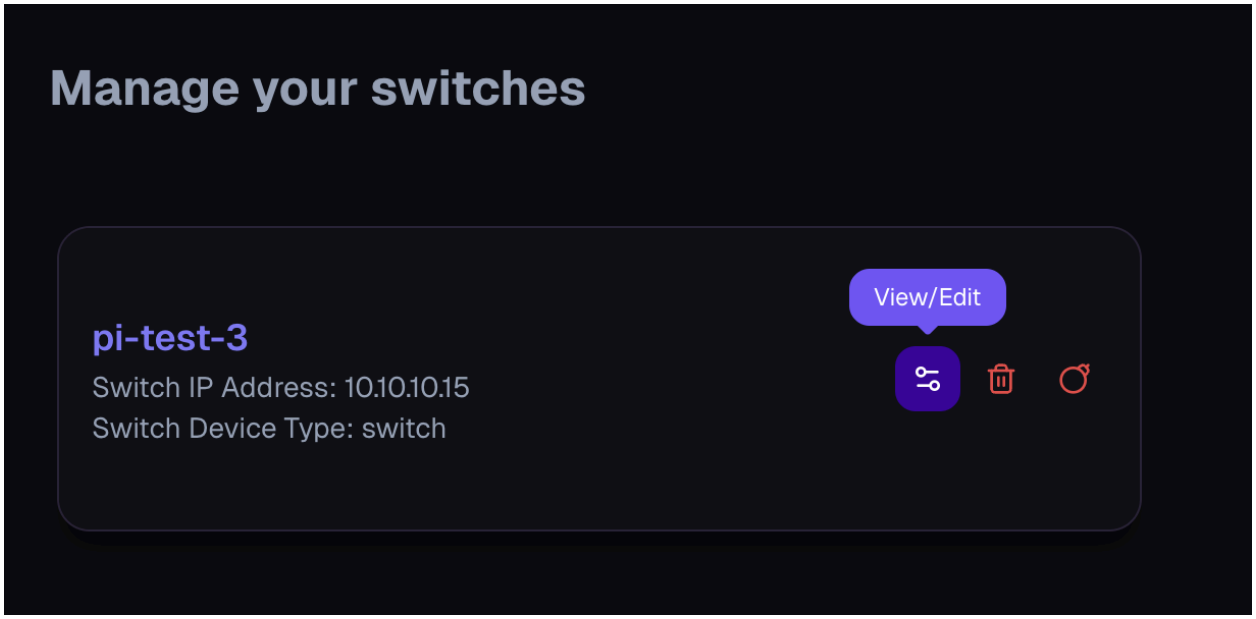

Figure 4.2: Switch Settings Icon

On this page you can monitor the device's resource utilization in real-time, as depicted in *Figure 4.3*. If this doesn't show up initially, refresh the page.

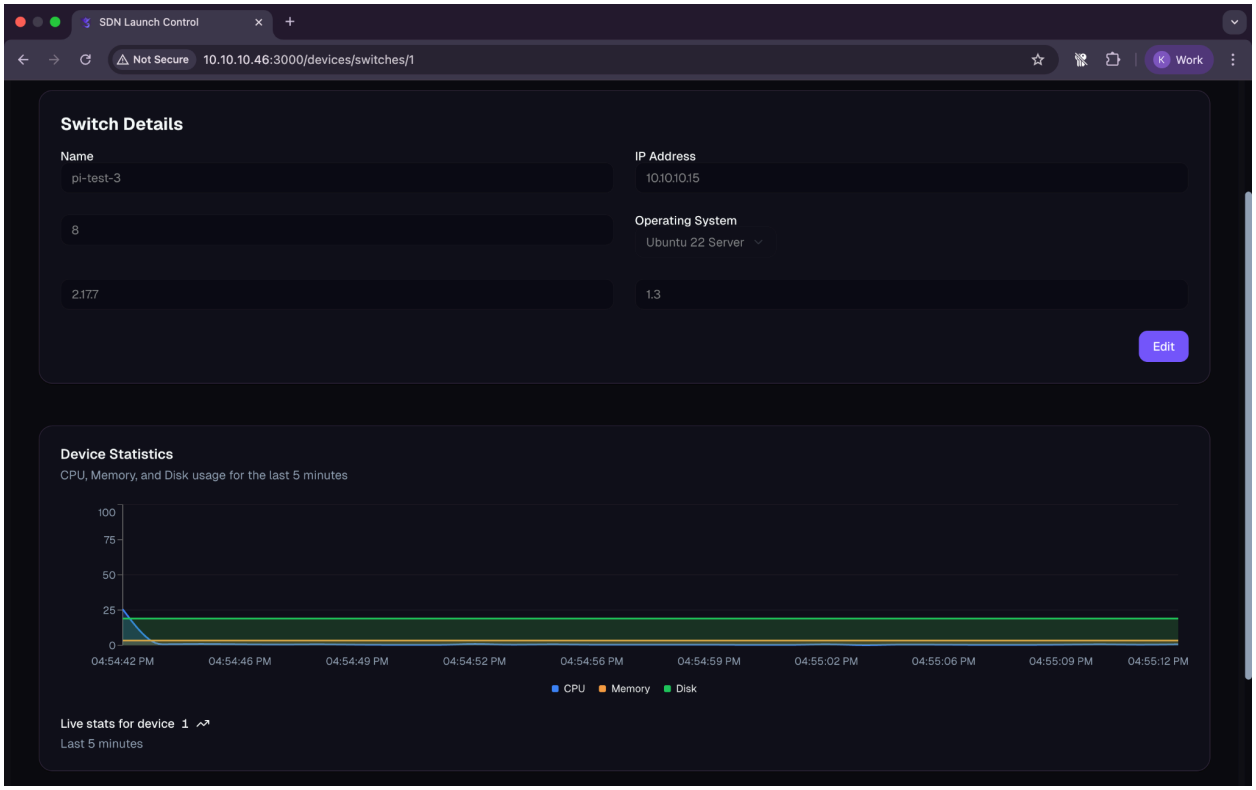

Figure 4.3: Device Resource Utilization Graph

5. Setting up an SDN Controller

In order for traffic to be routed in your network by your switch, it needs to have rules put in place so it knows where to forward traffic. This is done with an SDN controller. Navigate to the Controllers page by using the sidebar menu: Network > Controllers. Once on the controller page click on the '*+ Install Controller*' button. The dialogue that is shown will be used to install the Opendaylight SDN controller, see *Figure 5.1*. The Opendaylight controller is integrated with Launch Control's AI plugin and will allow you to install advanced traffic control rulesets without the need for much configuration.

The important fields to note in *Figure 5.1* are the LAN IP Address field which will be specific to your network. This is the IP address of your server. Launch Control will install the Opendaylight controller on your server to communicate with your switch. This is done through a TCP connection.

