PLEASE HAVE IN MIND THAT THIS FILE WILL BE UPDATED.
THIS IS JUST QUICK RECAP WHAT THE APP DOES AND HOW IT IS WORKING FOR NOW.

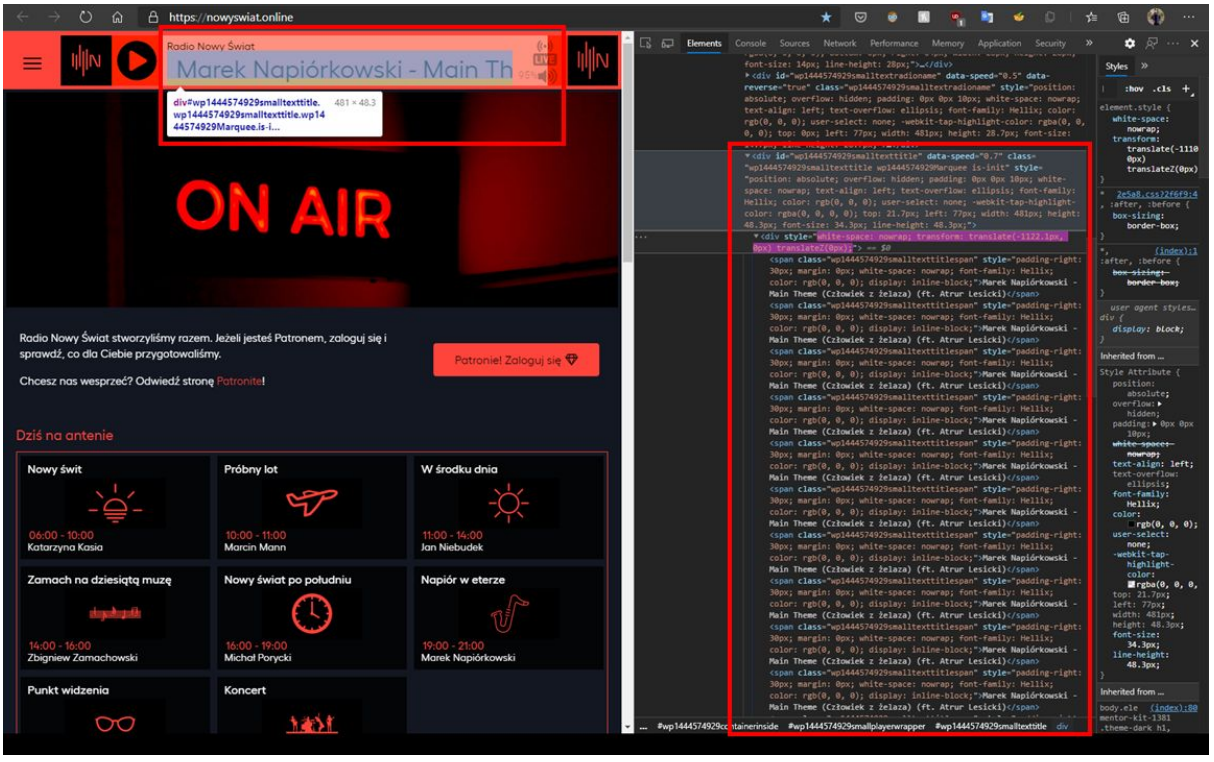
PART I - scrap songs

POC

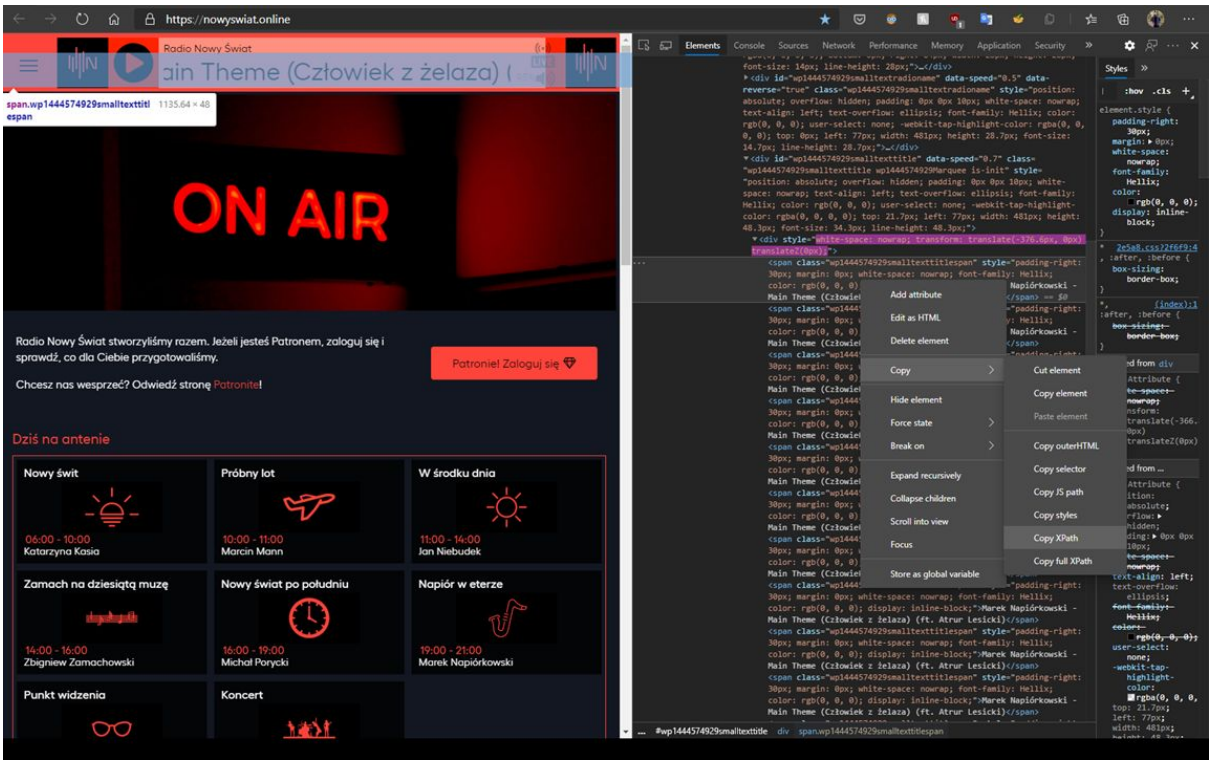
As a POC, I created a Python script that constantly runs on the machine and saves songs to the file.
Having investigated the website (as there is any API), I thought about 3 ways to find titles of songs and make the POC done:

1. scrap main page using requests,
2. scrap main page using selenium,
3. get a title found on subpage (.txt file hosted on the server).

Requests
I used XPath, so I could find a title and scrap it directly.
