

Contents

[Description 2](#_Toc514973742)

[How to Do It? 3](#_Toc514973743)

[Installing DNIF 3](#_Toc514973744)

[Identifying the dataset and its source 4](#_Toc514973745)

[Understanding DNIF 5](#_Toc514973746)

[Capturing the Dataset 6](#_Toc514973747)

[Uploading the data 8](#_Toc514973748)

[Creating Package 12](#_Toc514973749)

[Creating Widget(s) 14](#_Toc514973750)

[Creating Dashboard 16](#_Toc514973751)

[Configuring SMTP for DNIF 19](#_Toc514973752)

[Creating Alert(s) 19](#_Toc514973753)

# 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:*

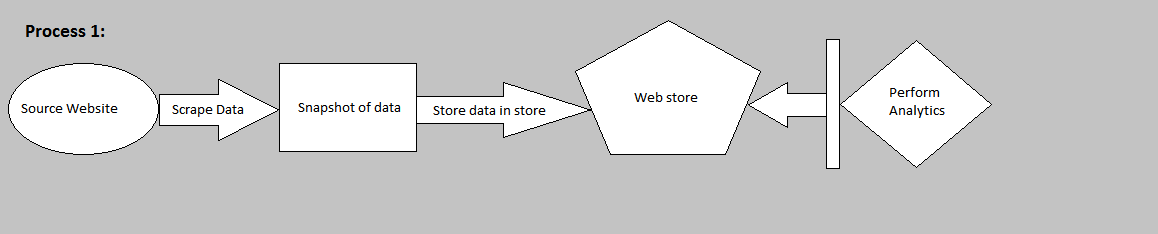
* Understand DNIF platform and its capabilities limited to project scope.
* This step was introduced so that, one is beforehand ready and understands what the platform is, how it works, what are its capabilities, its features, the query language, deployment models, limitations, etc.

*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:

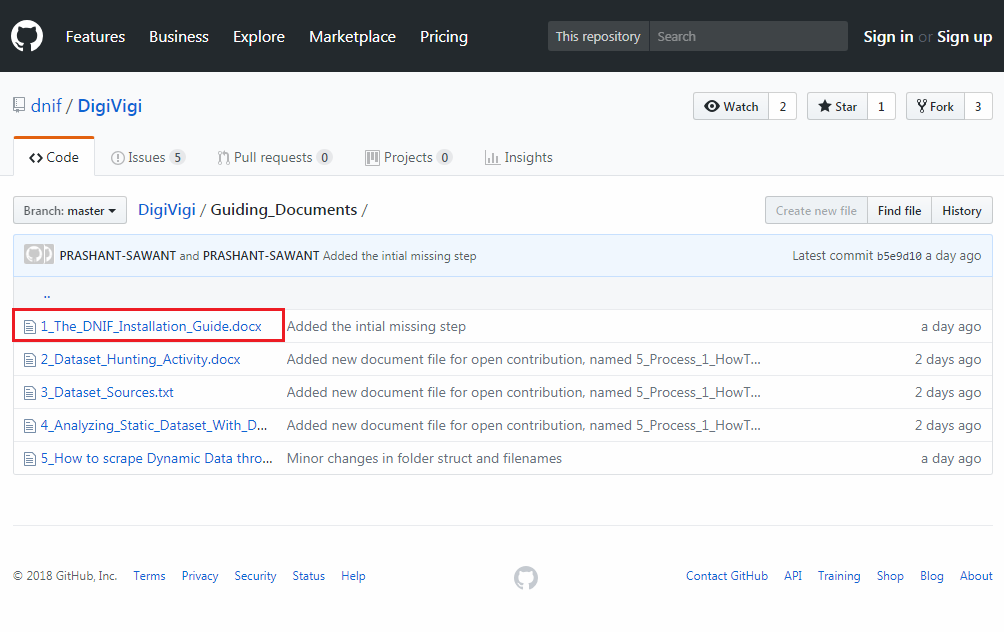
Sign-up: <https://dnif.it/signup.html>

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>

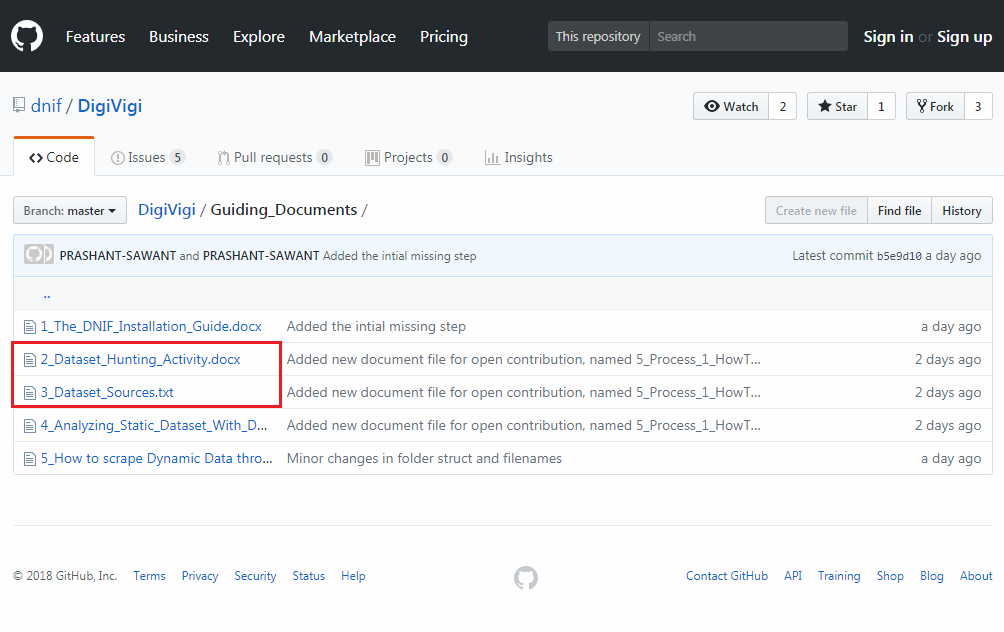
1. Jotting down dataset sources

<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:**

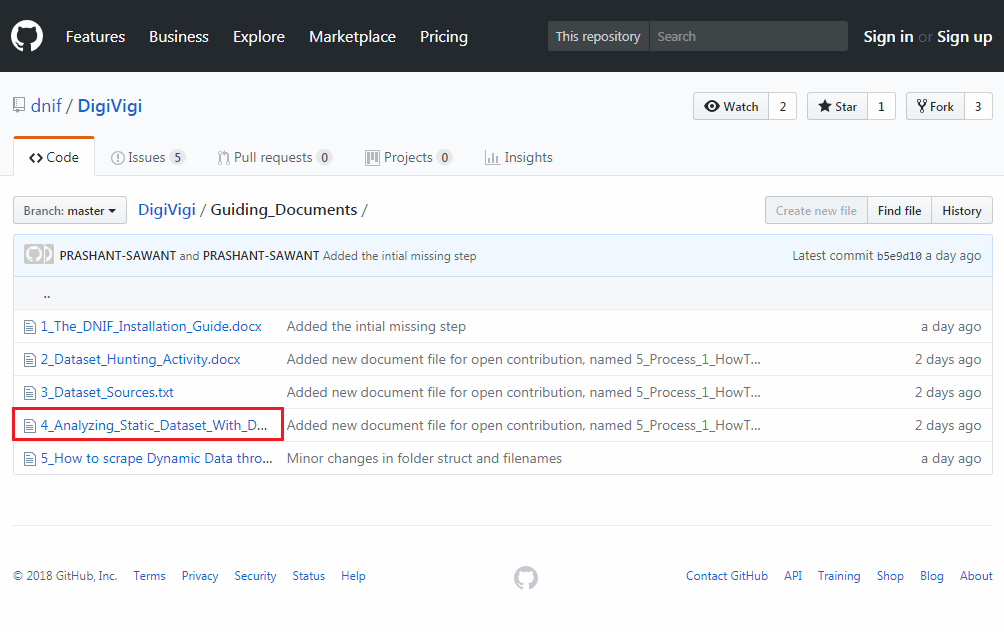
## Understanding DNIF

* 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:**

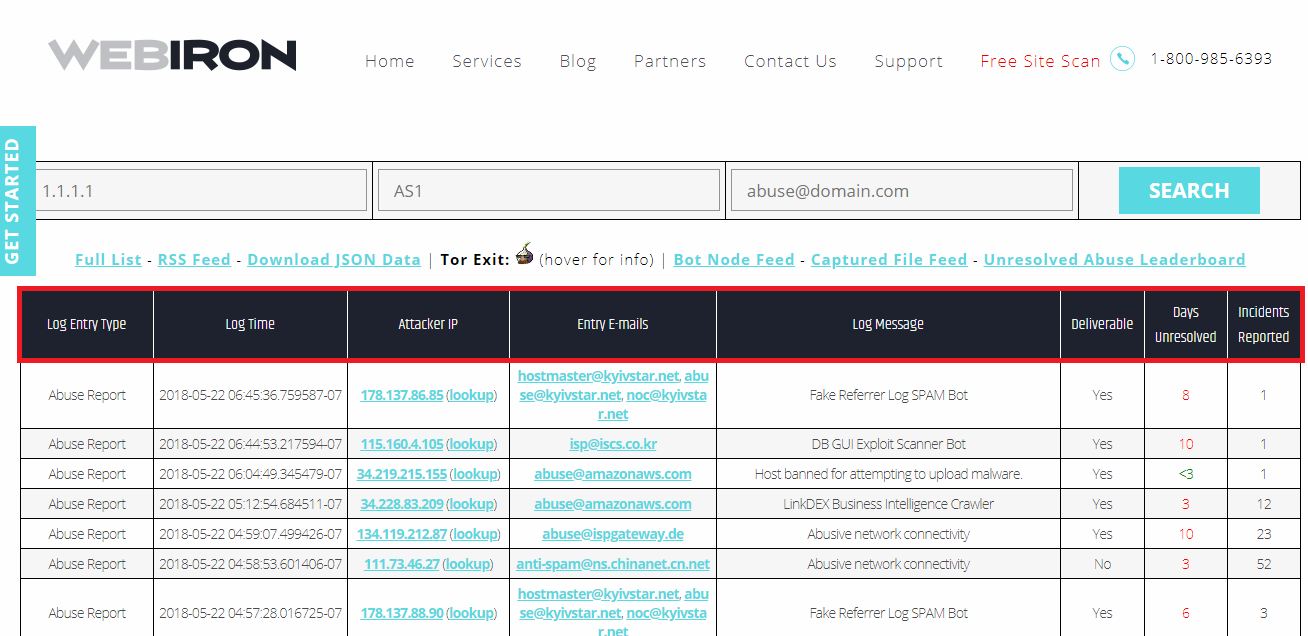
## Capturing the Dataset

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

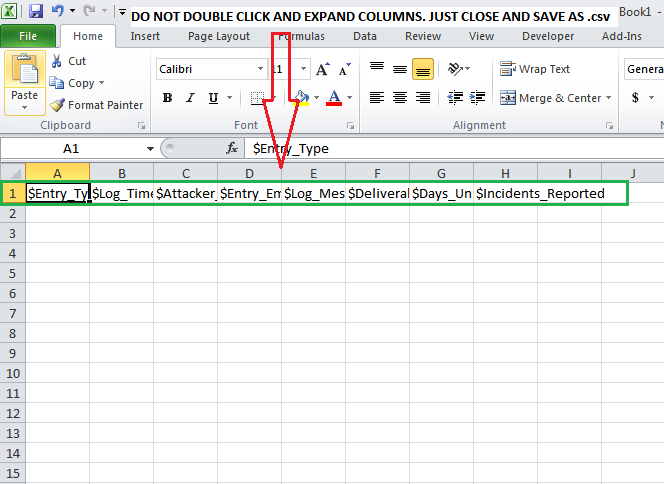
* Let us flash back a little
  + We have DNIF Installed with a connection to the container established.
  + We have our dataset source
