

Contents

[Description 2](#_Toc515204831)

[How to Do It? 3](#_Toc515204832)

[Installing DNIF 3](#_Toc515204833)

[Identifying the dataset and its source 4](#_Toc515204834)

[Understanding DNIF 5](#_Toc515204835)

[Capturing the Dataset 6](#_Toc515204836)

[Uploading the data 8](#_Toc515204837)

[Creating Package 12](#_Toc515204838)

[Creating Widget(s) 14](#_Toc515204839)

[Creating Dashboard 16](#_Toc515204840)

[Configuring SMTP for DNIF 19](#_Toc515204841)

[Creating Alert(s) 19](#_Toc515204842)

# Description

**Process 1: Static Data in DNIF**

The process is divided into 4 stages of small description. This will help us further to understand how we are solving problems in each stage or what is necessary for that particular stage to be completed and pass on to the next stage.

*Stage 1:*

* Select & understand data-set from a domain of interest.
* This is the initial step, which of-course sets the entire purpose of what kind of data you are looking to analyze and make visuals on. It is an important step as your quality of work depends entirely on this. (There is no limitation to the number of data sources, it will only be limited by imagination and lack of skills in further stages – meaning multiple datasets could be merged in to one single source using logic and code & then uploaded on DNIF)

*Stage 2:*

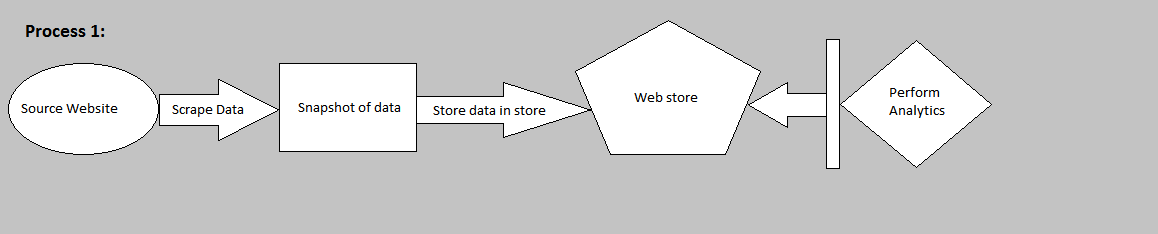
* Understand DNIF platform and its capabilities limited to project scope.
* This step was introduced so that, one is beforehand ready and understands what the platform is, how it works, what are its capabilities, its features, the query language, deployment models, limitations, etc.

*Stage 3:*

* Get the static data-set inside DNIF platform
* This stage is where the real game begins. Look out for the format of the file one is uploading, how it is being uploaded, and other requisites.

*Stage 4:*

* Perform analytics, create dashboards, & build alerts.
* Once your data is inside DNIF by uploading through the event store, all the skills and understanding acquired in stage 2 could be put into action.



*Fig 1: A Squeezed Perspective*

# How to Do It?

Let us go in steps in order to achieve the outcome desired. These steps are descriptive in details and do not directly highlight which stage they accord to. Let us dive in and perform these sequentially.

**STEP 1:**

## Installing DNIF

* To do this, first “Sign-Up” with DNIF and read the pre-requisites guide here:

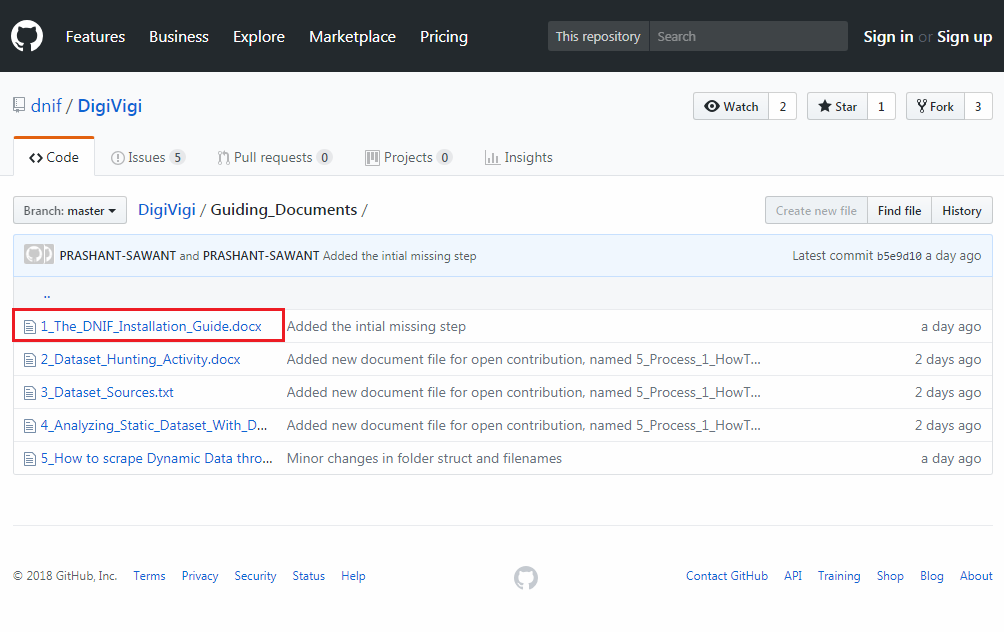
Sign-up: <https://dnif.it/signup.html>

Pre-requisites: <https://dnif.it/docs/guides/getting-started/prerequisites.html>

* Then follow the steps mentioned in “1\_The\_DNIF\_Installation\_Guide.docx” which is present in the repository. Here’s a link to it: <https://github.com/dnif/DigiVigi/tree/master/Guiding_Documents>
* There are videos and other documents as well which are provided by the Organization which can guide, but it does not cover the details in it which we require. Here’s a link to it:

<https://dnif.it/docs/guides/getting-started/installing-dnif.html>

**NOTE:** In our case we have done the setup on a Virtual Machine. One can have a separate server like machine with a dedicated Ubuntu/ CentOS.



*Fig 2: DNIF Installation Pointer*

**STEP 2:**

## Identifying the dataset and its source

* Here’s a guide of how we did it for starters. Refer following documents:

1. Activity initialization:

<https://github.com/dnif/DigiVigi/blob/master/Guiding_Documents/2_Dataset_Hunting_Activity.docx>

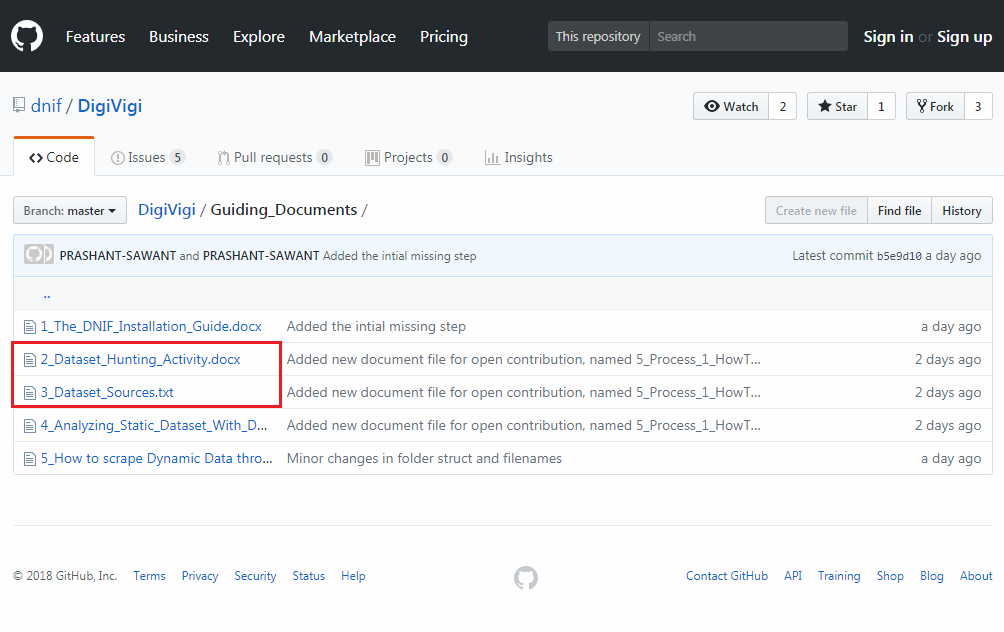
1. Jotting down dataset sources

<https://github.com/dnif/DigiVigi/blob/master/Guiding_Documents/3_Dataset_Sources.txt>

* Although it is not a must that our dataset sources might be of a good use to most analysts, since the purposes might differ.
* So in this current documented example, we’ve chosen a structured and static dataset from the source:

<https://www.webiron.com/abuse_feed/>

* Webiron is not a static source. In fact it posts newer data everyday on its website link mentioned above. Only in our case we have scrapped data from the web page for a course of 3-5 days just for demonstration purpose.



