A Simple Report Mail

**Contents**

[**Introduction 3**](#_Toc105538912)

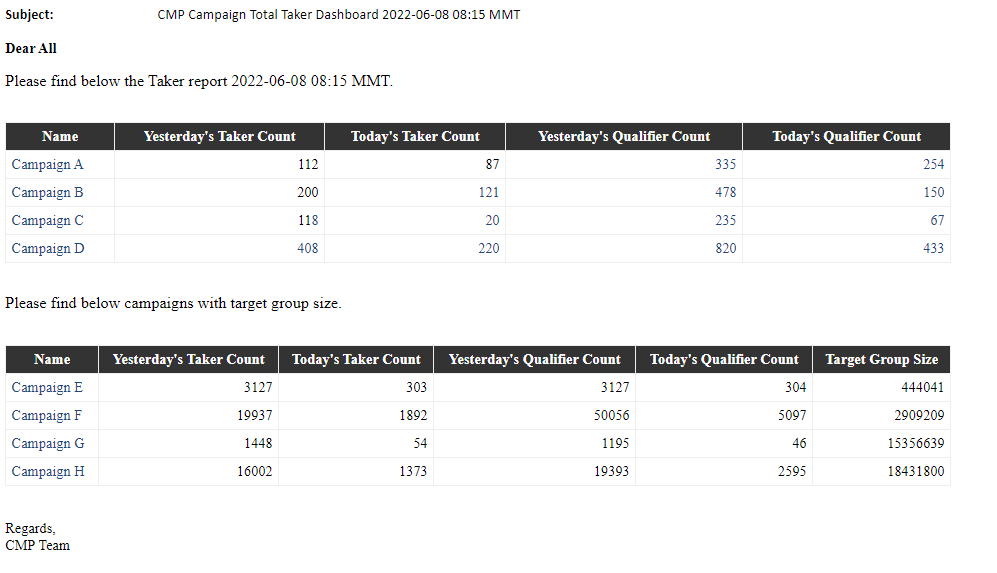
[**Process Flow 4**](#_Toc105538913)

[**Scripting 5**](#_Toc105538914)

# **Introduction**

The purpose of this document is to share experience of automation a mail report. A report which describes the count of a product purchased and the count of eligible participants for the product of each campaign by comparing two days data. The journey of data process from source dump file to smooth report on mail. It is developed by using two functional language that are Linux shell scripting and python.

Before development it is used to prepared manual by contributing numbers from each campaign by opening its report page via GUI. Which was time consumption and pain work on every four hours. Now this is almost automatic reporting on schedule time all effort need is to just add Ids of campaigns which we need to deliver.



The image represent the final result of the report in which Taker stand for a customer subscribed and Qualifier stand as a customer is eligible for the specific product.

And the target group can be expressed as how many customers are anticipated for the specific campaign.

# **Process Flow**

The journey of data from source to destination is brief in below diagram. The main script is trigger every 4 hours by scheduler then reading campaigns id, name and a flag and dump all of them from system CLI. Again the script start processing raw dump file according to its respective campaigns in which some logic present and export the result file to csv format. To make lower human intervention the csv file have to convert through html format to present on email as a report.



# **Scripting**

To complete the process two language are used those are bash script and python script. The running\_campaign.py script extracts all active running campaigns details and save it as table.txt file output.

The main script is written on bash name as ReportSample.sh and the generated output file is processed by dumping each data from backend and filtering those data and store it systematically as result.csv and Tresult.csv files.

The htmlformat.py file is used to read those result csv file and generate html file according.

Finally, mailTo python script render html file and deliver it as report format on mail.