  + Let us create a “csv” file named ‘HistoricalData.csv’ with headers predefined. This is a one-time manual thing after that every time the script is run; the data will 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:

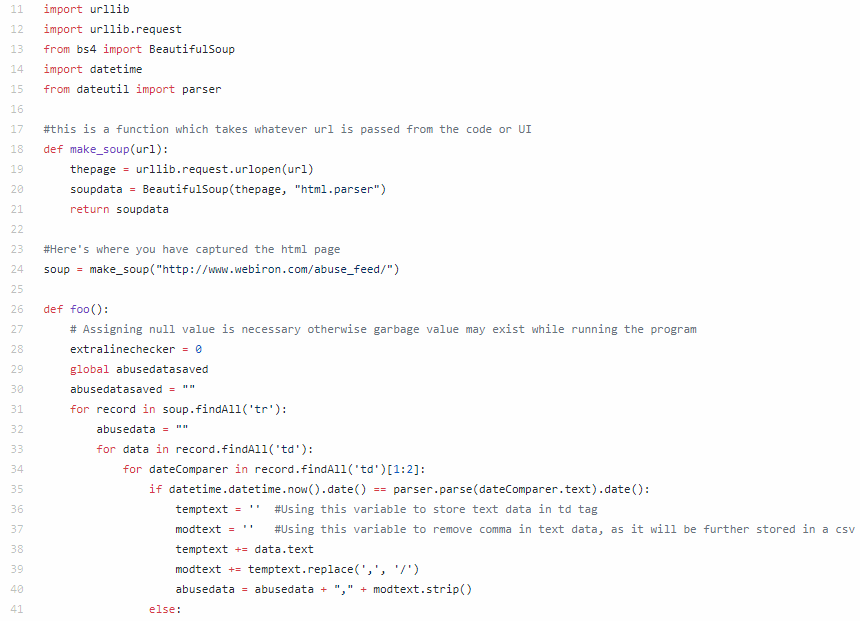


*Fig 5: Webiron Abuse Feed Header Highlight*



*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>

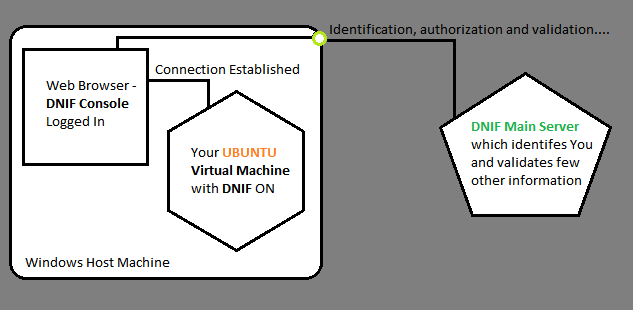
**NOTE:** If using a different website source with numerically different column numbers, please modify the code and CSV storage file accordingly.

**STEP 5:**

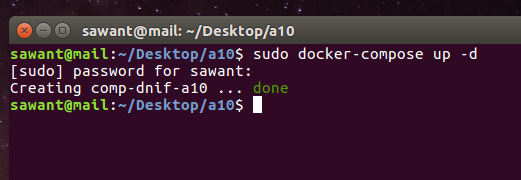
## Uploading the data

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.

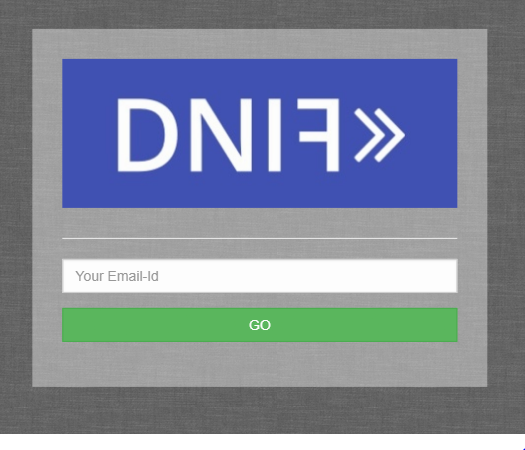
1st: Log in to your DNIF console and also check your entire ecosystem is successfully connected.