*Fig 3: Dataset Repoistory*

**STEP 3:**

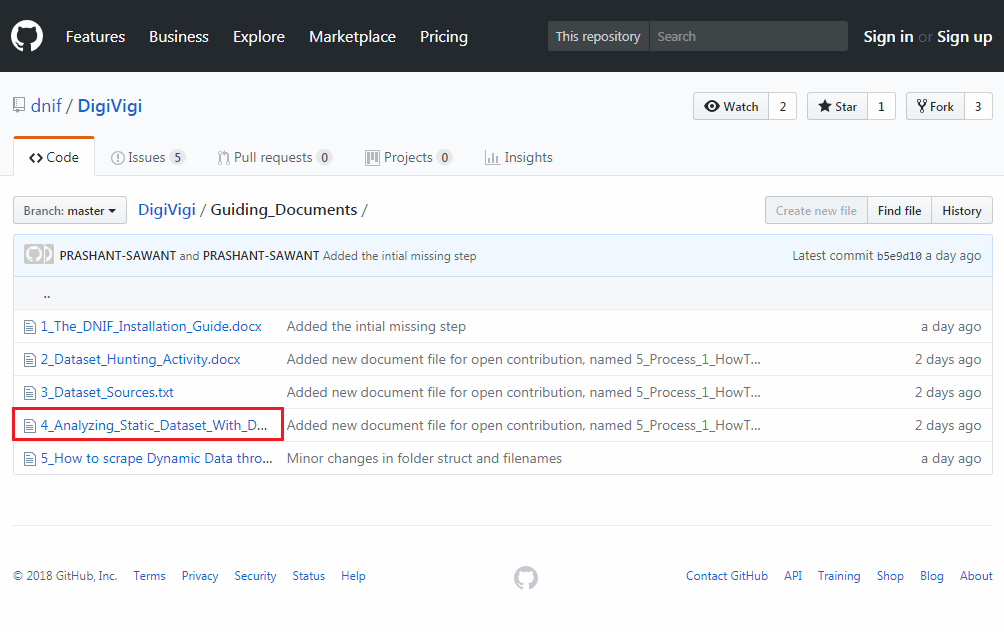
## Understanding DNIF

* This one here is a self-study step. It won’t be much of a tutorial based way through this document. But there’s no need worry, as DNIFs query language is very simple to learn if one is well versed with SQL. Yes, heard it right. They have this entire documentation of the query commands with examples.
* Head over here for getting an idea about it:

<https://dnif.it/docs/>

* The link above has everything in it for one to understand DNIF on a practical basis. From query language to tutorials and other documentations, most of it is present here originally.
* Also one of our contributors to “DNIF Open Source Project” has already given a glimpse of data analysis, widgets, dashboards and packages under the repository.

<https://github.com/dnif/DigiVigi/blob/master/Guiding_Documents/4_Analyzing_Static_Dataset_With_DNIF.docx>



*Fig 4: Analysis Documentation*

**STEP 4:**

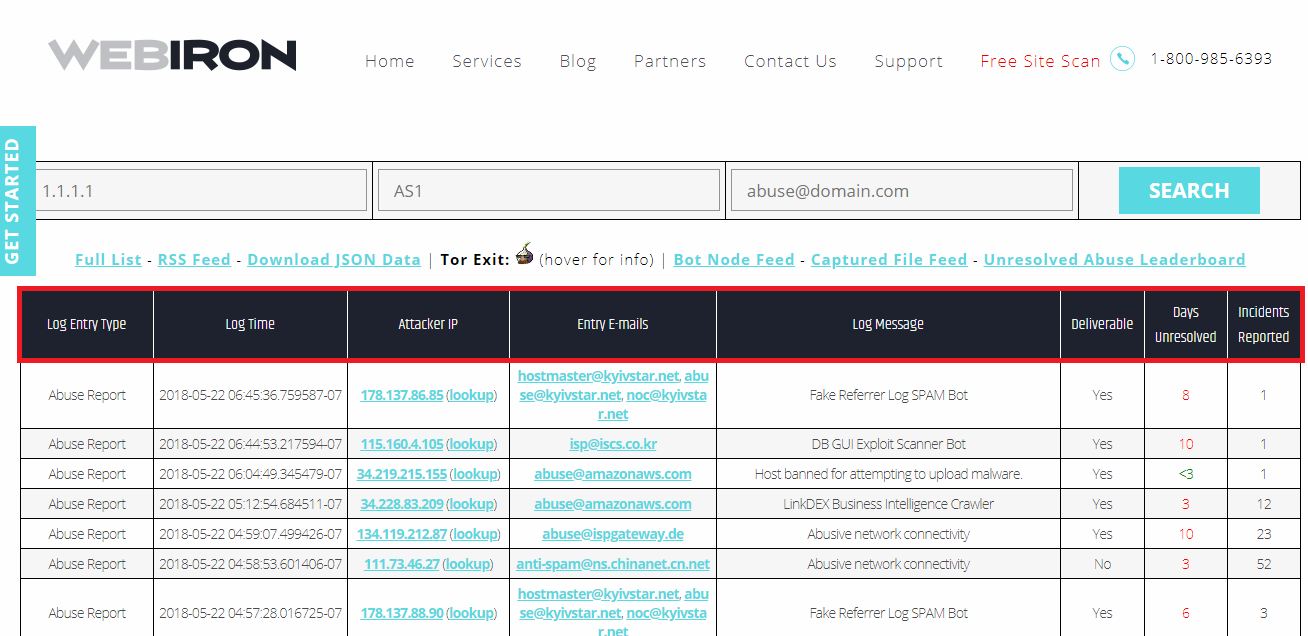
## Capturing the Dataset

Next thing is where we have written a python script which uses “Web Scrapping” – a technique used to scrape data off the HTML page.

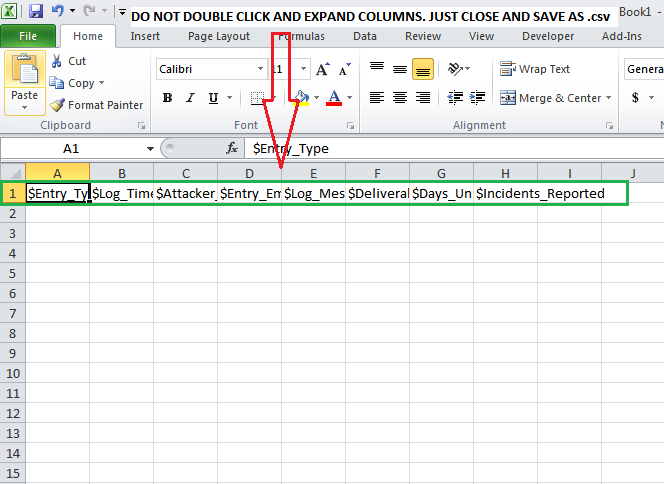
* Let us flash back a little
  + We have DNIF Installed with a connection to the container established.
  + We have our dataset source
  + Let us create a “csv” file named ‘HistoricalData.csv’ with headers predefined. This is a one-time manual thing after that every time the script is run; the data will just get appended.
  + We are being very careful with headers here since DNIF has a very convenient way of fetching records when queried. Also these headers were identified from the dataset source link.

1. Open Excel
2. Type in the headers with '$’ sign being the first character
3. Do not expand the columns, since this file is going to be saved as ‘HistoricalData.csv’
4. To save, just close and save it as ‘Comma Delimited’ file type.

* + Here’s a glimpse of how the file was created:

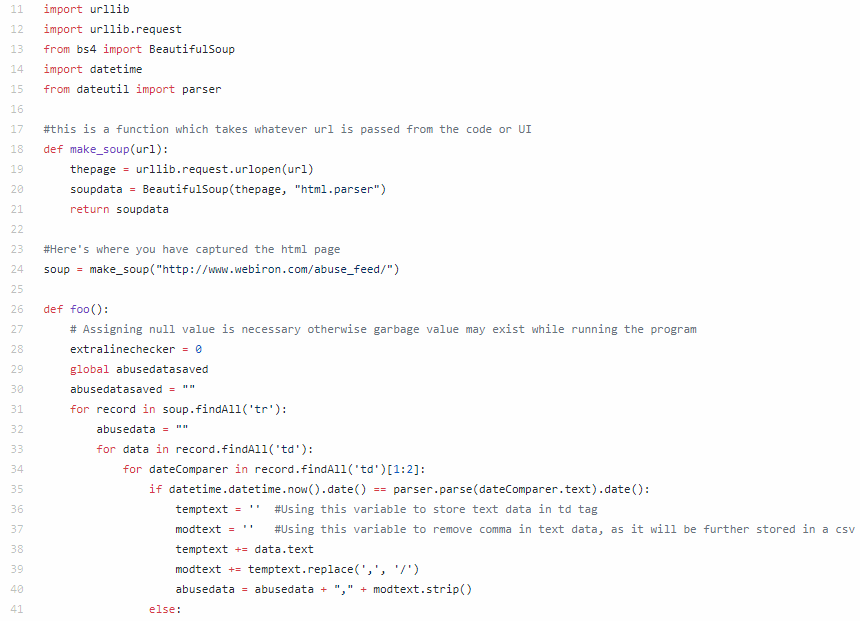


*Fig 5: Webiron Abuse Feed Header Highlight*



*Fig 6: Making that storage file*

* + Next we’ll scrape that data, using this script:



*Fig 7: Code Glimpse*

* Above is not the entire code. It is present on the repository named “PS\_KeepEveryDayData.py”

<https://github.com/dnif/DigiVigi/blob/master/Process_1/PS_KeepEverydayData.py>

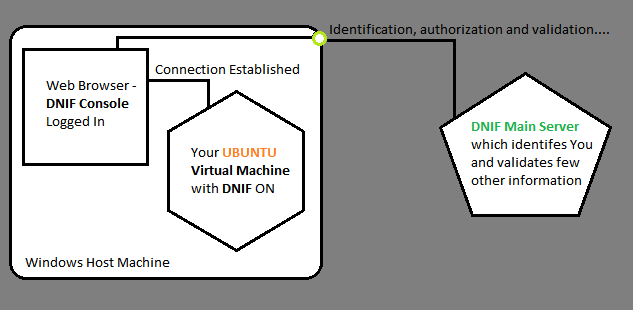
**NOTE:** If using a different website source with numerically different column numbers, please modify the code and CSV storage file accordingly.

**STEP 5:**

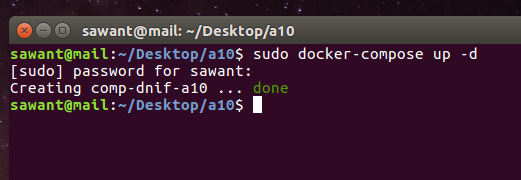
## Uploading the data

Now that our DNIF server is up, data being collected and successfully stored in the CSV file – we’re ready to upload our data through DNIF’s Web Console. Here’s how we proceed.