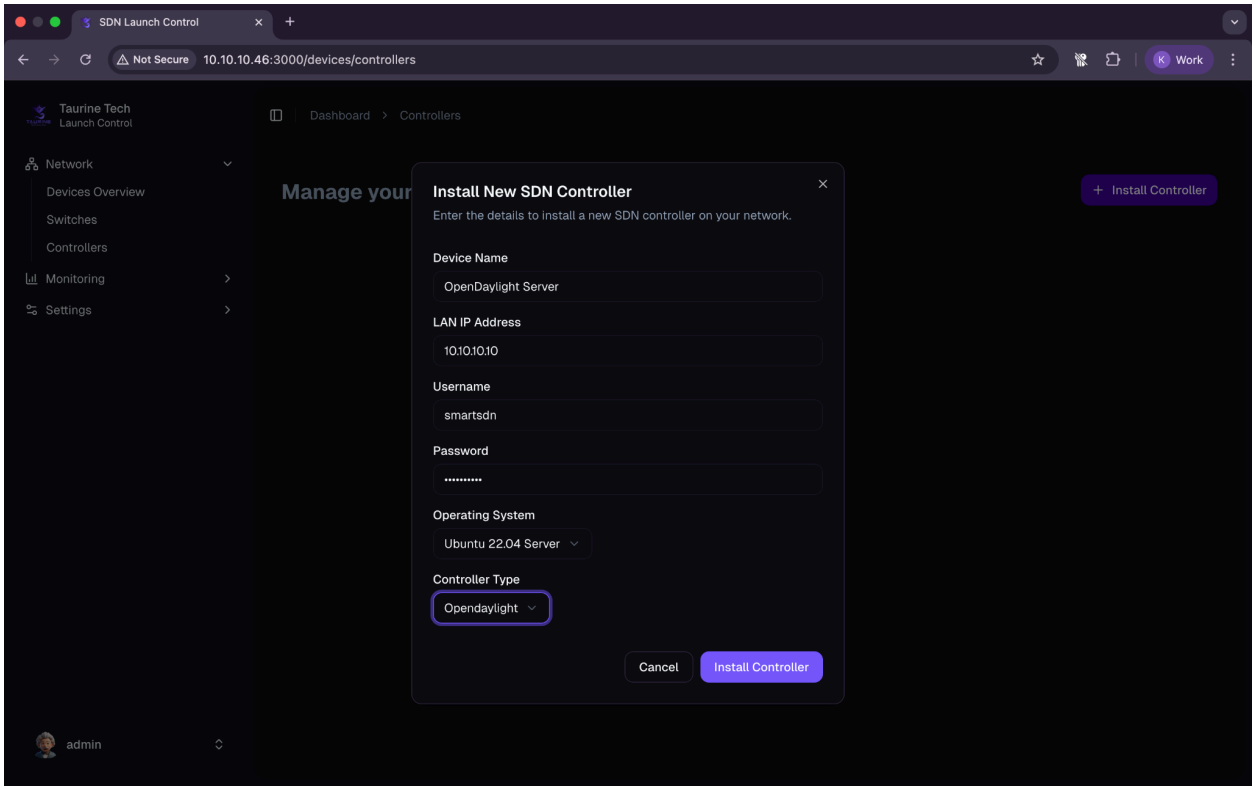

Figure 5.1: SDN Controller Creation Dialogue

6. OVS Bridges

In order to connect your AP to the network via your newly created switch you need to define an OVS bridge. This can be done by clicking on the settings icon next to the switch you created on the switches page, as seen in *Figure 4.2*. This will take you to a new page where you can edit and configure your switch.

You need to create a bridge in order to route traffic through your switch. To create a bridge you can scroll to the 'Bridge Configuration' section and click the 'Add Bridge'. In this form, refer to *Figure 6.1*, we want to assign a controller and two ports, a typical example for ports added via a USB to Ethernet adapter in the custom Ubuntu 22.04 LTS image would start with 'enx' as seen in *Figure 6.2* followed by the MAC address of the adapters. It is important that you **do not add 'eth0'** as this is the port that connects the switch to the controller and the network. Including eth0 would result in the switch going offline. The ports we add are the ones that have yellow ethernet cables connected to them in *Figure 3.1*. Once you have selected the controller you set up in the previous step from the drop down menu, you will see a field for the controller port, you can leave the default value of *'6653'* unchanged. The API URL will be the same as the one you set up in *Section 4*. This can take some time depending on the device, a Raspberry Pi 4 can take over 7 minutes.

Once this is done you will be able to monitor the devices port activity in the bridge statistics on this page, as seen in *Figure 6.3*. This graph depicts the real-time throughput of the ports. If it does not show up initially then refresh the page.