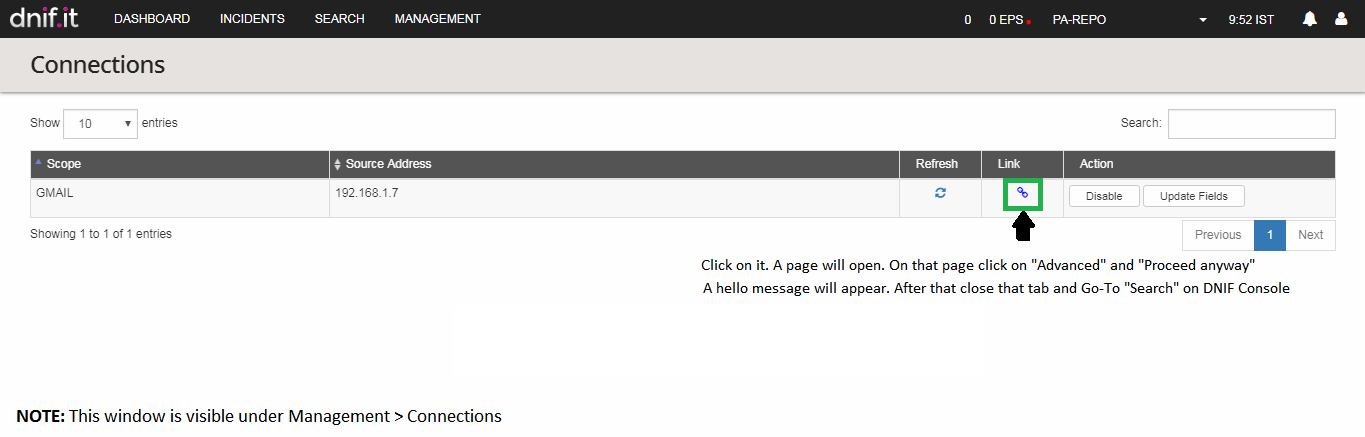
*Fig 8: Our DNIF Ecosystem*



*Fig 9: Running DNIF*



*Fig 10: DNIF Login Console from Browser*

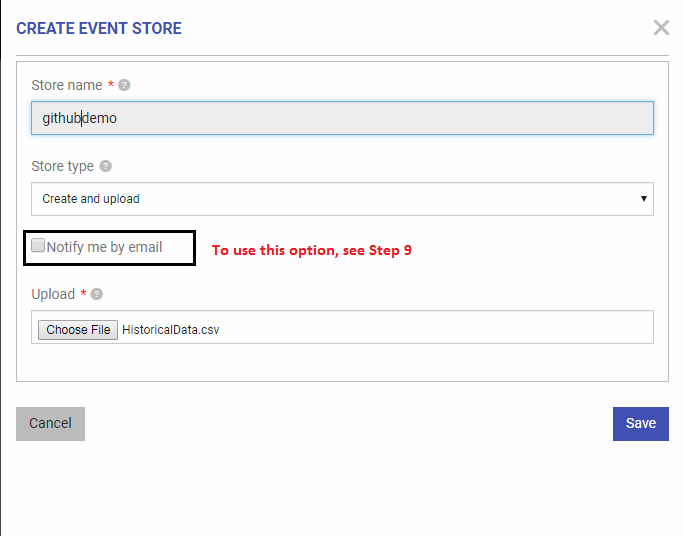


*Fig 11: Establishing Connection*

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

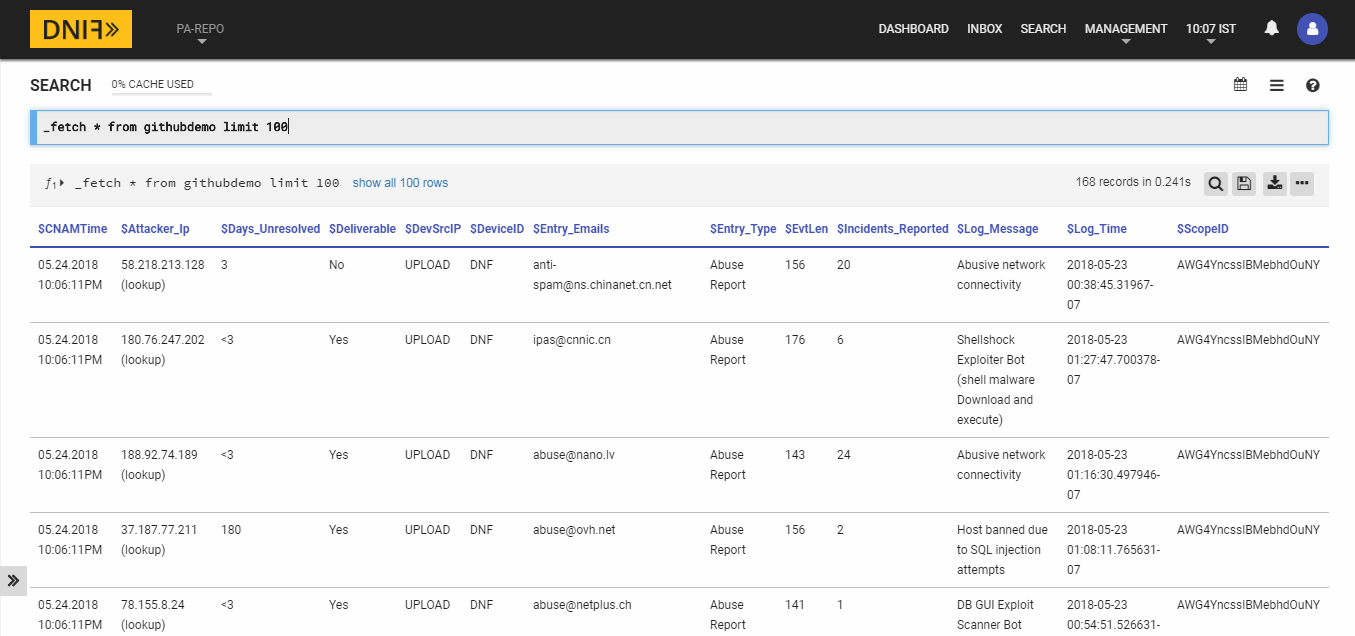


*Fig 12: Event Store Menu*



*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.



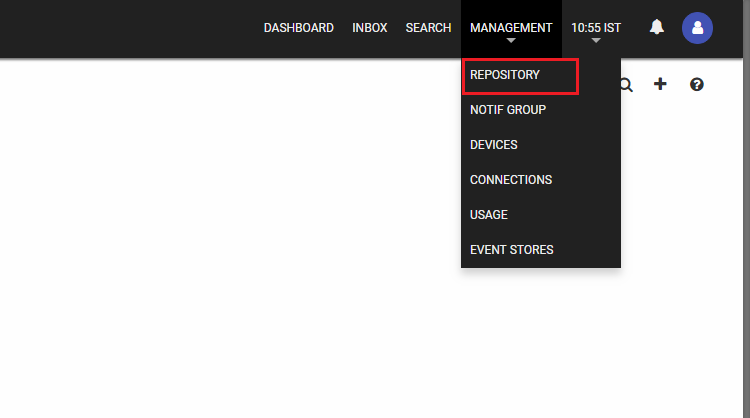
*Fig 14: First Query in DNIF*