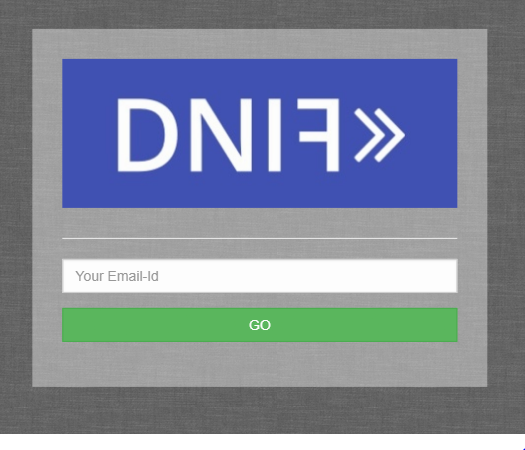
**1st:** Log in to your DNIF console and also check your entire ecosystem is successfully connected.



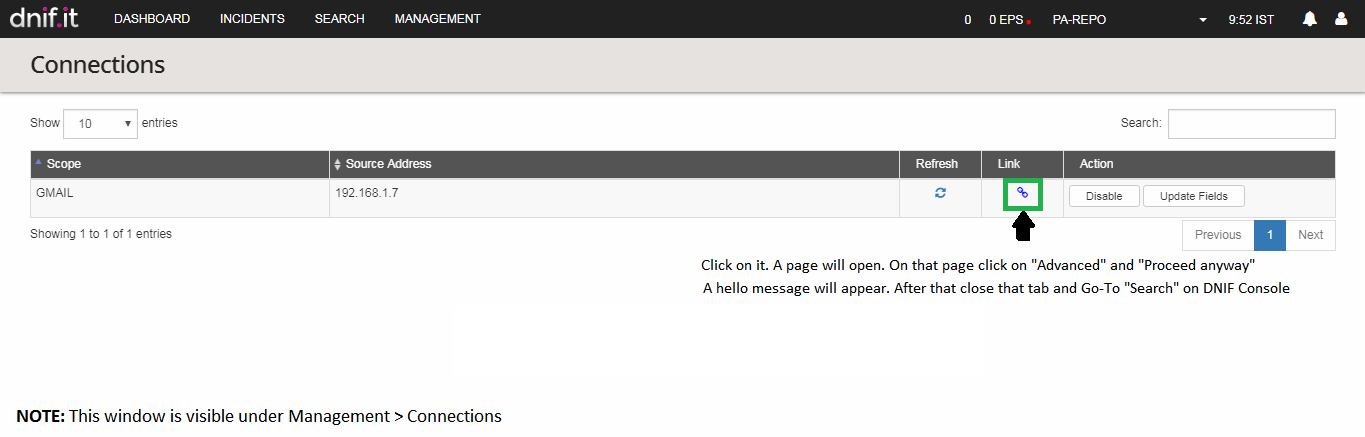
*Fig 8: Our DNIF Ecosystem*



*Fig 9: Running DNIF*



*Fig 10: DNIF Login Console from Browser*

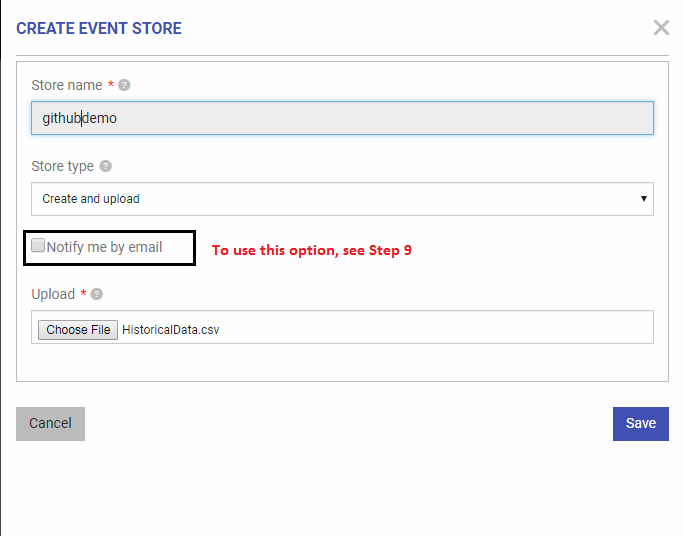


*Fig 11: Establishing Connection*

**2nd:** Next, go to search tab and upload the CSV file.

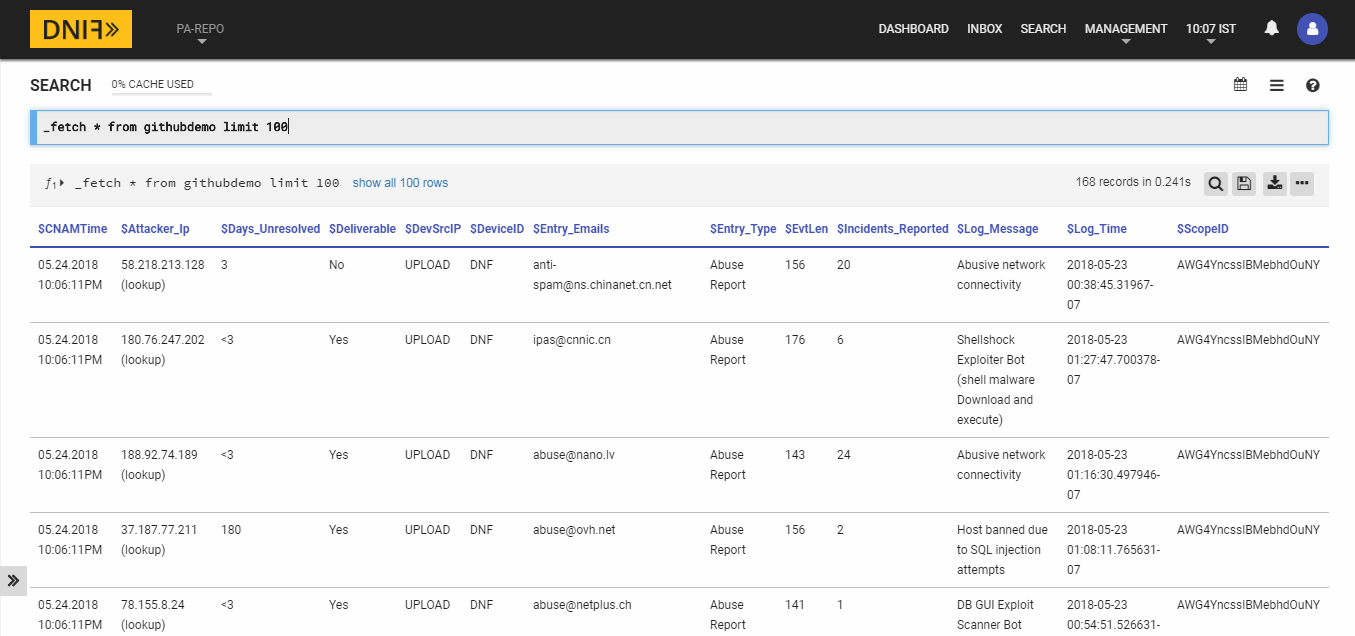


*Fig 12: Event Store Menu*



*Fig 13: Creating Event Store*

**3rd:** After successfully uploading the CSV file, first go and check if it has reflected under Management > Event Stores. If yes, then we’re good to go with running our first query and creating our first widget.



*Fig 14: First Query in DNIF*

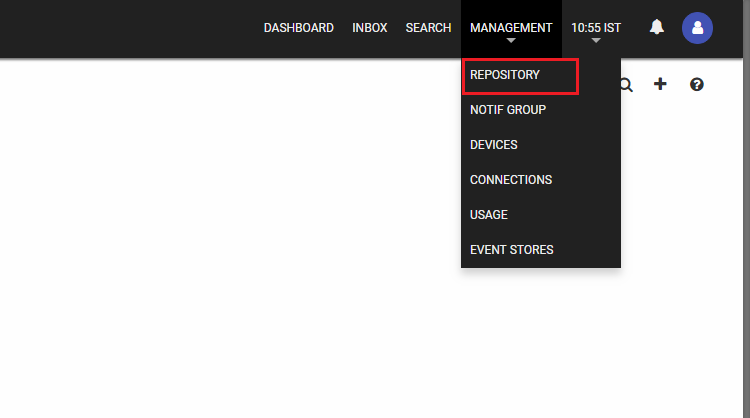
**STEP 6:**

## Creating Package

But before we create widgets and dashboard, we will first need to create a package. A package in DNIF has a larger purpose (Check out their documentation to know more)

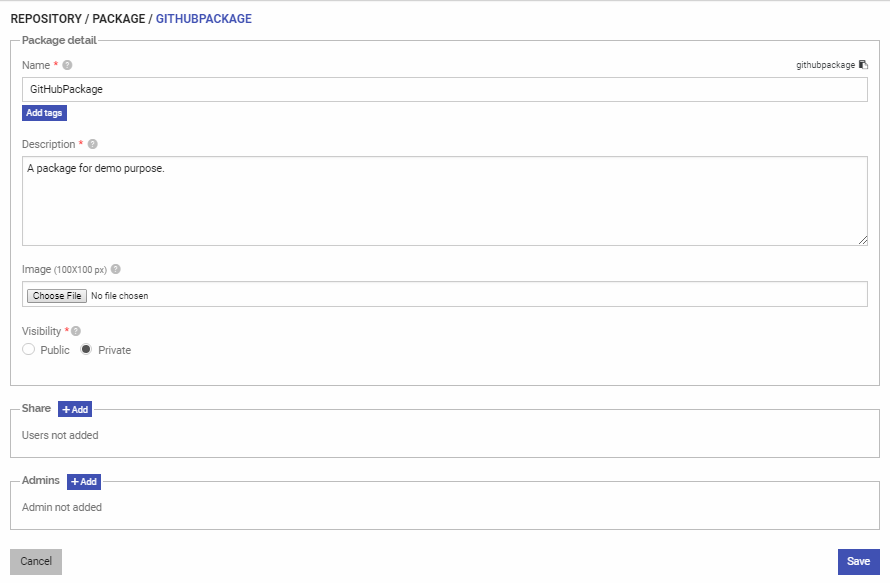
Here’s how we create a package in DNIF.