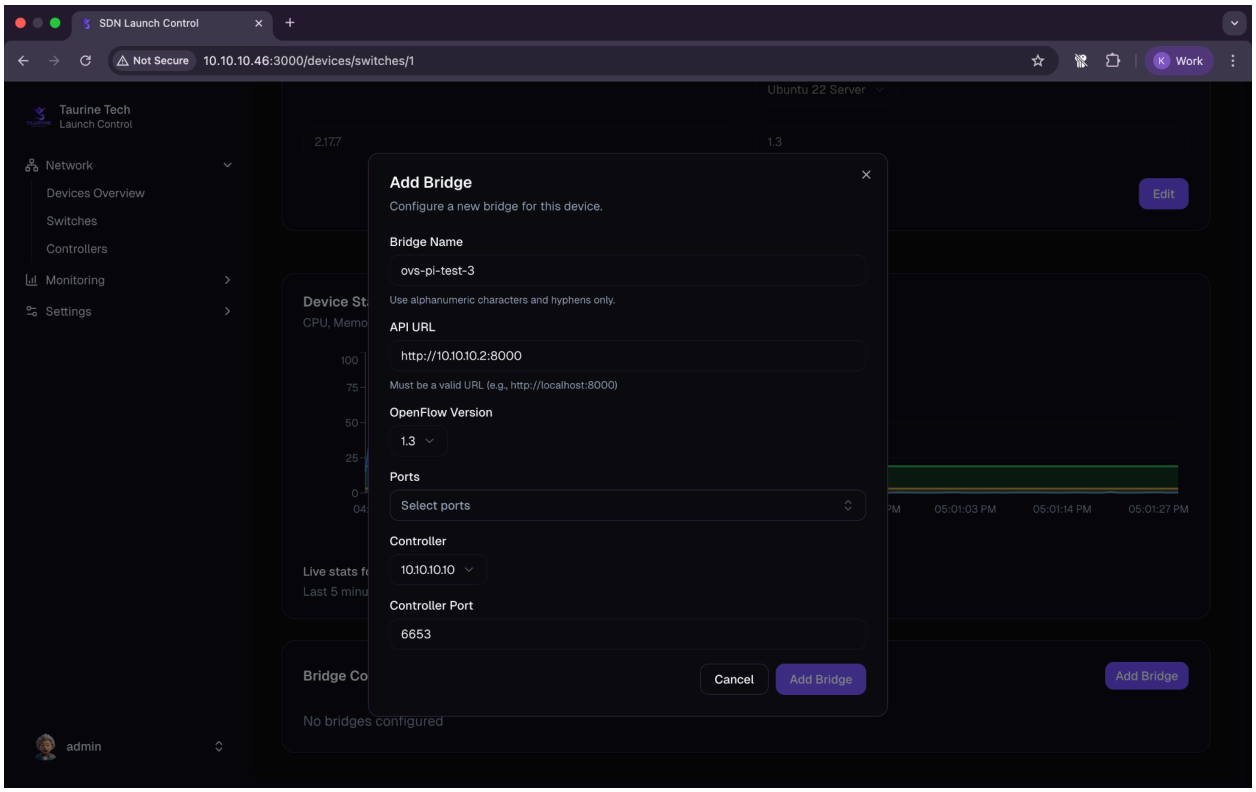

Figure 6.1: Bridge Creation Dialogue

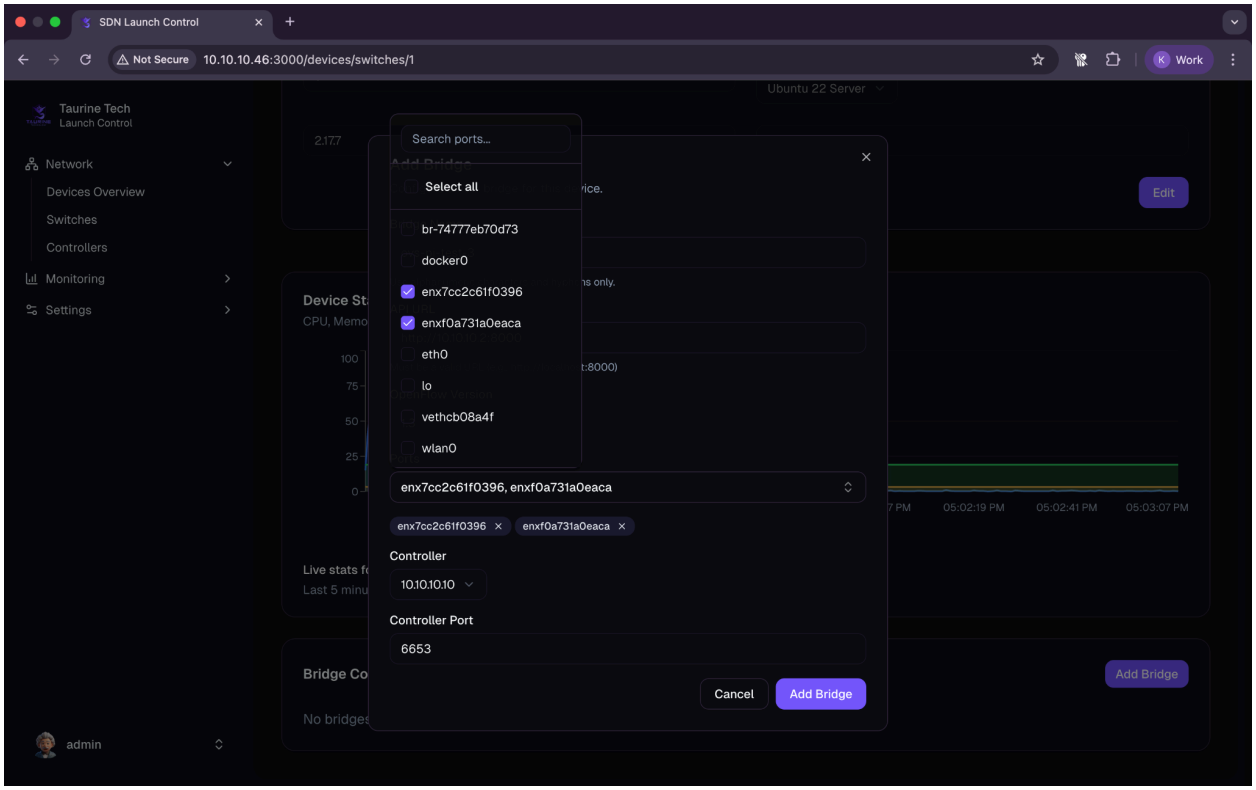

Figure 6.2: Example Port Names

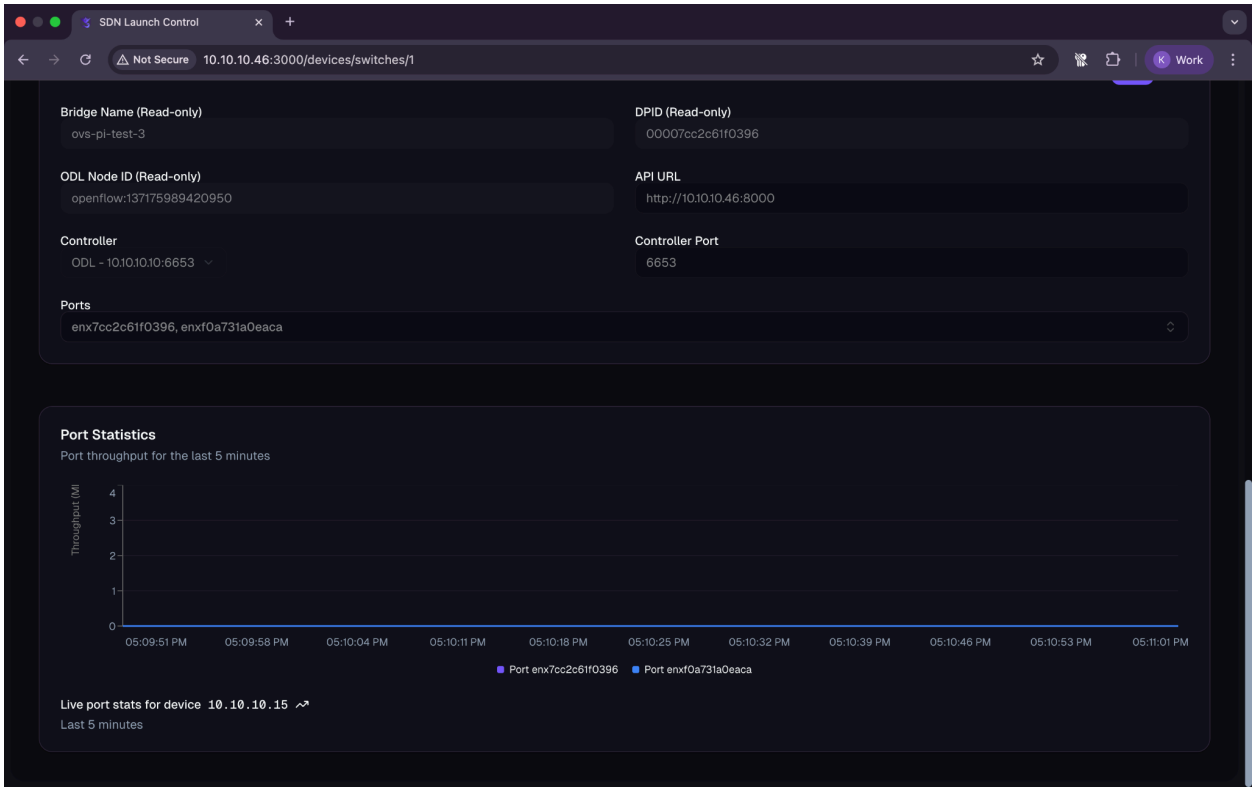

Figure 6.3: Port Statistics

7. Preliminary Tests

You will now be able to connect to your AP and access your network as you would normally. To the end user there is no difference but you are now using the SDN paradigm to route traffic. Navigate to the dashboard and you will see a network map as seen in Figure 7.1.

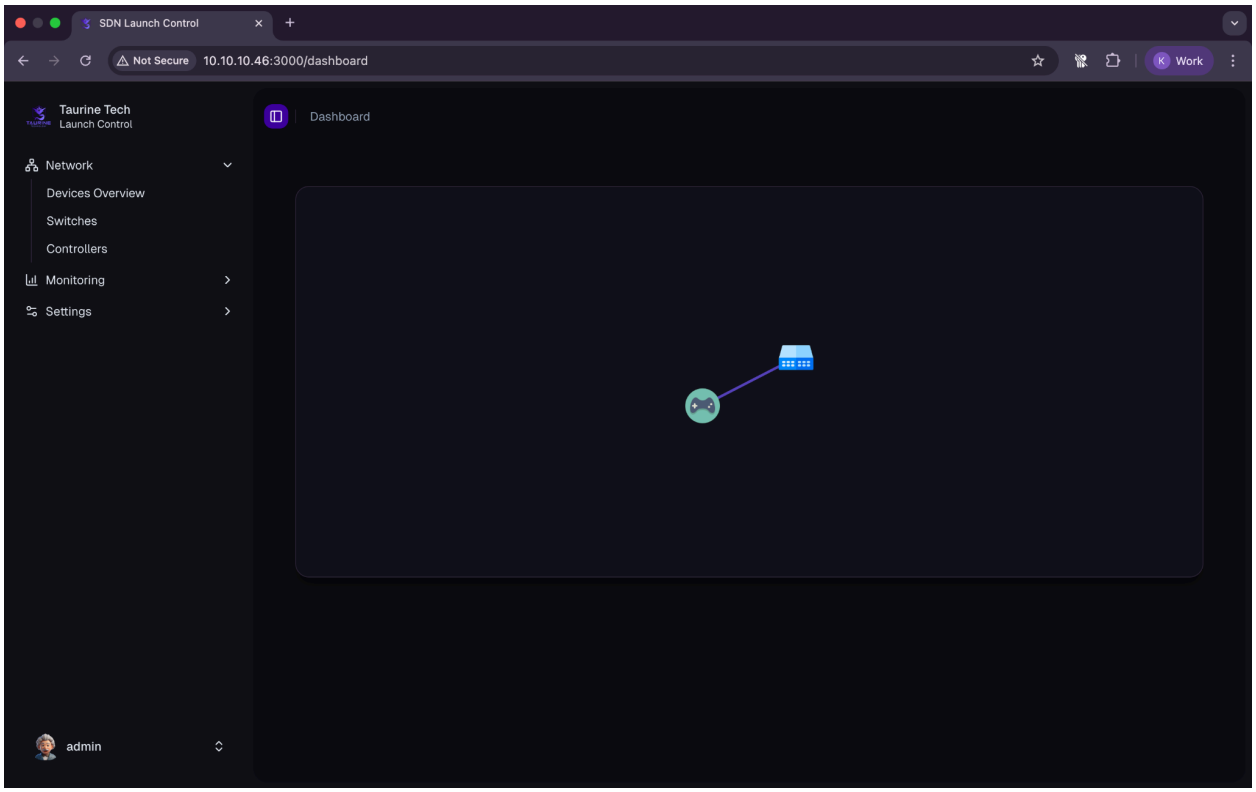

Figure 7.1: Network Map

Advanced Features and Set Up Guide

This section focuses on advanced features such as AI-based traffic classification and notification systems. We will build on the network we set up before so ensure you have set up the network as explained above.