**STEP 6:**

## Creating Package

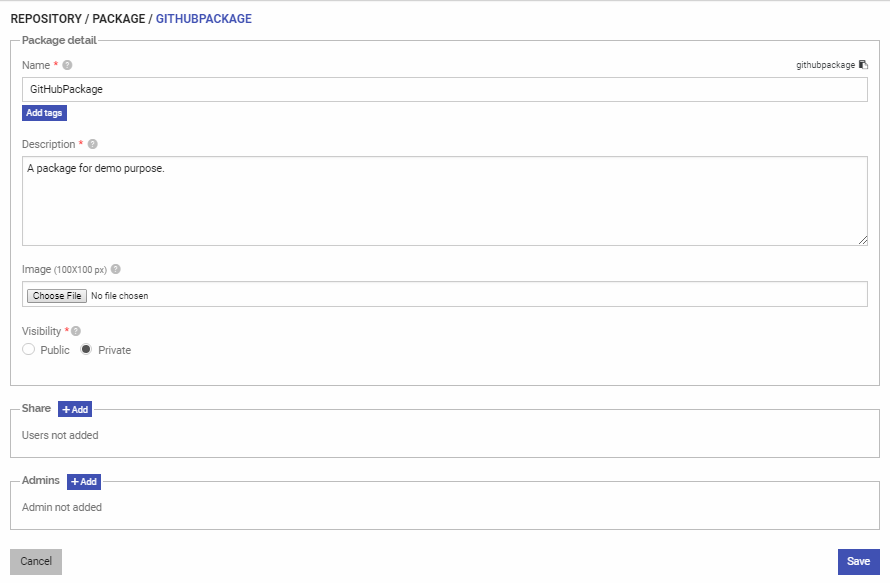
But before we create widgets and dashboard, we will first need to create a package. A package in DNIF has a larger purpose (Check out their documentation to know more)

Here’s how we create a package in DNIF.

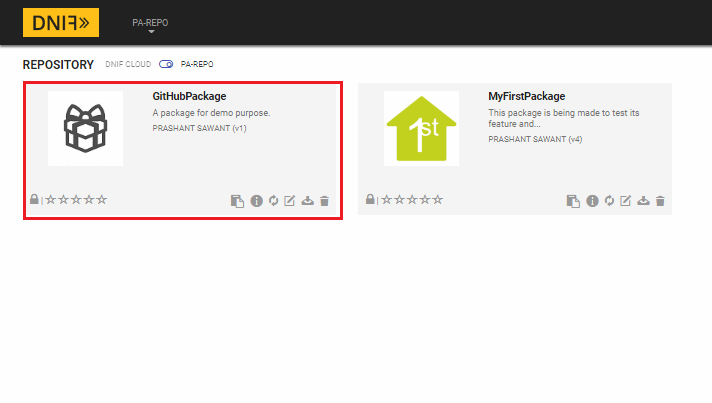


*Fig 15: Creating Package Navigation*

Once when under repository window, click on the plus symbol to add a new package.