However, it turned out that the page does not return the value of XPath. This could be caused because data is run by JavaScript (song's title is not in the website html). This is because (as I think) if a title is long and it cannot be displayed on the screen, the text flows from left to right (like text on TV news at the bottom of screen).



Copy XPath



and try to request and get the value from XPath.

```python
import requests
from lxml import html

url = 'http://nowyswiat.online'
page = requests.get(url)

tree = html.fromstring(page.content)
song = tree.xpath('//*[@id="wp1444574929smalltexttitle"]/div/span[1]/text()')

print(song)
```

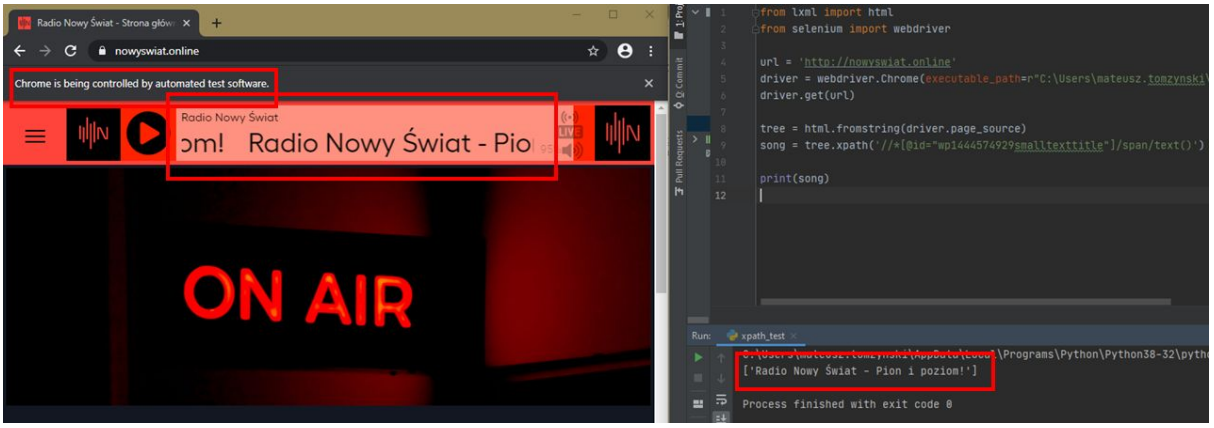
```
xpath_test
C:\Users\mateusz.tomzynski\AppData\Local\Programs\Python\Python38-32\python.exe "C
[]

Process finished with exit code 0
```

As a result, it returns an empty list.