Prerequisites

- An Opendaylight SDN Controller.
- A running instance of Launch Control.
- A switch running OVS with a bridge connecting it to your SDN Controller and network.

8. AI Traffic Classification

Navigate to the ‘Plugins’ page from the side bar Settings > Plugins. You will see the two options displayed in *Figure 8.1*. Click the install button for the ‘Opendaylight *Traffic Classifier Plugin*’ first and accept the prompt to install the plugin. Once this is installed you will now be able to view the ‘AI Services’ tab in the sidebar menu. Before accessing this install the ‘*Traffic Classification Sniffer*’ plugin on the switch you set up in the previous section.

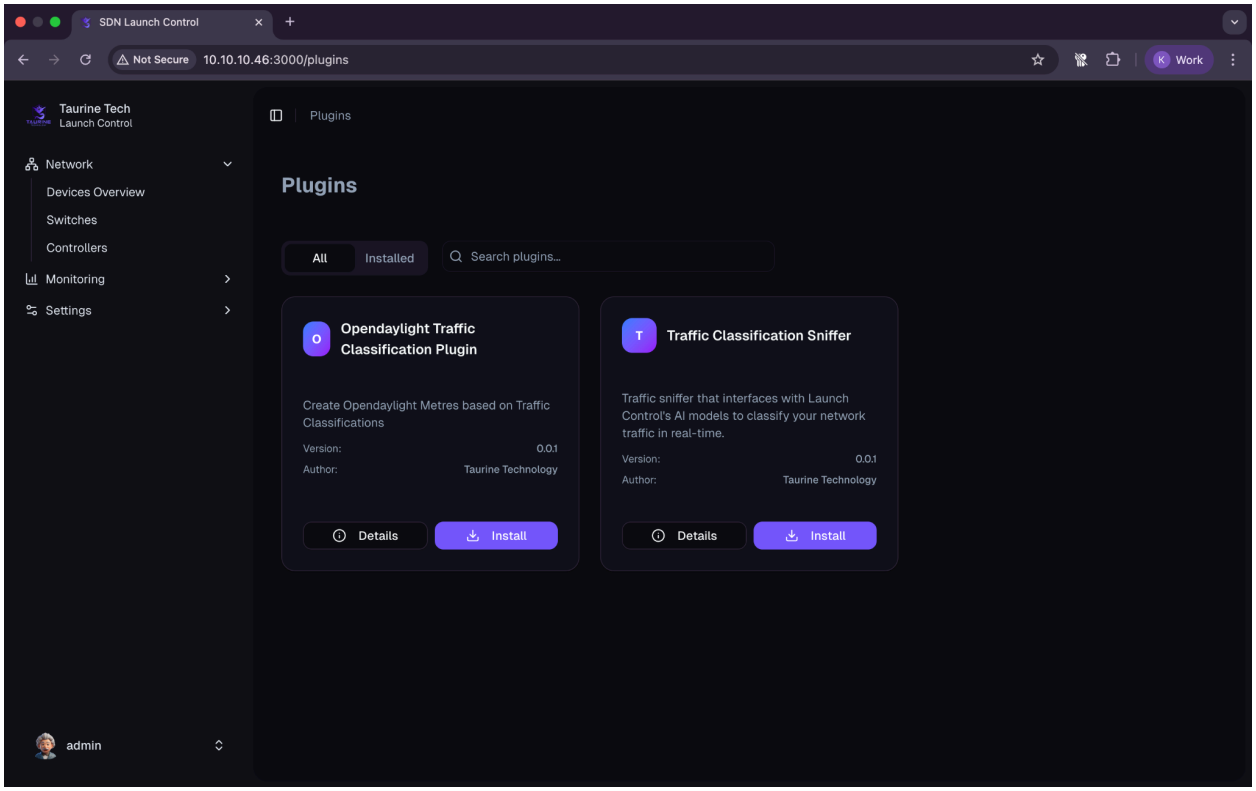

Figure 8.1: The Plugin Page

It is important to ensure that your network set up matches the set up shown in *Figure 3.1* to ensure the example in *Figure 8.2* will work for you. In the ‘Install Sniffer’ dialogue we select our switch, insert the API URL, which is explained in the ‘Setting Up Open vSwitch on your Target Device’ section, and then we need to insert port details which are crucial for traffic classification, however, they can be complex to set up. The port to monitor is the name of the port connected to your AP, refer to Figure 3.1 to see the overall network diagram and refer to Figure 3.2 to see the Raspberry Pi’s ports. The port to monitor, the monitoring interface name in *Figure 8.2*, is the bottom port so this will have the higher integer value for its port number. Therefore, the port to the client is port 2 (or the higher port number if you don't have ports number 1 and 2) and the port to WAN is port 1 (or the lower port number if you don't have ports number 1 and 2) in our example below.

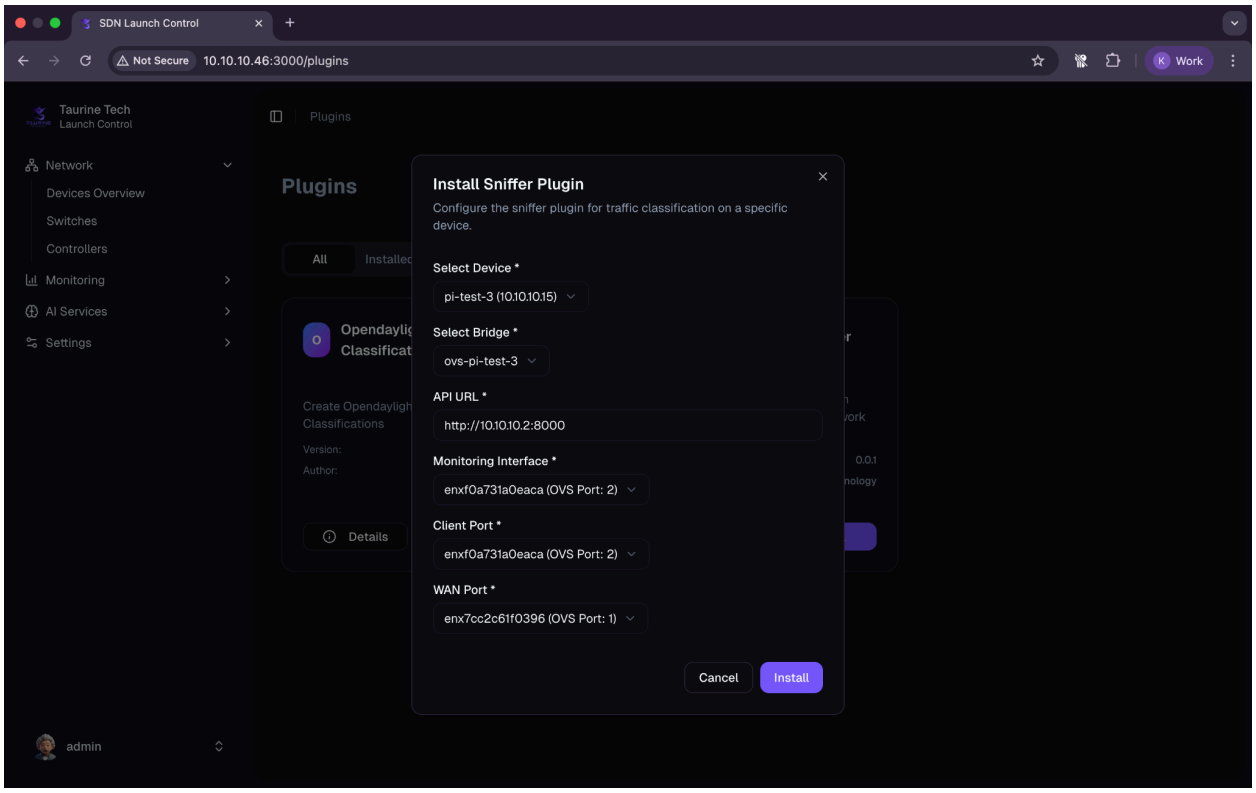

Figure 8.2: The Install Sniffer Dialogue

You can see these details applied to the set up from *Figure 3.1* and *Figure 3.2* in *Figure 8.2*. Once you have installed the sniffer you should see it displayed on the page under the ‘Installed’ tab as seen in *Figure 8.3*.