*Fig 16: Package Creation*

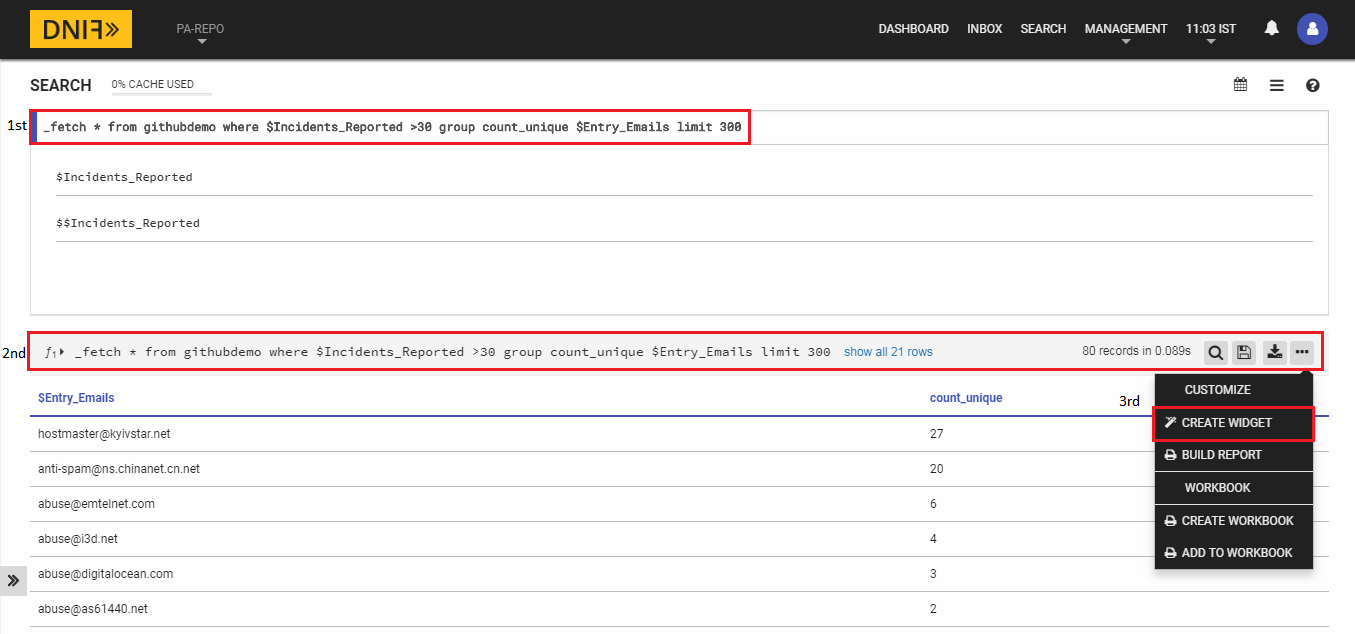


*Fig 17: Package Created*

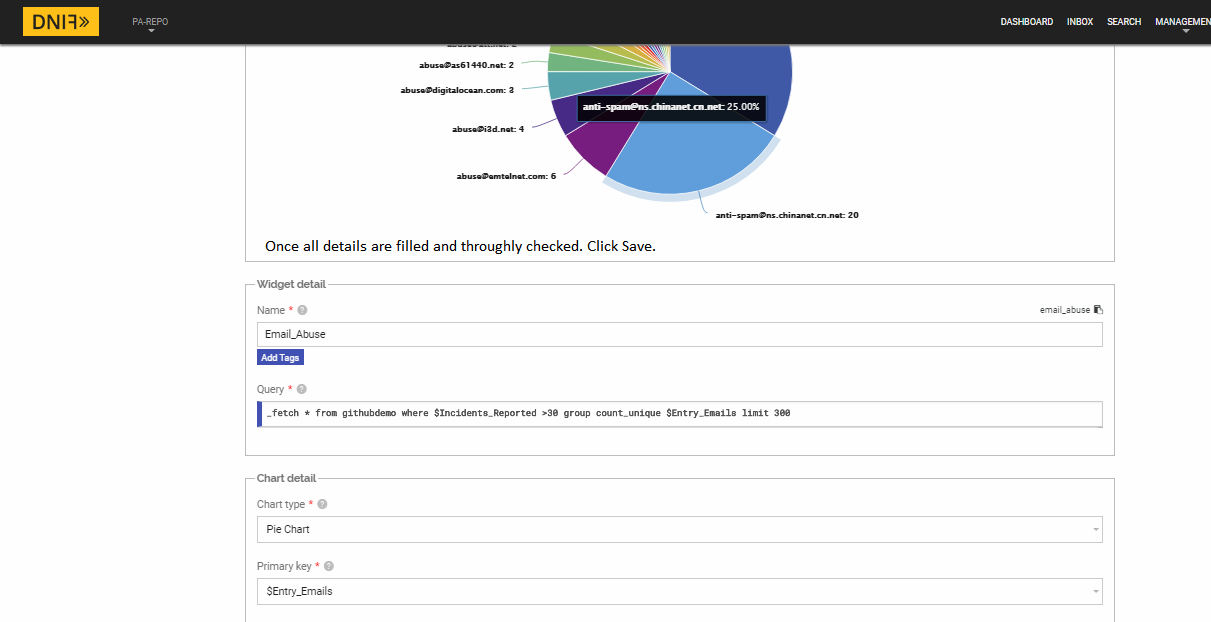
**STEP 7:**

## Creating Widget(s)

To create a widget, we first write a meaningful query which can then be visually addressed in a widget.