Selenium
I did the same thing for selenium. The code for selenium opened a web browser (I used Google Chrome driver) and got value from XPath.

In the result, the current value was returned as an element of a list.



This solution works but I was not sure whether using selenium is the most convenient way to find a song's title.

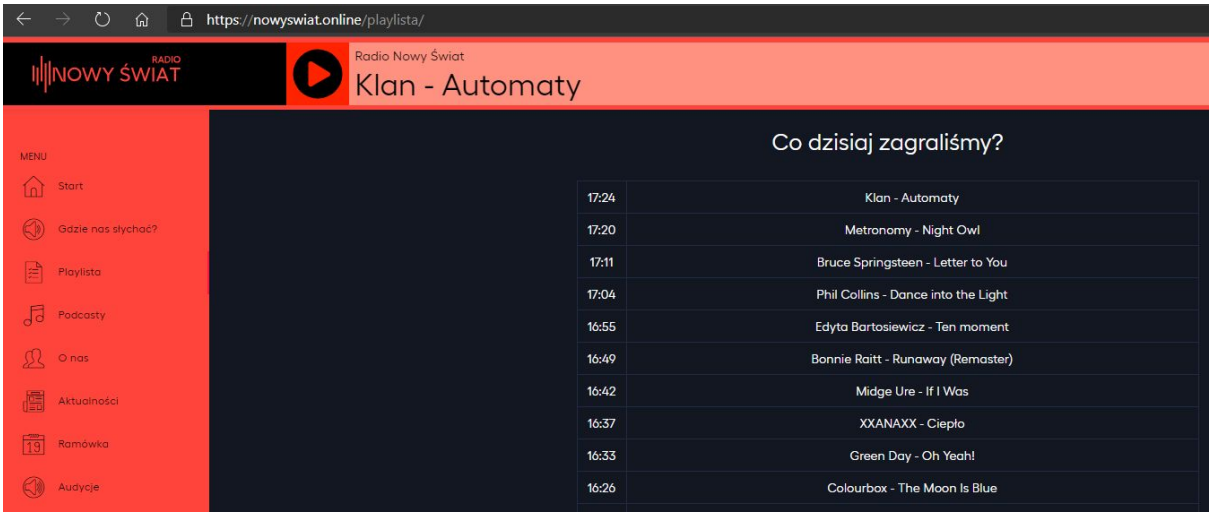
Read file being hosted on subpage
Finally, I started digging into the page and found the subpage which in fact is the text file with currently playing song or - if no music is playing, the name of the radio.

This was my first solution which worked!

However… one day I loaded the page and saw that there is a subpage “Playlist”. Then basically I could forget about my previous challenges.

Ready to take playlist
As I saw the new subpage “Playlist”, I thought that scrapping songs’ title will be as easy as I could not even imagine.

The playlist for the current day.

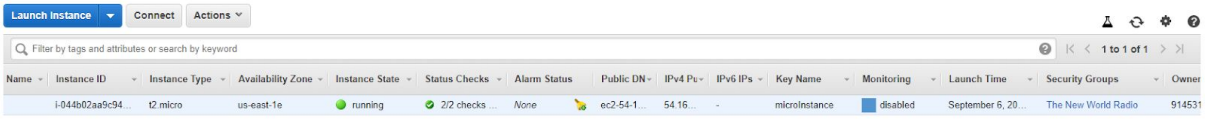


The only thing I had to do was to read songs and save them.