**1st:** Under management tab, click on “Repository”, to further create package. A package is like a box full of widgets, dashboards, reports and stuff. It’s like a collection.

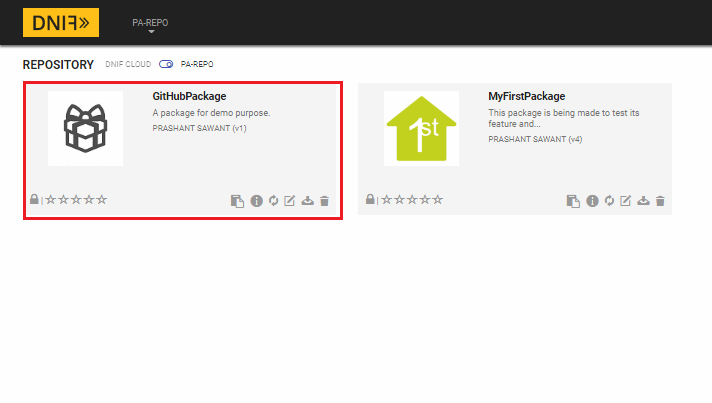


*Fig 15: Creating Package Navigation*

**2nd:** Once when under repository window, click on the plus symbol to add a new package. Add the necessary details which are fundamentally understood. When all is done – SAVE.



*Fig 16: Package Creation*

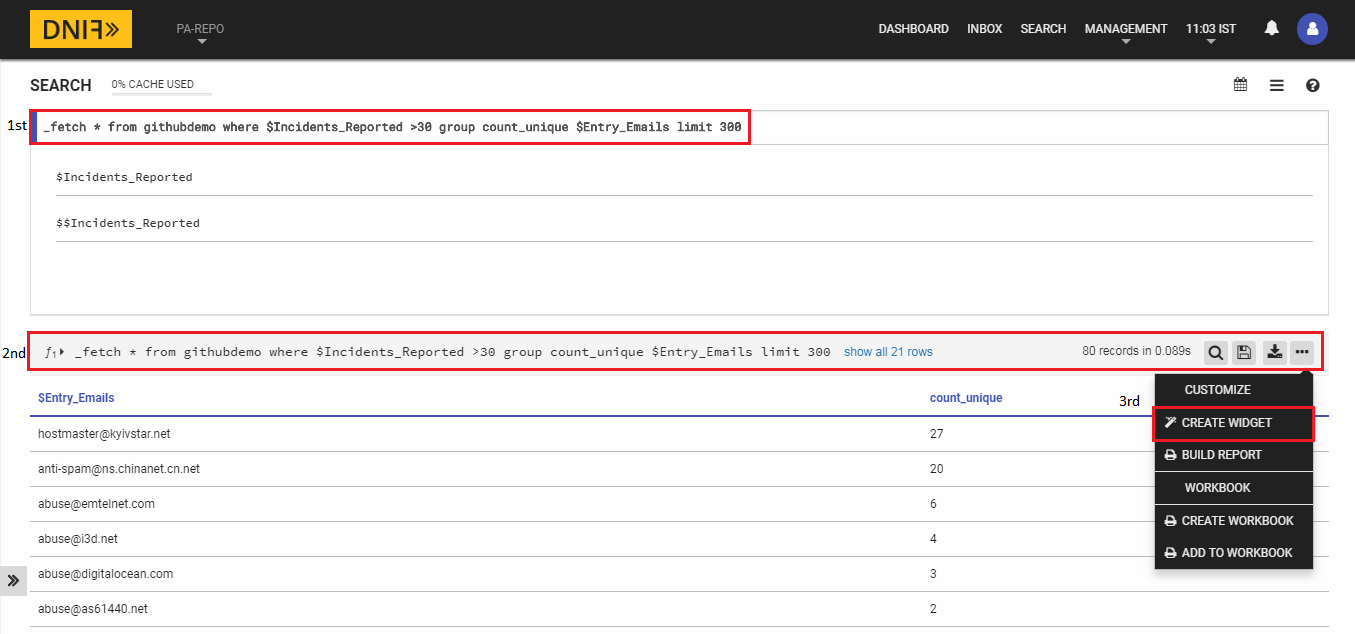


*Fig 17: Package Created*

**STEP 7:**

## Creating Widget(s)

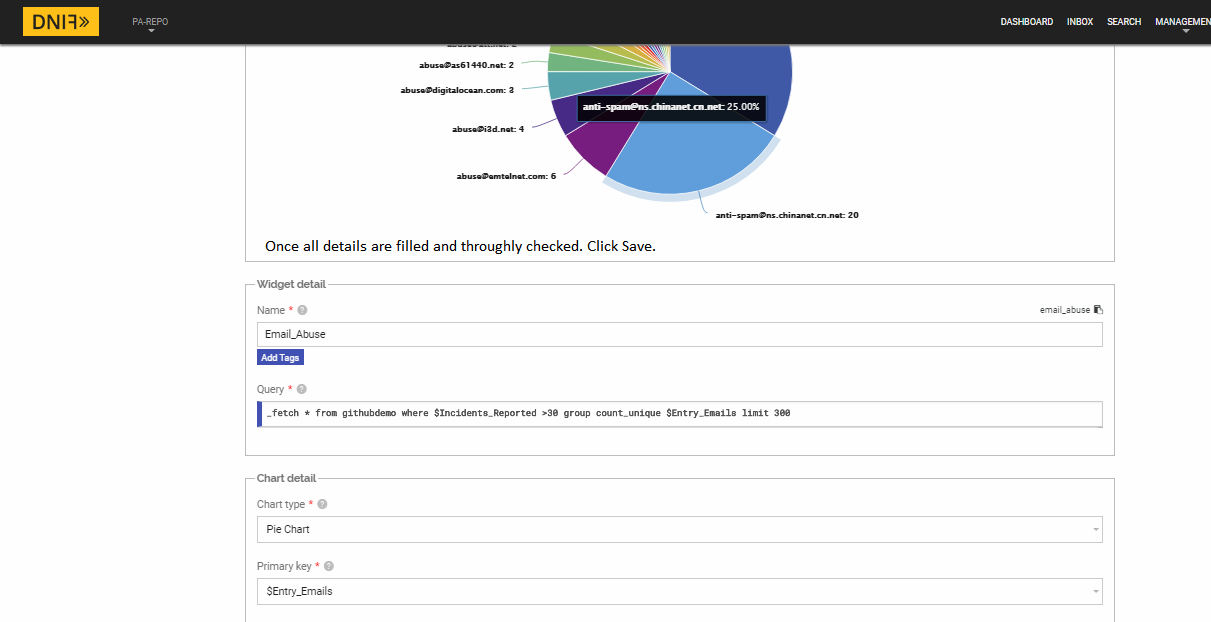
**1st:** To create a widget, we first write a meaningful query which can then be visually addressed in a widget. Then select Create Widget as shown in the image below.



*Fig 18: Creating Widget - Navigation*

Then you will be asked to select which package this widget will go under before actually creating the widget.

**2nd:** Next we fill in fundamental details that are required to give meaning to our widget. Again, very detailed information is provided on DNIF’s website.



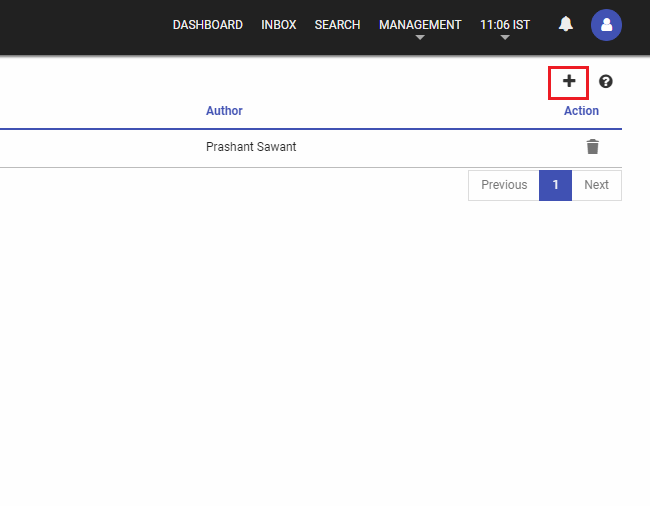
*Fig 19: Widget Information*

**STEP 8:**

## Creating Dashboard

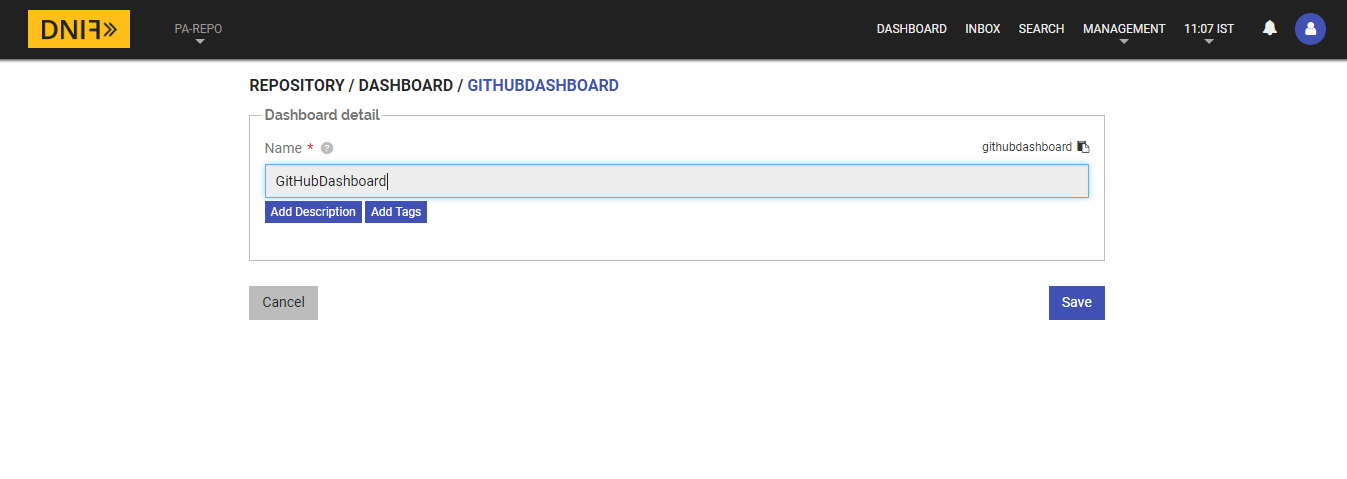
To use widgets, that are to showcase, we will need to put it on a dashboard. But before that we need a dashboard. So we create one.