*Fig 18: Creating Widget - Navigation*

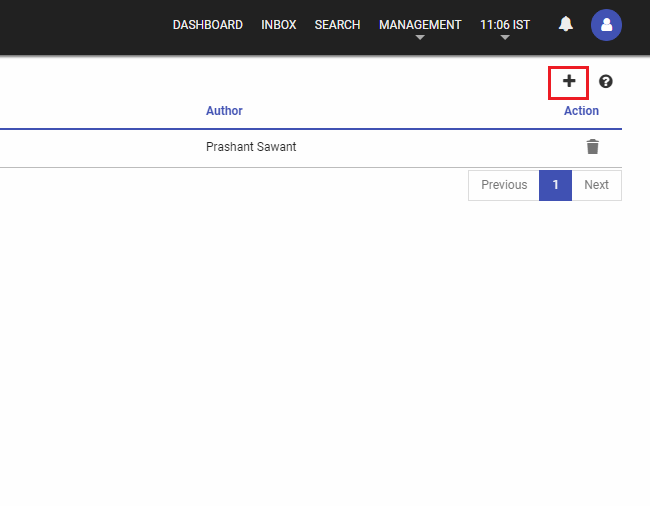


*Fig 19: Widget Information*

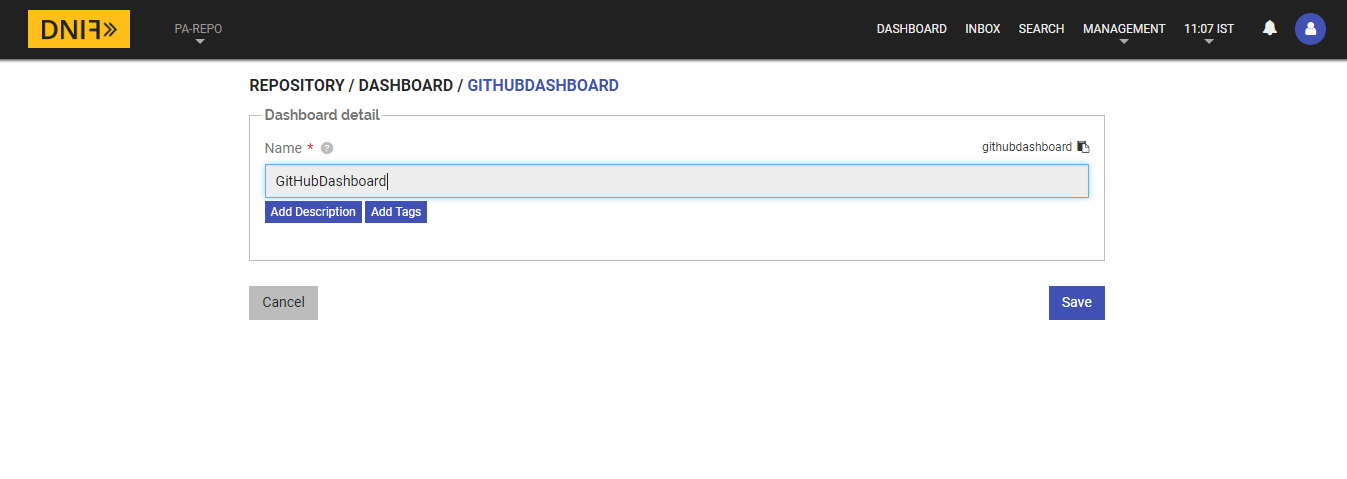
**STEP 8:**

## 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, under the “Dashboard” tab.

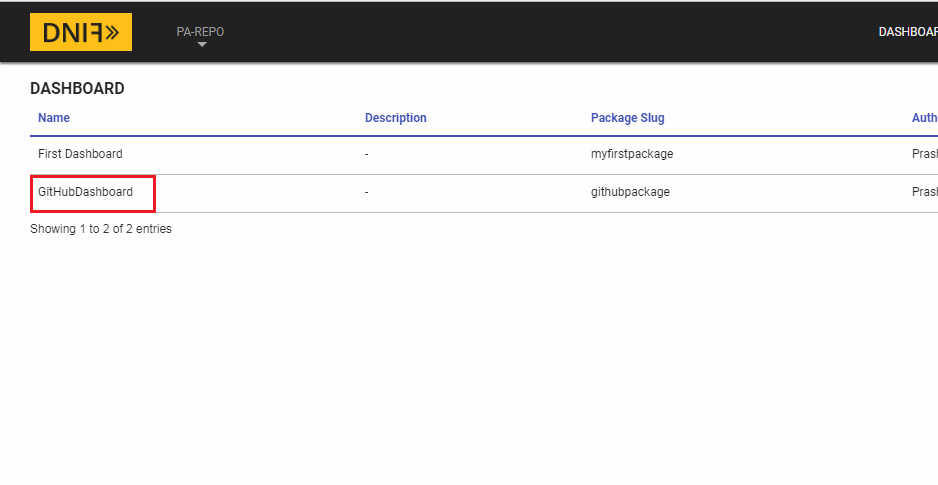