Run the script

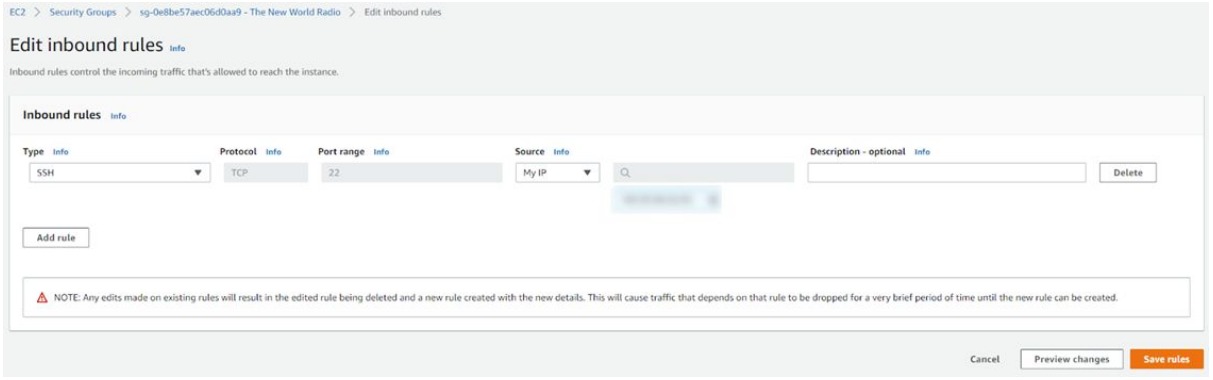
This is how I run the script of POC for my first working solution.
However, this will not work like described below! As a final solution I would like to use AWS Lambda

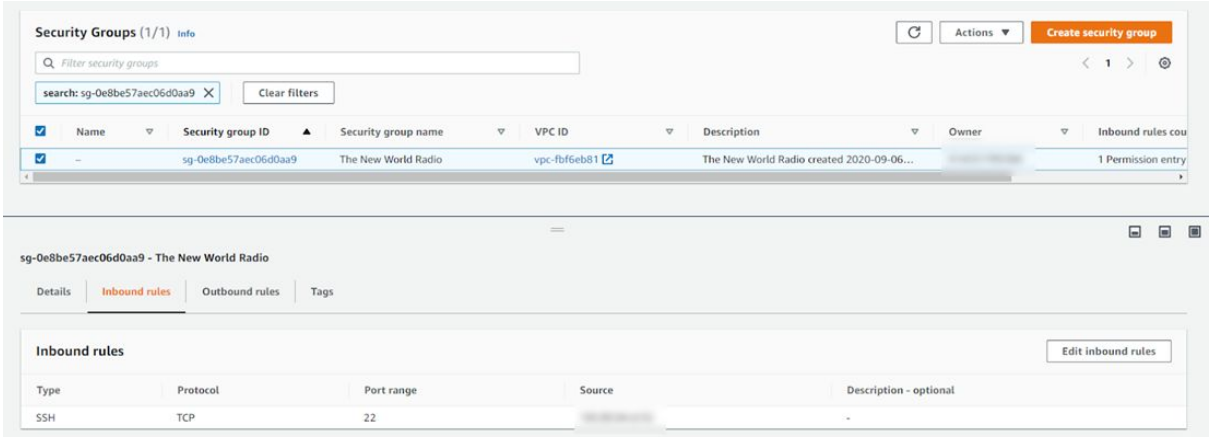
1. Script is running on AWS E2C micro machine



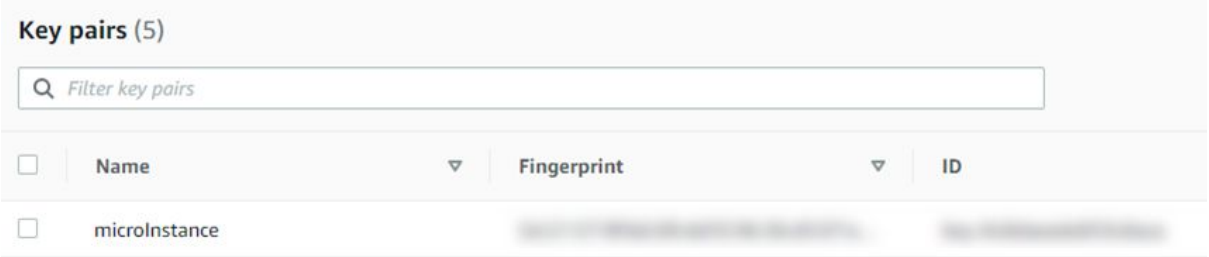
2. Security Group

The Security Group for the instance was modified. Inbound Rule was set up to allow connecting to SSH only using “My IP”, so this prevents connection to my machine from other networks.