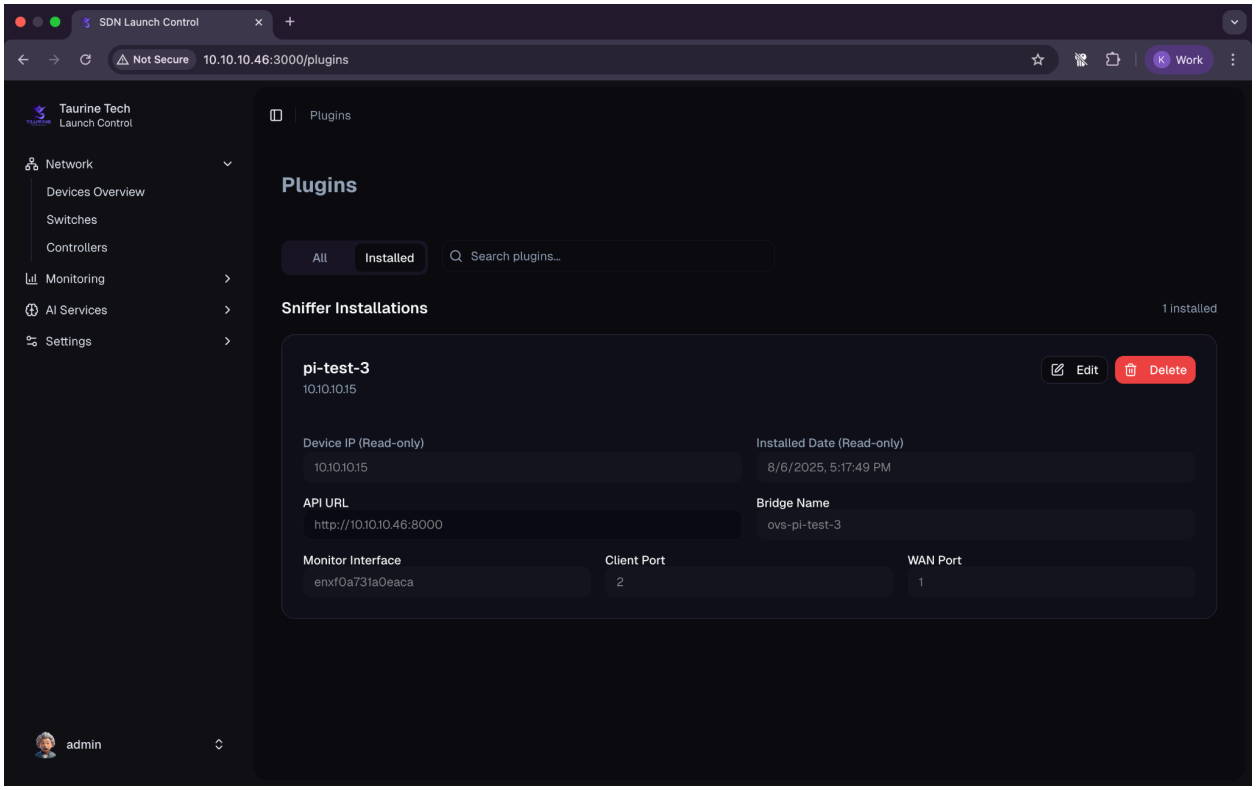

Figure 8.3: Sniffer Installation List

Once this is complete you will be able to navigate to the Classification Graphs (Monitoring > Traffic Classification) and see real-time traffic classifications when you have devices connected to your AP, as seen in *Figure 8.4*.

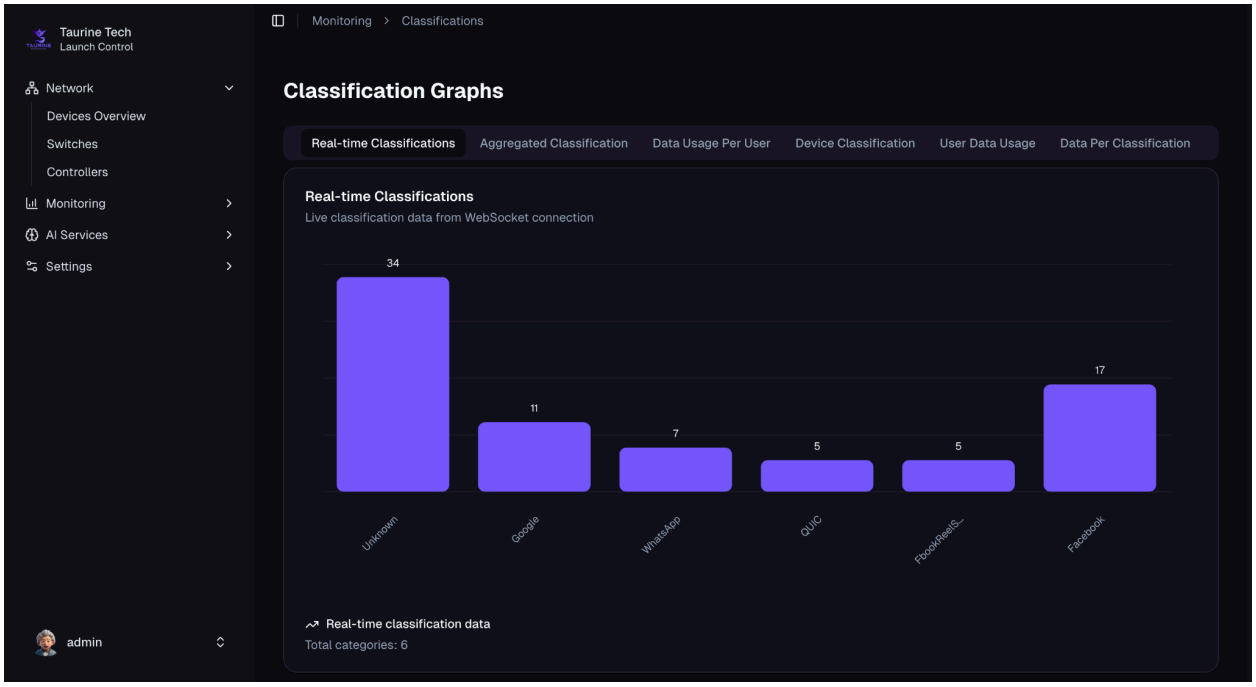

Figure 8.4: Real Time Traffic Classification Graph

9. Setting Up Traffic Shaping with AI and Opendaylight

Now that you have set up the required plugins you can navigate to the Traffic Classification Rules page using the sidebar: AI Services > Traffic Classification Plugin.

On this page you will see the meter entries for your switch and controller combination, see *Figure 9.1*. This should be empty at this point in the process.