**1st:** Under the “Dashboard” tab, click on the ‘+’ symbol. Then a package selection screen will appear. So select the appropriate package we have created earlier.



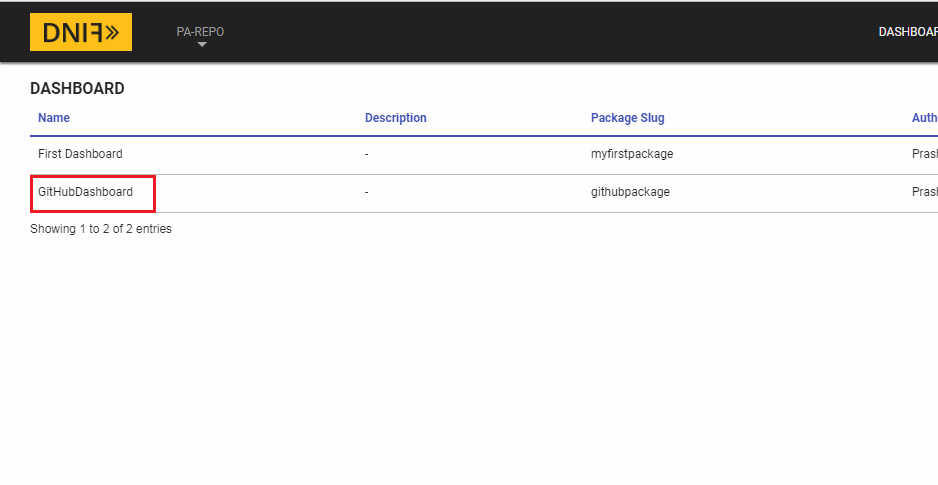
*Fig 20: Adding a Dashboard*

**2nd:** Give the dashboard a name of your choice.



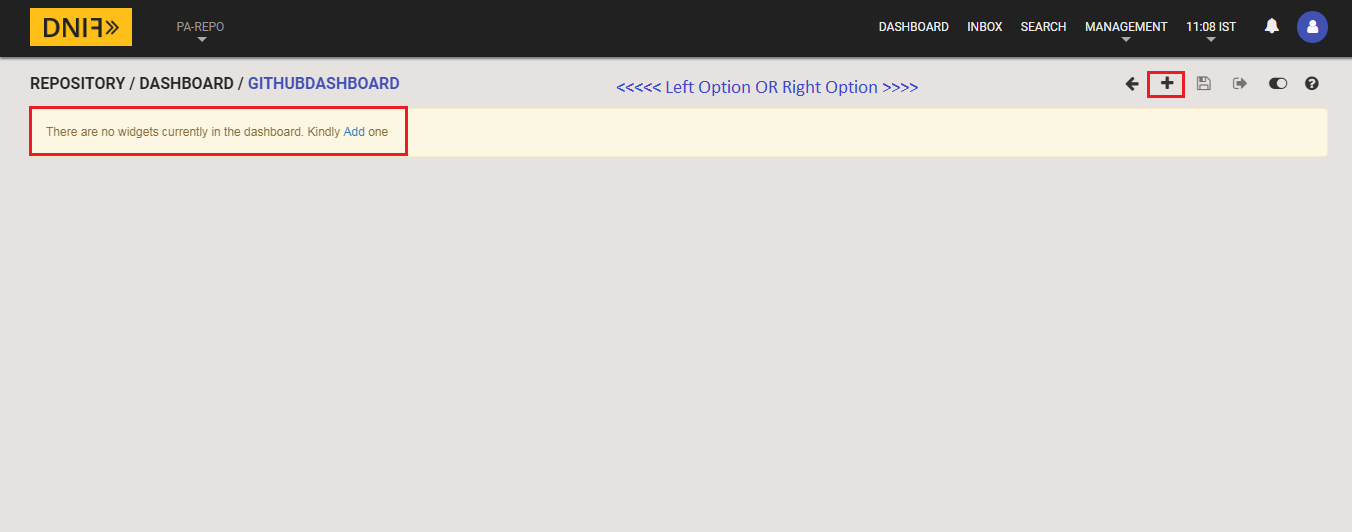
*Fig 21: Creating Dashboard*

**3rd:** Now that our dashboard is created let us put that widget we created earlier. So under dashboard page, click on the dashboard we just created.



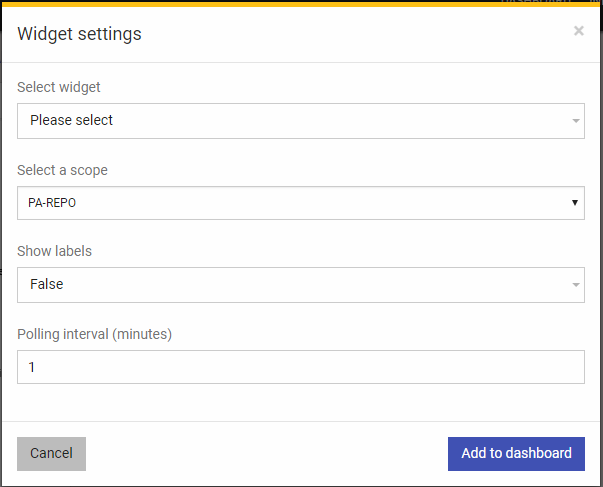
*Fig 22: Our Dashboard*

**4th:** Add a widget.

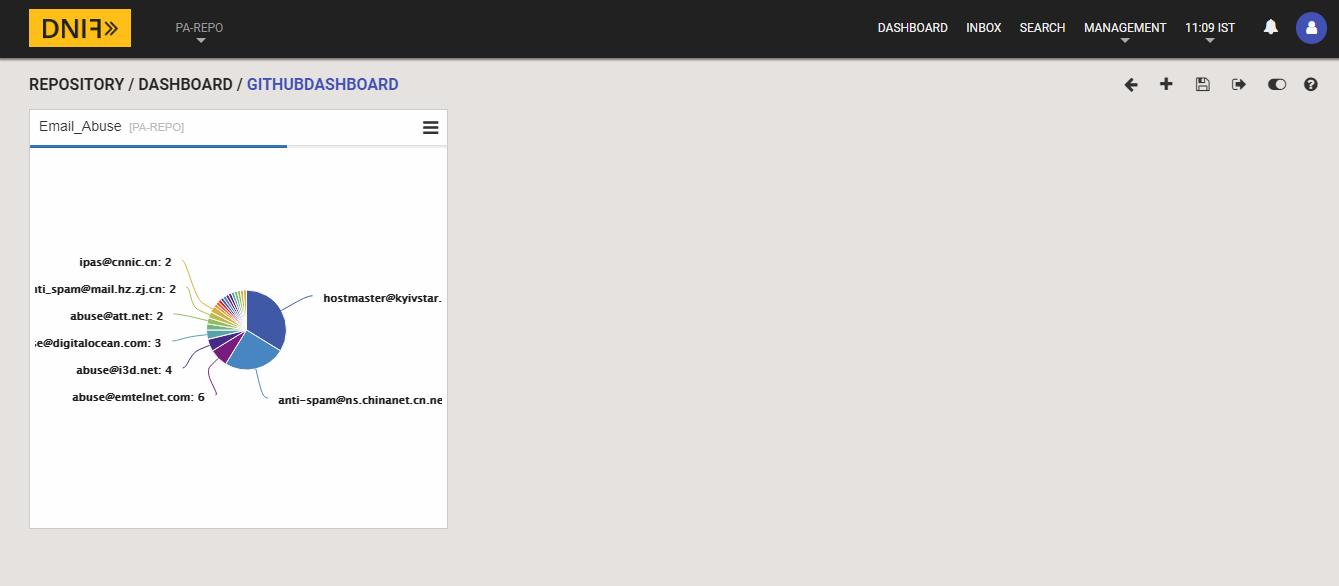


*Fig 23: Adding Widgets to Dashboard*

**5th:** After checking upon the widget settings, click on “Add to Dashboard” and your dashboard has now started to build up.



*Fig 24: Widget to Dashboard Settings*



*Fig 25: Dashboard Created*

**STEP 9:**

## Configuring SMTP for DNIF

Configuring SMTP is not a tricky thing at all. All you need to do is follow some simple steps which are documented separately under the “dnif/DigiVigi” repository. We’re not adding the document content here to avoid duplicity and avoid the already lengthy document.

Here’s the link to that document:

<https://github.com/dnif/DigiVigi/blob/master/Guiding_Documents/6_The_SMTP_Config_Guide.docx>

**STEP 10:**

## Creating Alert(s)

Creating alerts means you want notifications if certain business/ security rule is violated or if you want some daily reporting mechanism. And this can be done inside DNIF as well. This is possible only after SMTP is configured and run on DNIF, which is a one-time thing.