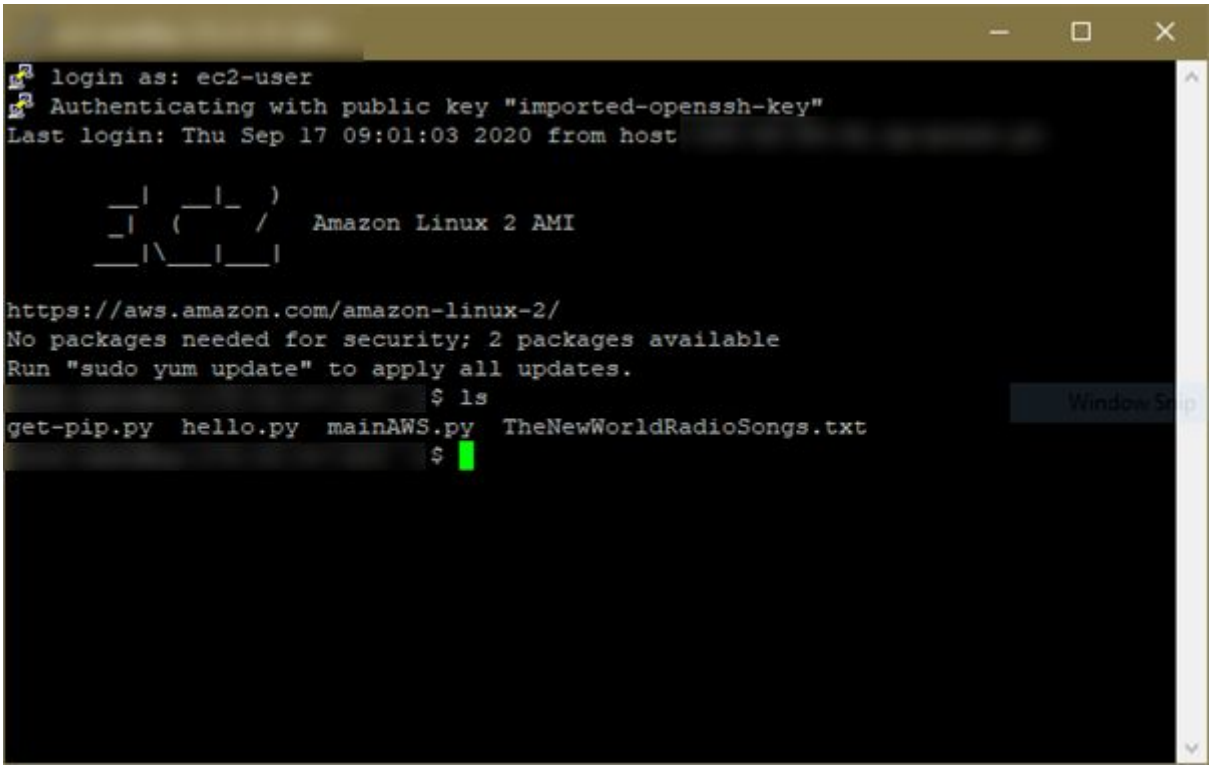




3. Create Key Pair for SSH connection

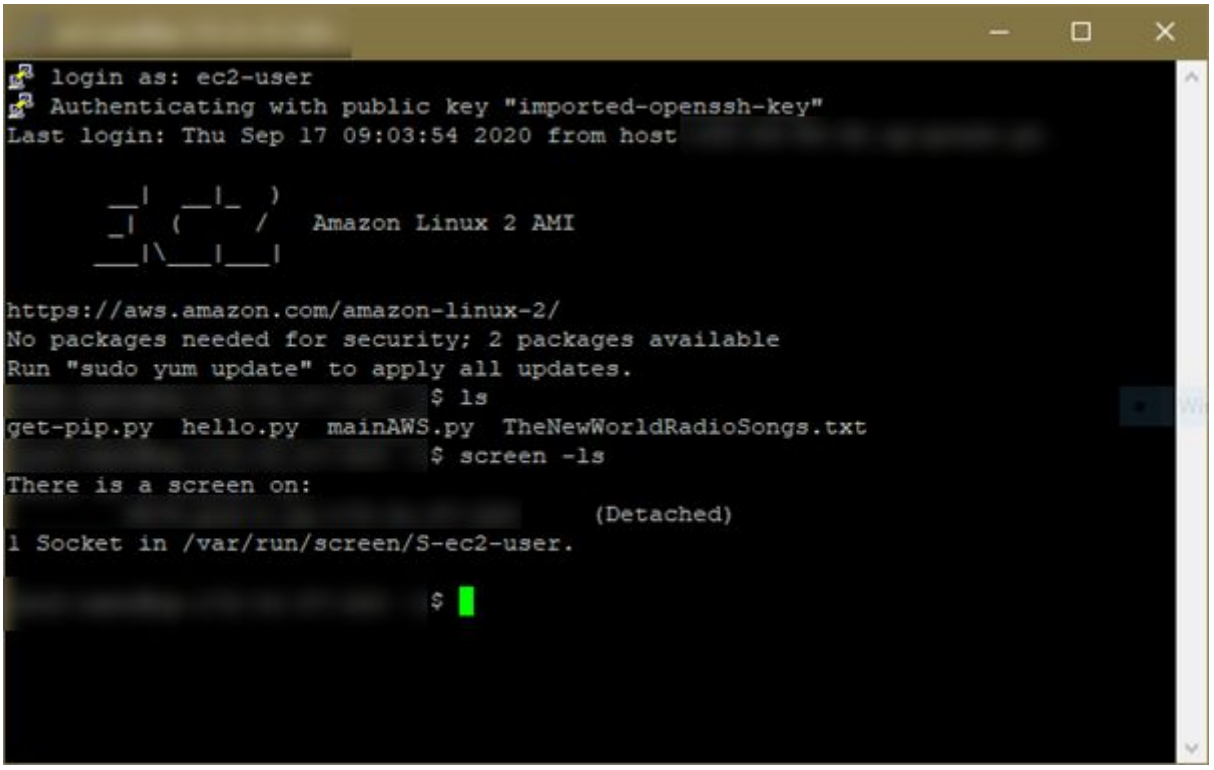