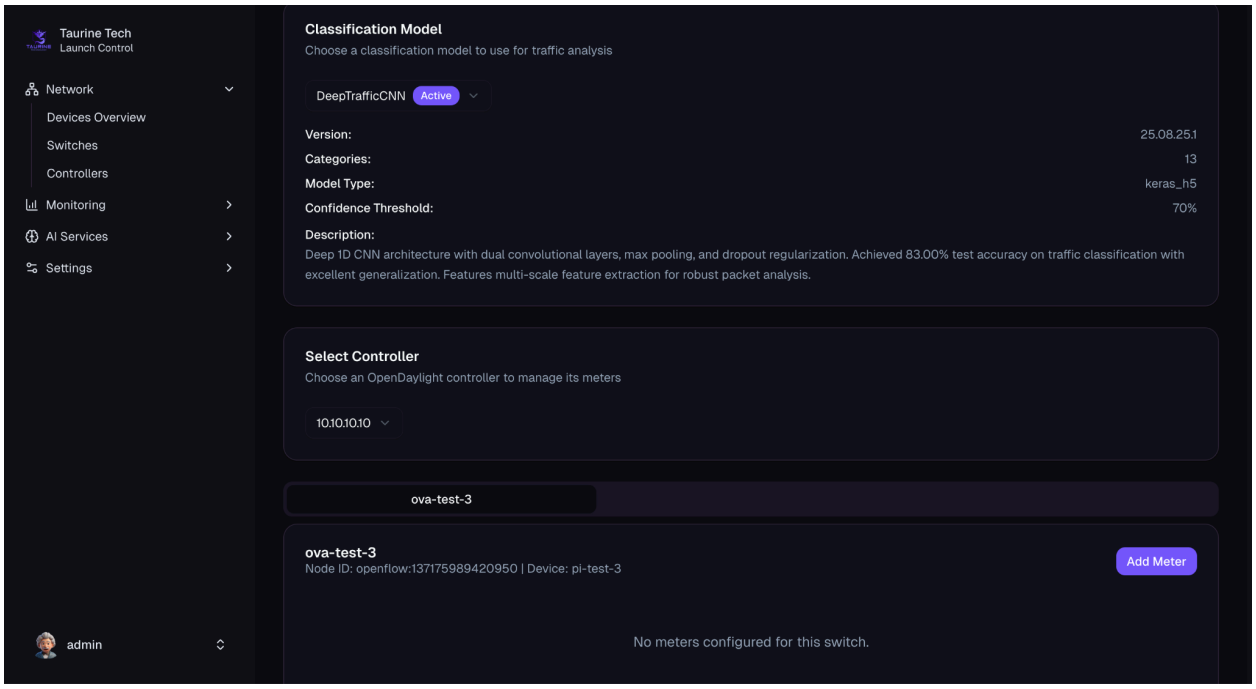

Figure 9.1: Meter Entries Page

Click on the ‘Add Meter’ button. This will open the meter creation dialogue displayed in ***Figure 9.2***. Here you can set up rate limiters for different applications. Note you can assign more than one traffic class per meter. You can also assign a device to the meter meaning that particular user will have a rule that applies to them specifically. There are also time options to add more granularity to your rules.

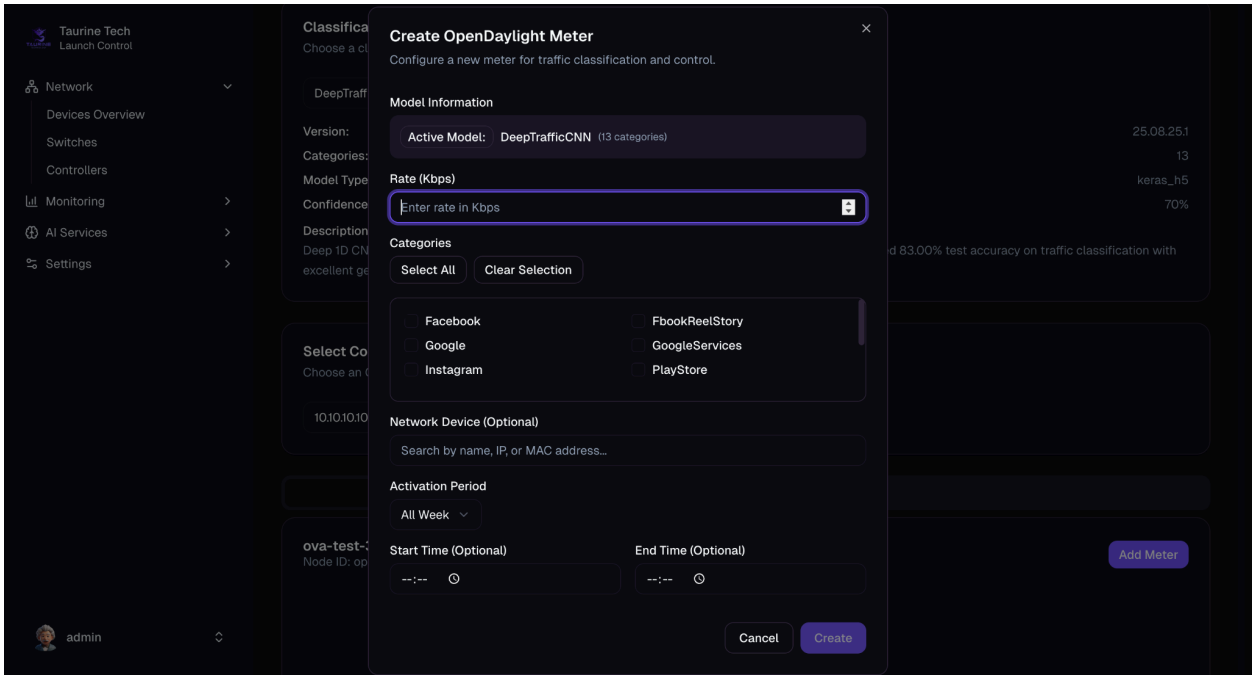

Figure 9.2: Meter Creation Dialogue

Every meter entry you create will allow you to monitor the amount of data consumed by that class of traffic and monitor the data consumed by users. Any bandwidth you see consumed by ‘Unallocated Traffic’ in the monitoring dashboard is traffic that does not have a meter to manage it. For this reason it is **recommended you create a meter for every traffic category** even if you don’t want to limit the traffic. You can set the rate to an arbitrarily high value if you don’t want the meter to influence this traffic but you still wish to monitor it. Once you have successfully created a meter they will appear on the page as seen in figure 9.3:

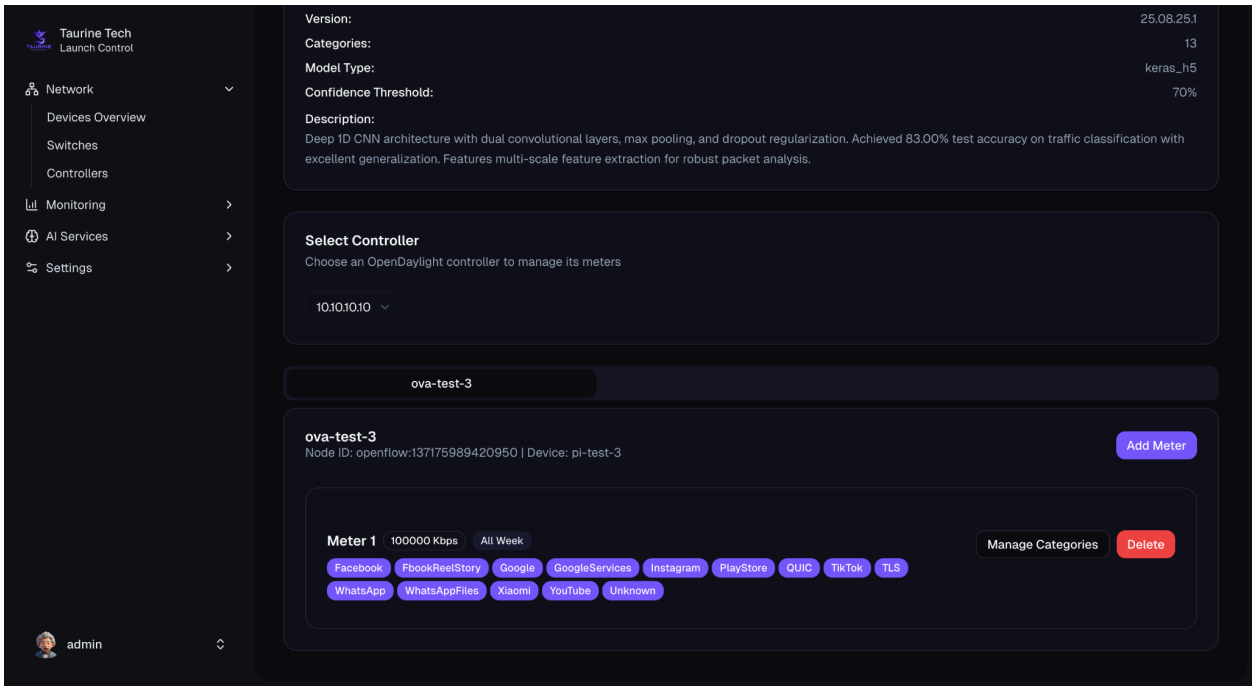

Figure 9.3: Successfully Created Meter

10. Monitoring

Once your switch and Launch Control have been configured to classify and monitor traffic you will be able to access user classification and data consumption data. Navigate to the traffic classifications monitoring page using the sidebar menu: Monitoring > Traffic Classification. Here you will see 5 different tabs that can be used to monitor traffic and bandwidth consumption, see *Figure 10.1*.