Here are some heads up links one should consider going through. They will help.

**Configuring SMTP:**

<https://dnif.it/docs/guides/tutorials/configuring-smtp-in-docker.html>

Or one can always refer previous step.

**How to create reports in DNIF:**

<https://dnif.it/docs/guides/tutorials/how-to-create-reports.html>

There some detailed examples given on this page which may or may not be covered under this exercise.

**Hot to create notification groups:**

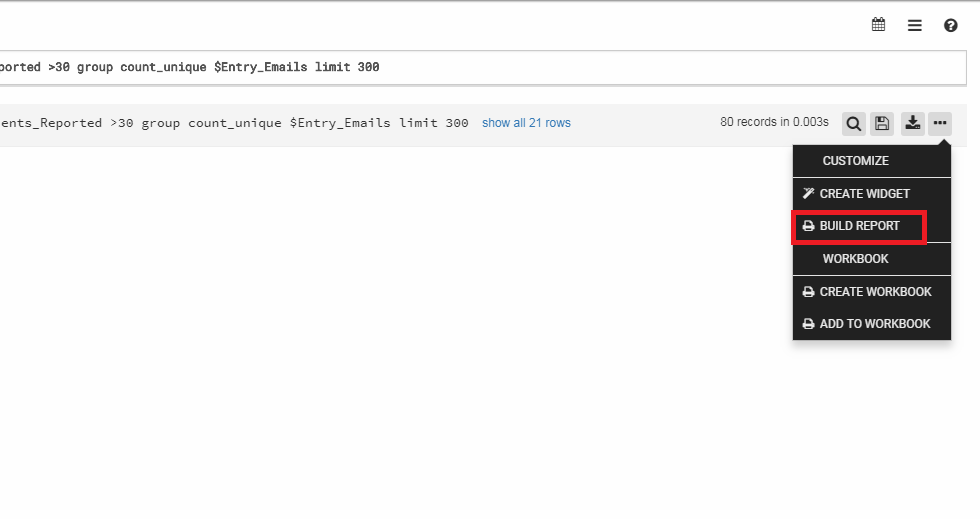
<https://dnif.it/docs/guides/tutorials/create-and-view-notif-group.html>

**How to raise alerts:**

<https://dnif.it/docs/learn/DQL/trigger.html>

There some detailed examples given on this page which may or may not be covered under this exercise.

**1st:** Let us build a report using our previous widget query.

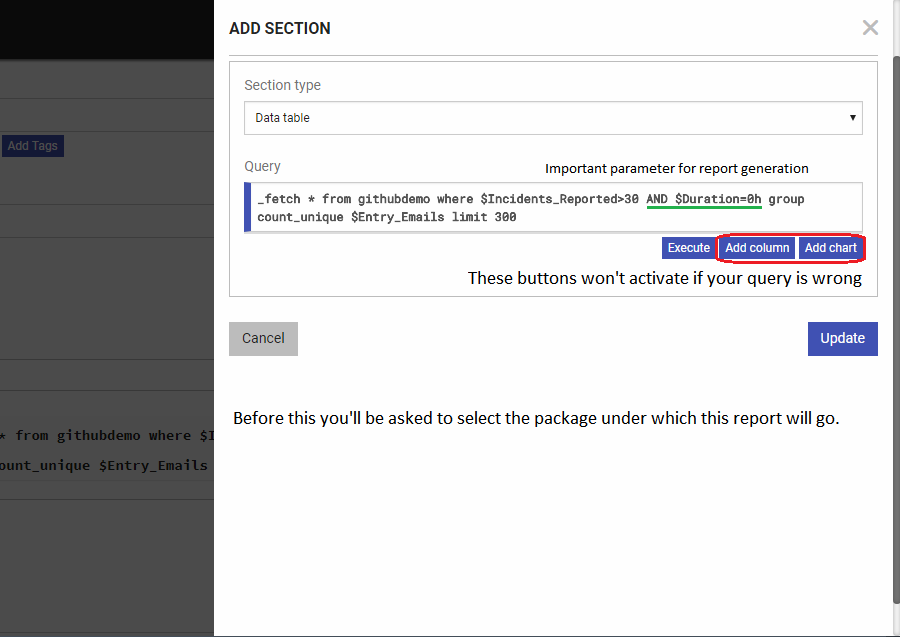


*Fig 26: Build Report Option*

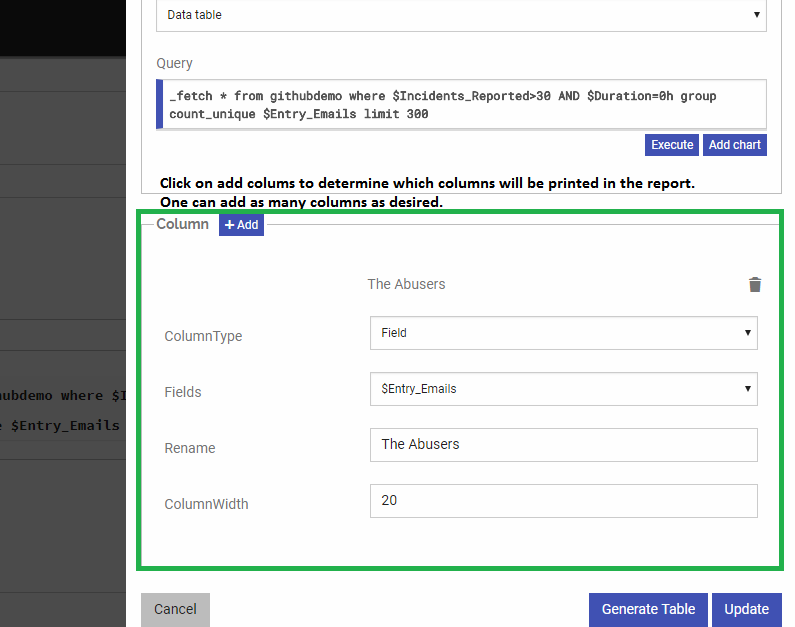
**2nd:** After selecting the package (of which a screenshot is not added), under section type – query, check if your query for report is valid or not (We’re using the same query from the widget here). In order to check/ validate your query, click on execute (the execute button is for validation only). And only after the query is fine, will the “Add column” & “Add chart” buttons will activate.

**NOTE 1:** There is other “Section Type” as well – like Title, Sub-title, Description, etc.

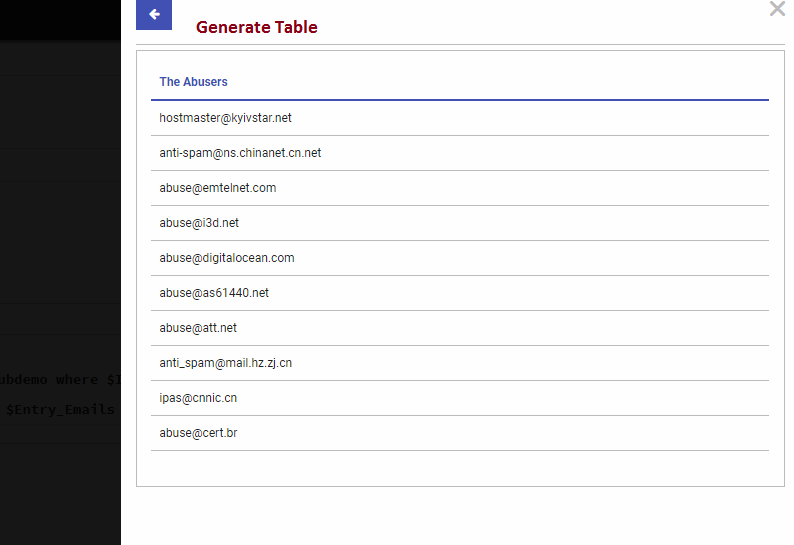
**NOTE 2:** The report query should have “$Duration” in the where clause. It is a necessary parameter.