4. Connect to SSH via Putty



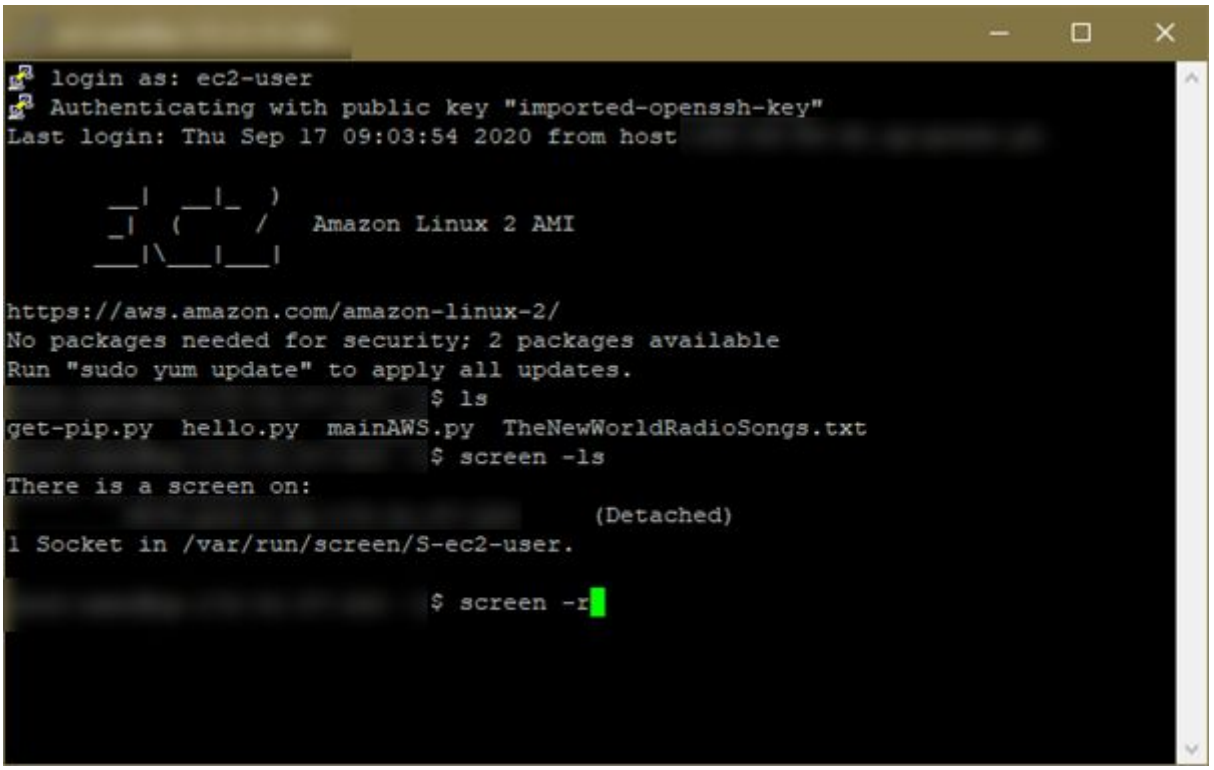
5. See screen command

```
screen -ls
```



