Unknown File writeup

* Possibly helpful commands: file, base64, tr, tar, gzip, bzip2, xxd *
* You will only need these commands to solve the challenge *

Essentially, all you need to do is find the picture, so let's start by looking at the file we were given.

First, we should figure out what file type it is.

```
kuisc@kuisc:~/Documents$ file unknown
unknown: ASCII text
kuisc@kuisc:~/Documents$
```

FILE - shows the type of file

Ascii text doesn't really help us, so let's look at the content of the file.

```
kuisc@kuisc:~/Documents$ cat unknown | head -n15
1f8b0808a8a9895a0003756e6b6e6f776e00947dc77aa36cd8de9e536121
ba600902d17b67070810bd8a76f4793d5f923fdbd897c7635b9279cadd54
f083691e04cbda6e9b284ec572accd82afc067967f6007c7b3ac2f70e083
3dfc37f82c72952fbe584279b1b9f4e2049db747f3cd09c19bd31d4838f5
50e4e8ecfdcd8bf7a97be2e5e752cd7ee54b6fdeced8ab6e5cab0e5dcb97
f6df8ddb31b8c10adc10abbc1cb67e7ff550aeff7b87fecf7ffeffdfb5c7
51b1ce8b556296b079d6115888fd822a4e817bd0362bfd2bd107a582effd
3f6f1c2bbc409922cbe6f2ab7c082cb7829febff5dcefe9fcb41ff2e5be9
0267c7d2831959feef42ac6cb3e7df8f5fe4f331d41c26bcf737b62279cf
486cd5c655bfe9098b06815822f76ce824bbeb9b0643061ff8d5e493efac
35c5393efc5b7d8df3d1648ab722cf345b63440b3c9e6c228dd3f3f1f522
eb951092972b1eb6cc365ce52998f0c382704e5248aac6dc98e5d3617534
e46cfdfebc25777efaec173b4ffe1773ce5754753f13e5f83ca6af50af2f
0fb9fd5ab0a4db4f39cad7e4aa7ee9f211bf78283d973c26bee7981aa3e1
75f69cb2e4f062d242219e83fa9cb355c92be47db49ced4c62cbb1ef745e
kuisc@kuisc:~/Documents$
```

CAT - outputs the contents of the file to the terminal
HEAD - only outputs the first 15 lines
I did this for simplicity, neither of these things are required to solve the challenge.

Looking at the data, it seems to be hex encoded. Let's try to decode it from hex.

```
kuisc@kuisc:~/Documents$ xxd -r -p unknown > non_hex
```

XXD - this command deals with hex encoding and decoding
-r - this reverses (decodes) the hex process
-p - this outputs the file ini postscript form
> non_hex - stores the output in the file "non_hex"

This command will take it out of hex form, into something else and store it in a file. Now we need to know what type this new file is.

```
kuisc@kuisc:~/Documents$ file non_hex
non_hex: gzip compressed data, was "unknown", last modified: Sun
```

It looks like this is now a gzip file, so let's unzip it.

```
kuisc@kuisc:~/Documents$ gzip -d non_hex
gzip: non_hex: unknown suffix -- ignored
```

GZIP - a type of zip file, this command deals with compression of data to this file type
-d - stands for decompress
* can also use the GUNZIP command, this functions the same as "gzip -d" *

So, we have an error here. Gzip doesn't recognize the suffix (file type) of non_hex. Let's change the file extension and try decompression it again.

```
kuisc@kuisc:~/Documents/unknown_test$ mv non_hex non_hex.gz
kuisc@kuisc:~/Documents/unknown_test$ ls
non_hex.gz  unknown
kuisc@kuisc:~/Documents/unknown_test$ gzip -d non_hex.gz
kuisc@kuisc:~/Documents/unknown_test$ ls
non_hex  unknown
```

MV - moves the file from one place to another, in this case it is used to rename the file

It worked! It looks like it created a new file out of it. Let's check the type again.

```
kuisc@kuisc:~/Documents/unknown_test$ file non_hex
non_hex: ASCII text
```

Ugh, Ascii again, let's take a look inside.

```
kuisc@kuisc:~/Documents/unknown_test$ cat non_hex | head -n15
/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAUEBAUEAwUFBAUGBgUGCA4JCAcHCBEMDQoOFBEVFBMR
ExMWGB8bFhceFxMTGyUcHiAhIyMjFRomKSYiKR8iIyL/2wBDAQYGBggHCBAJCRAiFhMWIiIiIiIi
IiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIiIL/wgARCAJYA4QDAREA
AhEBAxEB/8QAHQAAAQUBAQEBAAAAAAAAAAAABAECAwUGAAcICf/EABsBAAMBAQEBAQAAAAAAAAAA
AAABAgMEBQYH/9oADAMBAAIQAxAAAAC57/niB2EFvF2s0cm9HAgkYgmtMZA1VVGf0zqNM5AvMtL+
NDVUgpU5FbkOGqYwUzKCoqwjbJTs07absY0LVTD5jXLBMcoCC5is4EZCSGwQIAjBgTJ2Eu2VWqZa
HgocNqIxRAM1WBQMzdFHSq7UAh2xxDuYBRhGKMUbGIYxwphEisCT0zUiEPHzUaB6ULIgiCMIwYCD
axrcY4hxoaNoNTlQqaA5nC9aeJ47nK7qbsJcg0FwkBQRpGkBAFaqrioqJAuos9W8T25FTk1RIh80
8oRlC5pKkRk6LSbtp1tI0LVSDY1DUQuWNIJB81w1alQoAVNc0OEQOTLTtpdqmWh4OHyEBjGgxEQx
2qgMzRmmUuiDJgagQ0I2mCgaFAQUDShIKVSXSsEj2FyTIehRRsYETImMYwTk5UEJkpyjRg4Asr2A
jGCIbgRNocz154W01fZ3czc6fA0XCUOZwuBBqJrQtIBy8ZqbgcDinJuB8kiolUSqUK5qkpVzTQNm
7eNbONClT2omh3A1SxiNcCAjJkWqRA69rOVIDTGSp2E1bSz0SCVN4+E0GgwI04mMAQdGzMMoqAWo
2kEgoWhxCiDJAaBogYoilJwj3JyDESIc0kjKIhtDhNRMM8dvFXM1bKjU1GO1VNUbnO2qFqrsFT5t
BND1943Od6GLs5p40EglDgVrgVDg4EY1qFpw1CNpSlB43IllmTRqsgp44xANVwCNPVHxofOhA3Aw
kapGqY2kDgQEbLStwegZrN0qmk0CU7OHZpzg4SjcHAgMQ1DBxiibjGEFIOtY0l4IArVcKtc11Ksc
```

It looks like random strings of letters, but that /9j/ at the start sticks out. A quick google search of "file /9j/" leads to results related to base64. Let's try and decode it.

```
kuisc@kuisc:~/Documents/unknown_test$ base64 -d non_hex > new_file
```

BASE64 - this command deals with everything base64
-d - stands for decode

New file created, let's check the type.

```
kuisc@kuisc:~/Documents/unknown_test$ file new_file
new_file: JPEG image data, JFIF standard 1.01, aspect r
kuisc@kuisc:~/Documents/unknown_test$
```

JPEG file? Aren't we looking for a picture? Go open it.

You win!