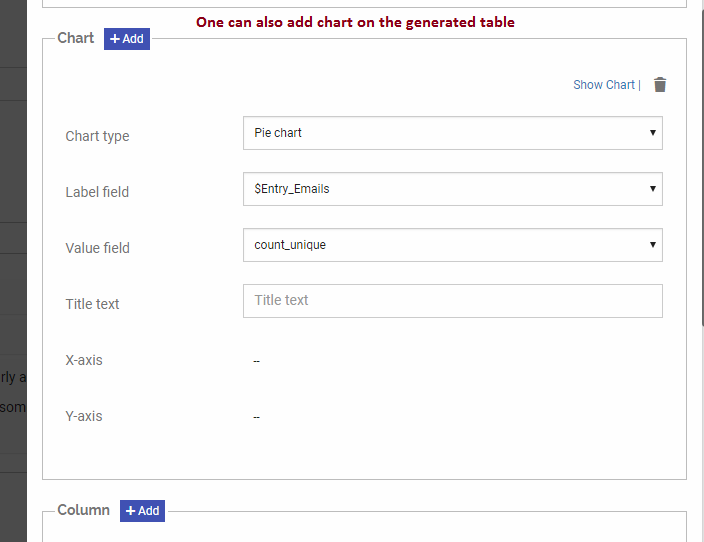
*Fig 27: Report Build Section*



*Fig 28: Adding columns to Report*



*Fig 29: Generate Table Output*

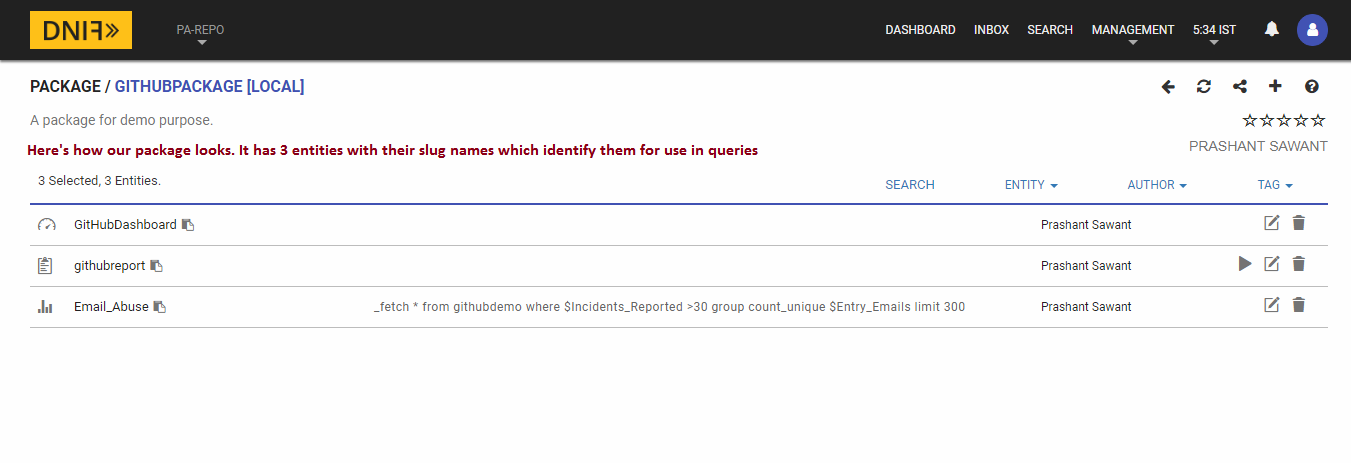


*Fig 30: Adding Visuals to Report*

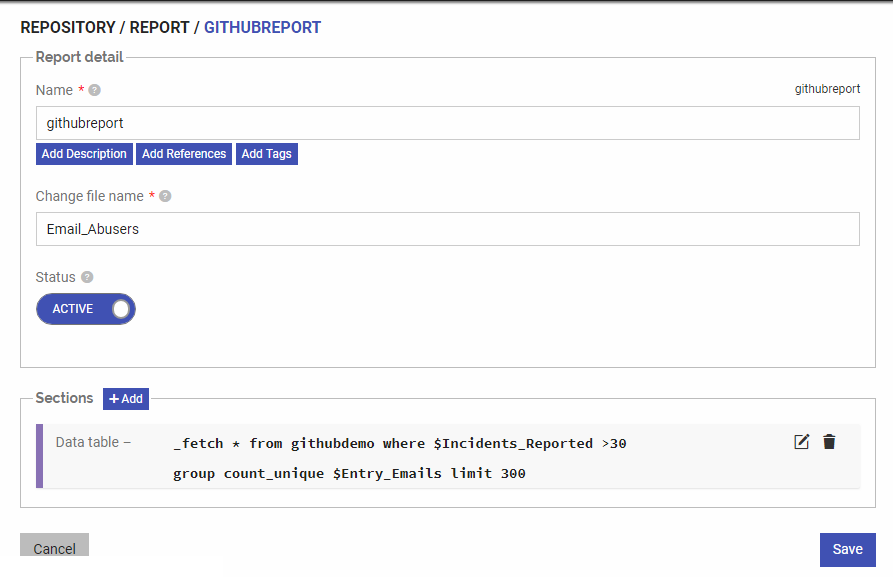
Once all the section information is provided, click on “Update” or “Done” button, whichever is visible.

**3rd:** Next we will have to check if all those contents we created are saved properly or not.

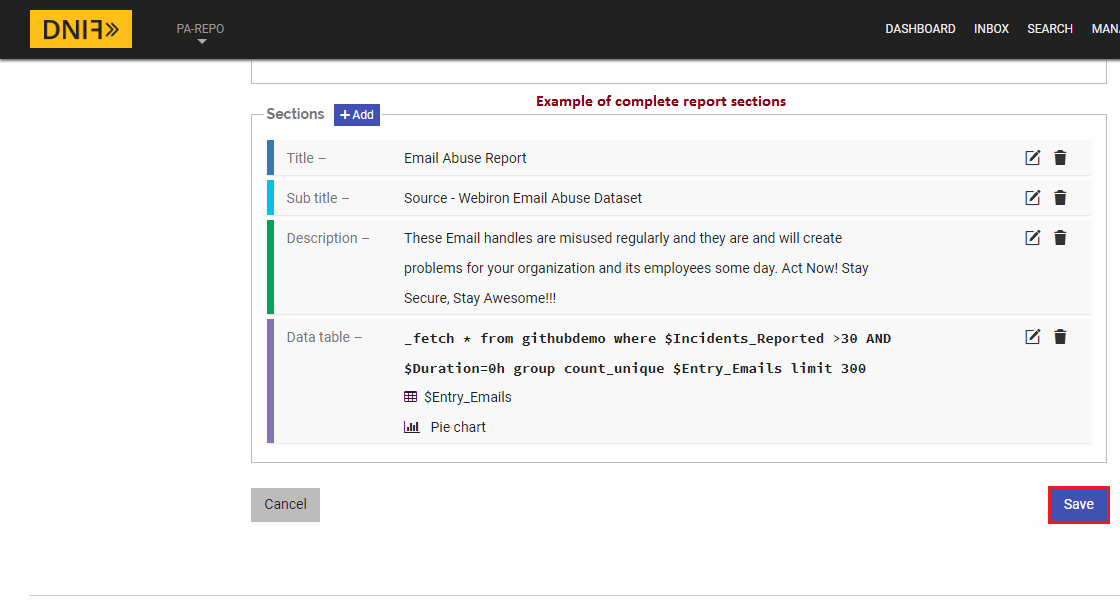
* Under “Repository Tab” where our package is – view that package and click on editing button (one with pencil symbol)
* So one can also add a report from under package.