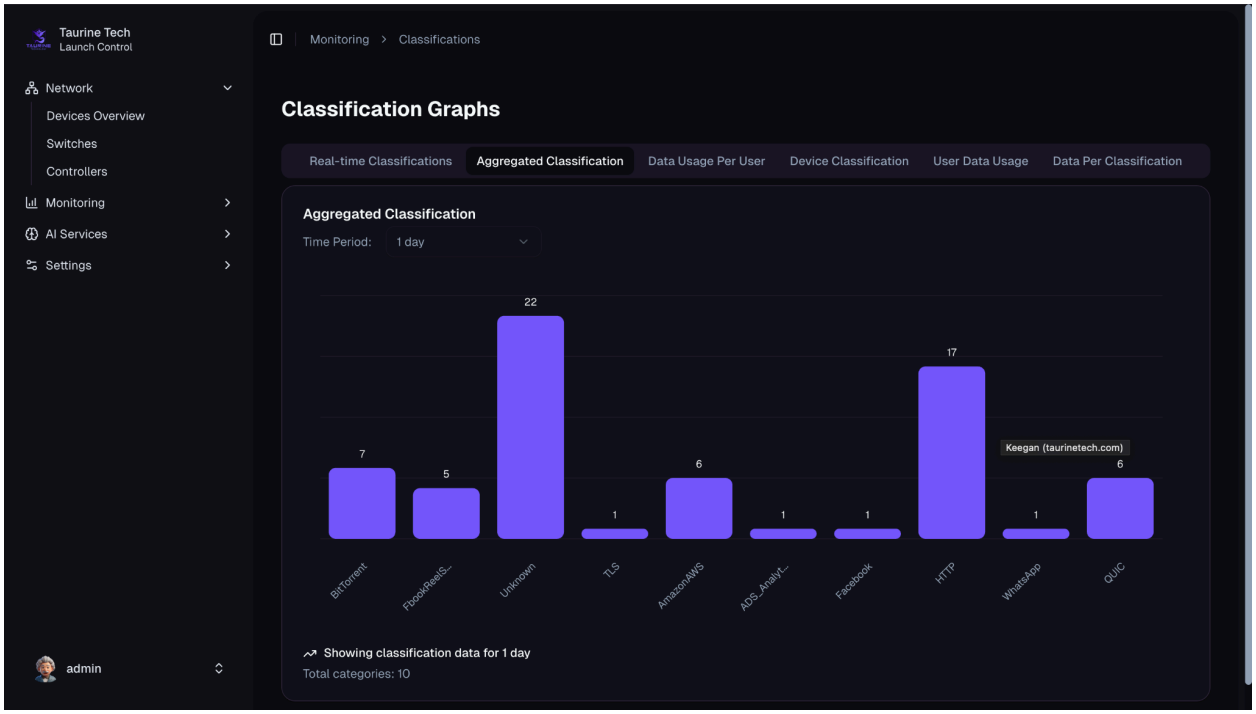

Figure 10.1: Traffic Classification and Consumption Dashboard

11. Notifications

Users can link their Telegram account to their Launch Control account to enable notifications about bandwidth consumption and network summaries. Navigate to your profile page by clicking on the profile icon in the bottom right corner of the page and selecting the ‘Settings’ option in the menu. Once you are on this page select the ‘Telegram’ tab as seen in *Figure 11.1*. Here you can link your Telegram account by clicking the ‘Link Telegram Account’ button.

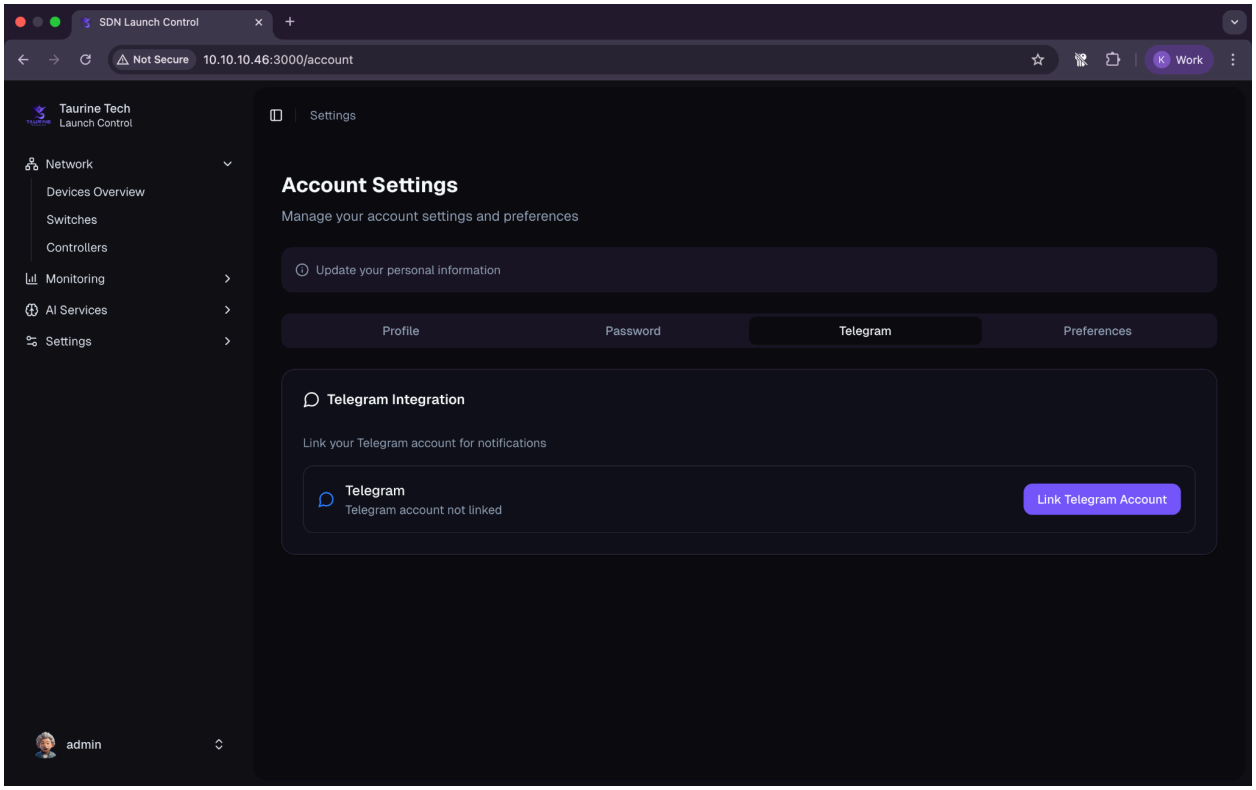

Figure 11.1: Profile Page

This will bring up a dialogue that displays a QR code that is only valid for 45 seconds, see *Figure 11.2*. Keep this dialogue open during this process.