6. Script is running and outputting results (songs)

I have only one screen, so I can get it using
`screen -r`



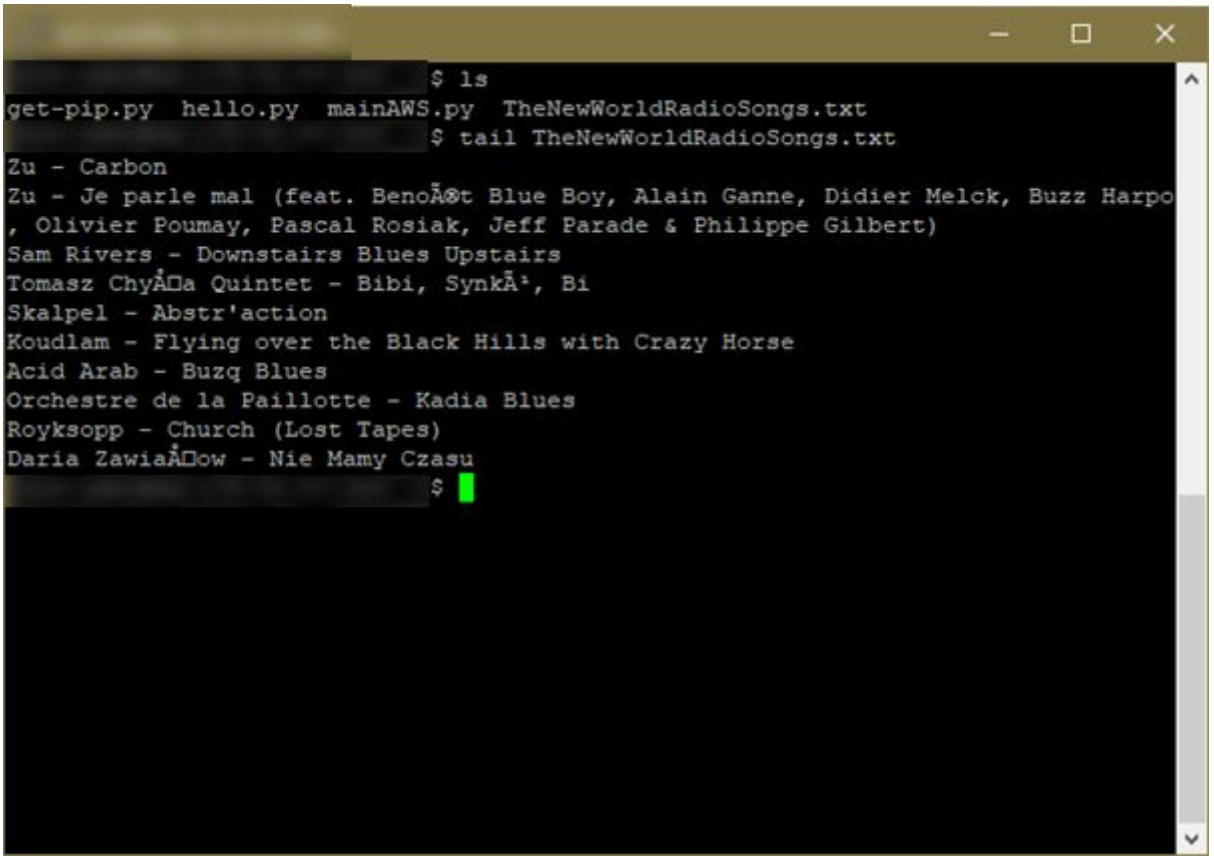
As you can see, the app prints songs to the console

7. Songs are saved in a text file

Using

```
tail TheNewWorldRadioSongs.txt
```

we can see the latest saved songs and compare them to those printed on screen terminal.



As you can see there is still an issue with encoding. The issue appears already on the website, so I will need to find the workaround for that.

PART II - Web Spotify API

Description to be updated