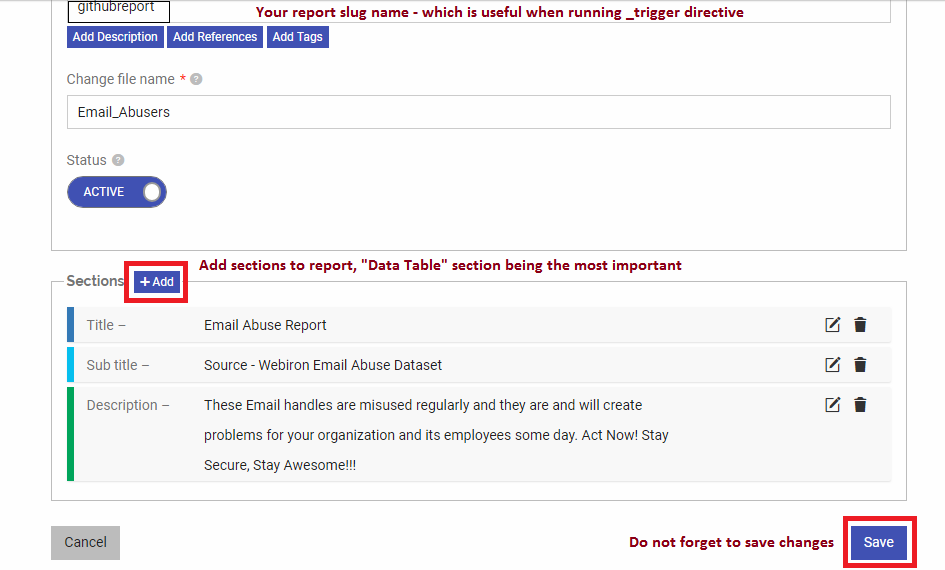
*Fig 31: Adding report option 2*



*Fig 32: Report Creation Page*



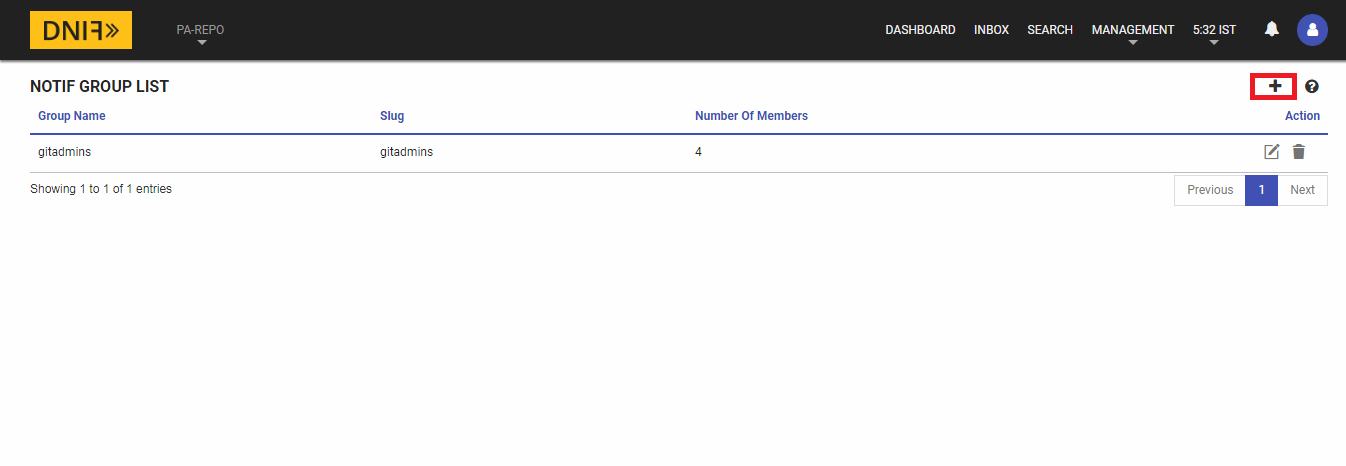
*Fig 33: Report Sections*



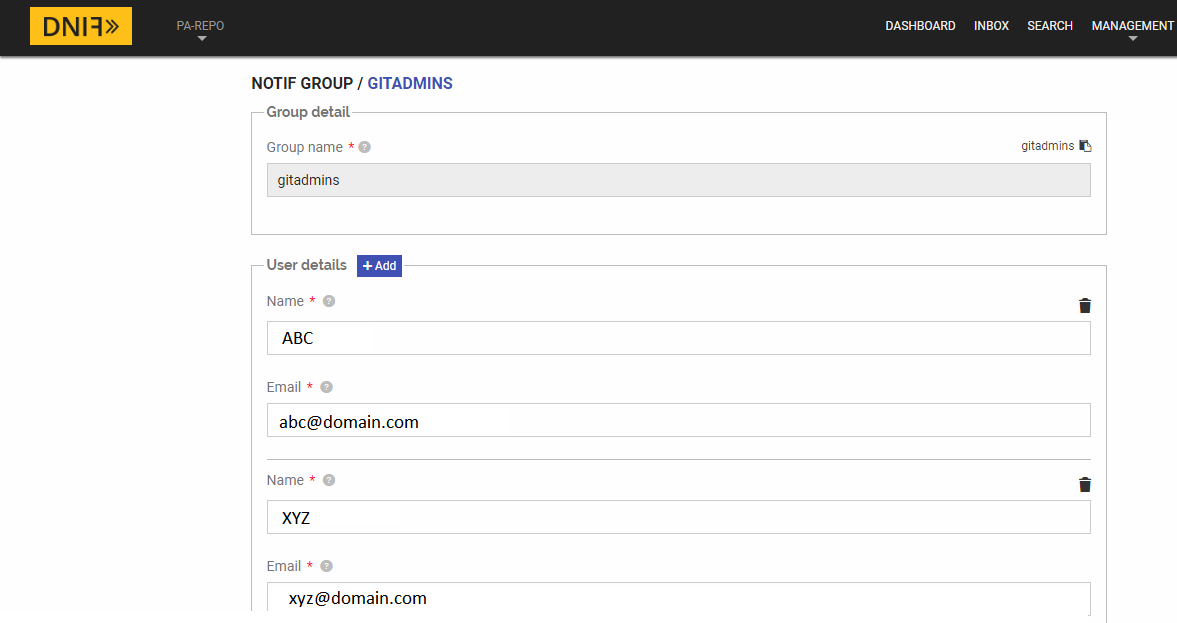
*Fig 34: Adding Section & Saving*

**4th:** Sending alerts/ reports to email users.

* One can send email alert/ report to individual email address or to a notification group – group of email addresses.
* Although the demonstrated example is one which sends email alert to a single user. But let us learn to create a notification group as well.

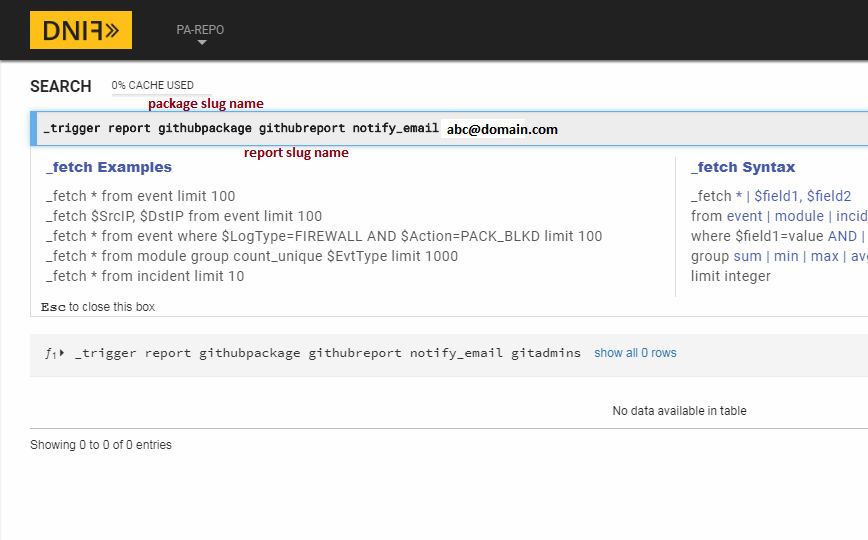


*Fig 35: Notification Groups Page*



*Fig 36: Creating Notification Group*

**5th:** Running our first trigger directive query.



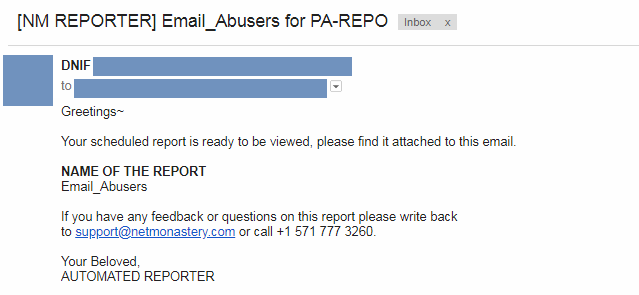
*Fig 37: Running the trigger*

Once this trigger query is run, all we have to do is check for that email.

This trigger can also be automated by putting it inside \_raise directive.

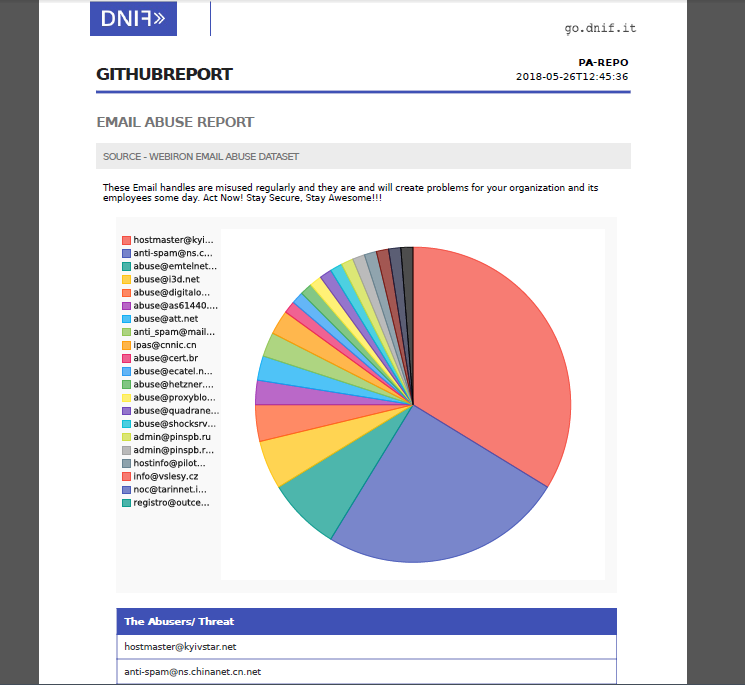
To know about \_raise, visit DNIF Website Documentations page.

**6th:** Email Alert Generated



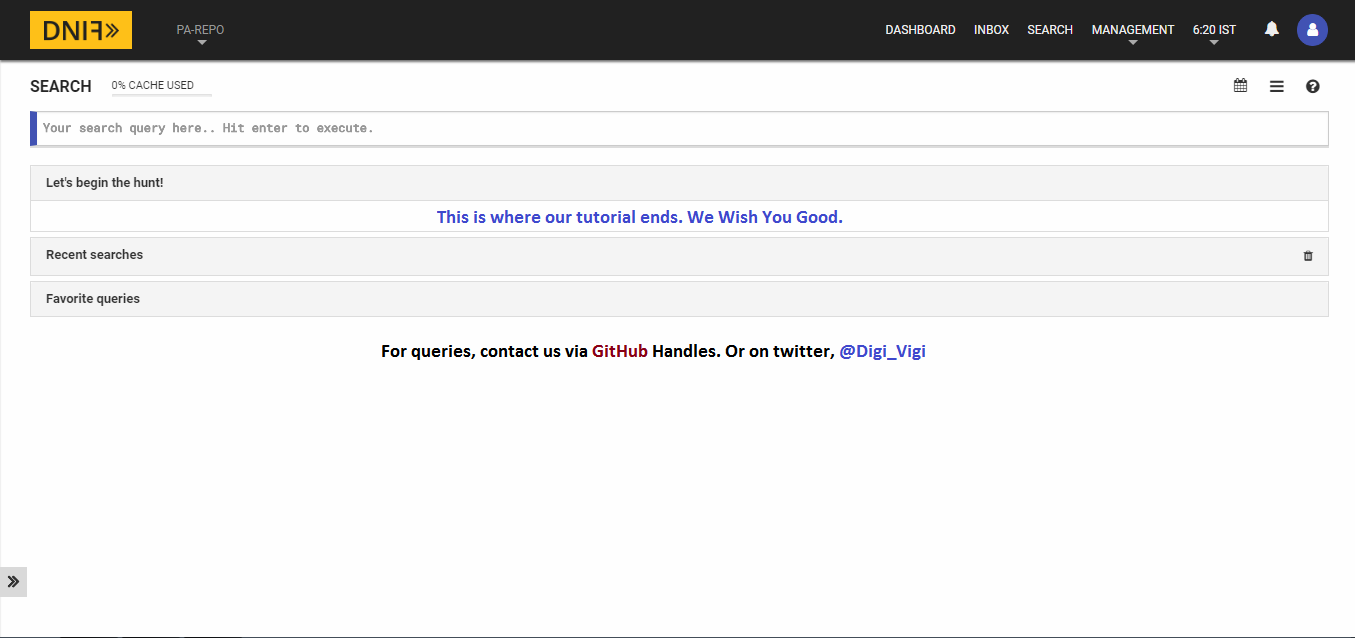
*Fig 38: Email Generation*

**7th:** Our report, which comes as an attachment to the email above.



*Fig 39: PDF Report Generated*

**PROCESS 1 COMPLETE:**

****

*Fig 40: End Credits*

**END NOTE:**

The DNIF documentation guide, the guideline on our repository, & DNIF Google Group – dnifHQ is all the help one would need to complete this process. Thank You.