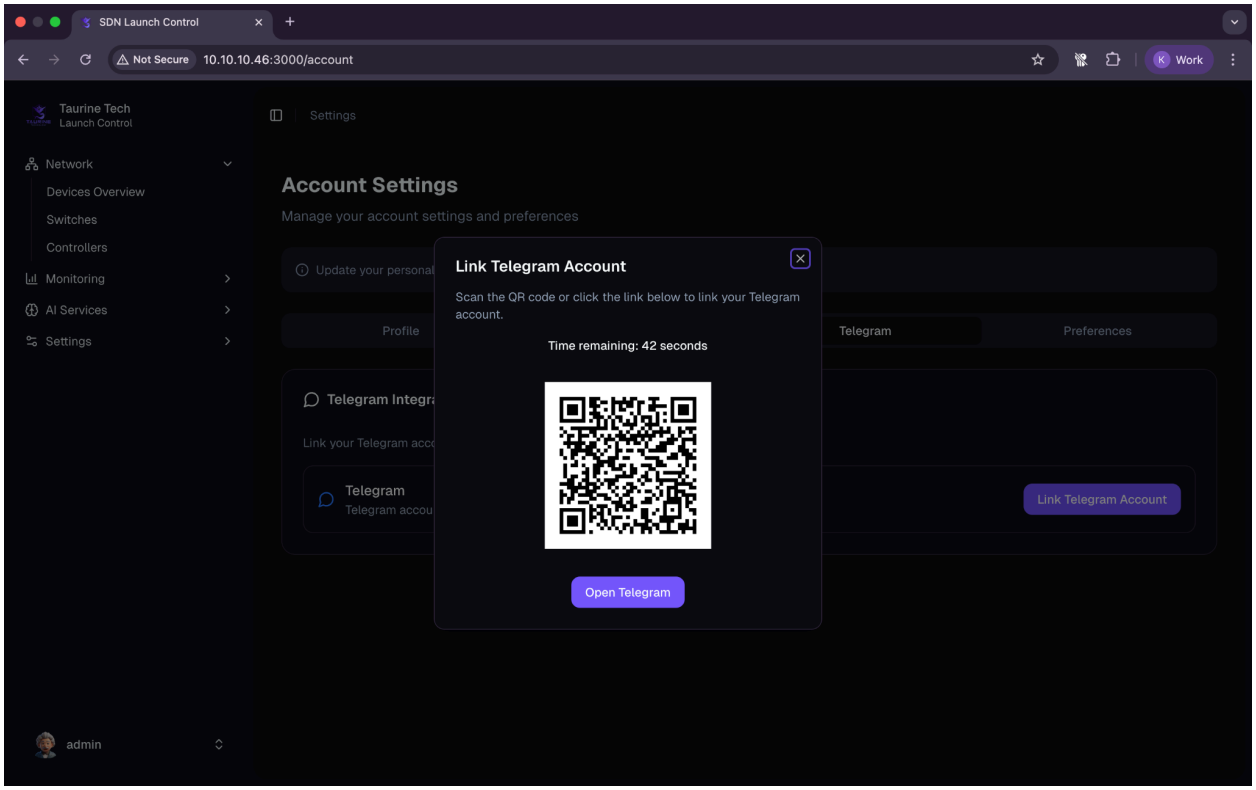

Figure 11.2: Telegram Linking Dialogue

Ensure you have installed Telegram on your phone, have set up an account and then scan this code with your phone's camera. This will automatically open Telegram and prompt you to start a chat with the Launch Control bot. Start this chat and you should receive a message from the bot confirming your account was linked, see *Figure 11.3*.

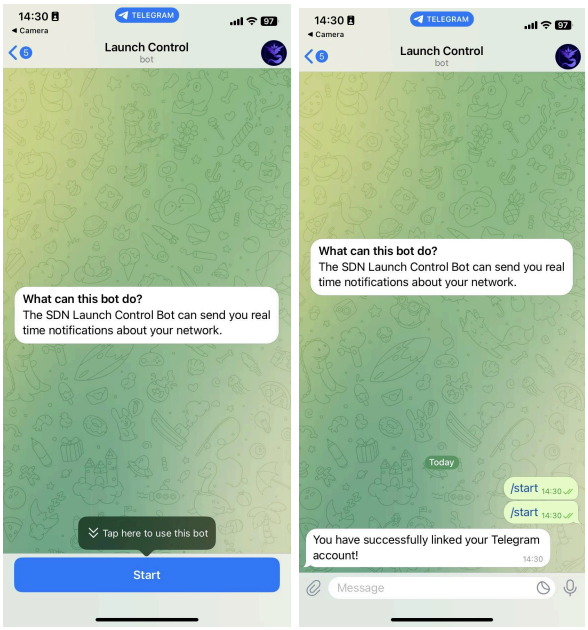

Figure 11.3: Telegram Notifications

Once you have linked your account navigate to the notifications page from the sidebar menu: System > Notifications. Here, see *Figure 11.4*, you can set up notifications based on network activity or receive general network summary notifications.

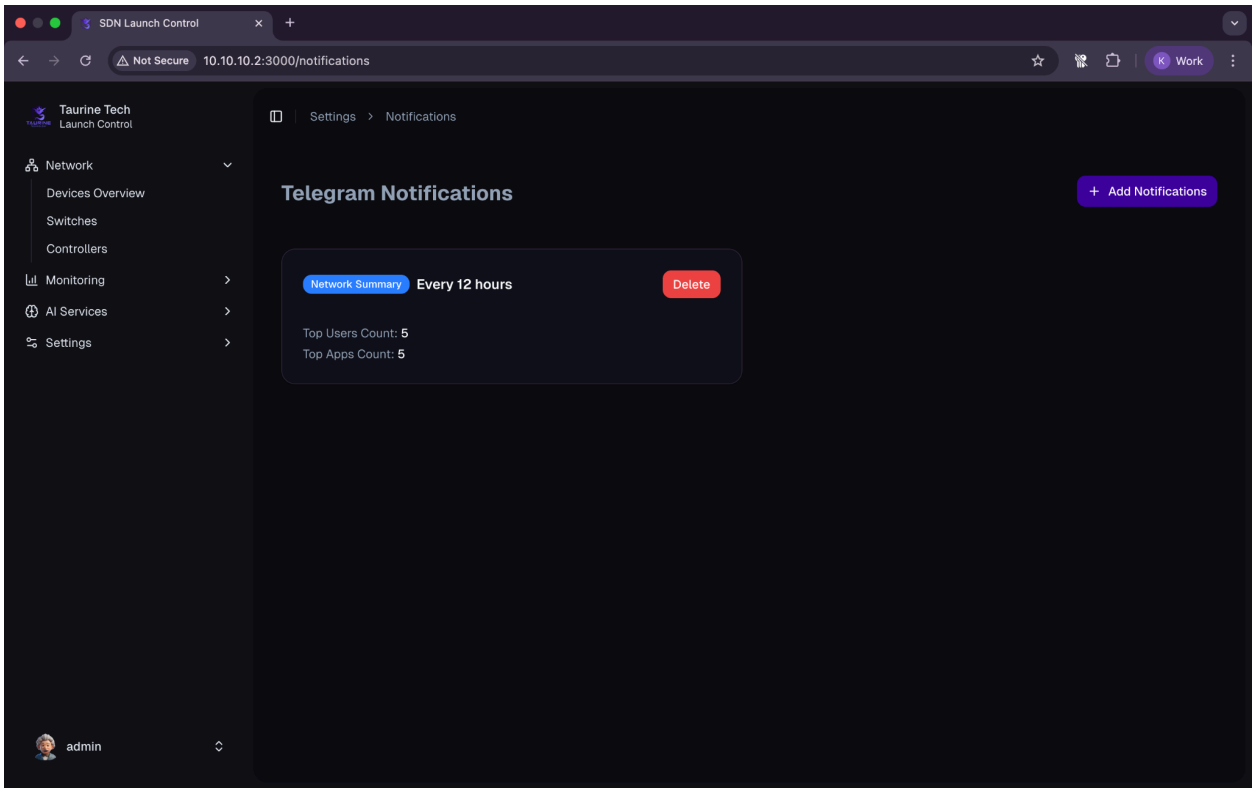

Figure 11.4: Notifications Page