*Fig 20: Adding a Dashboard*

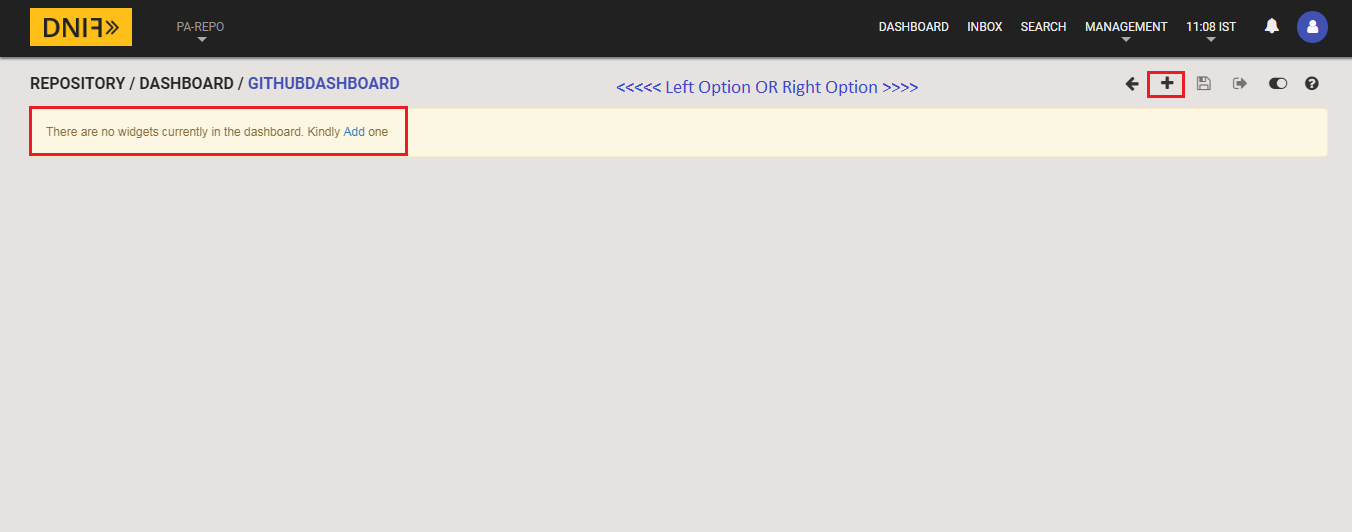


*Fig 21: Creating Dashboard*

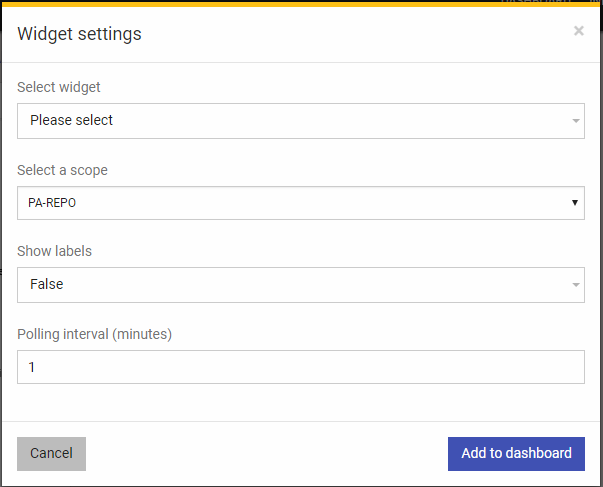
Now that our dashboard is created let us put that widget we created earlier.



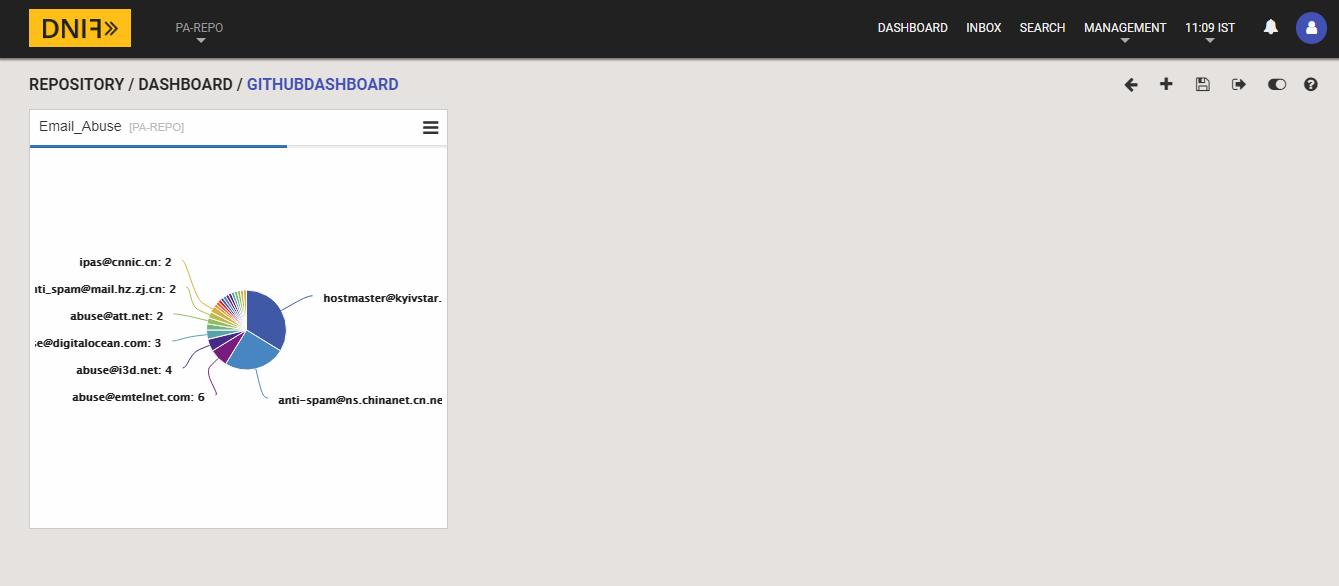
*Fig 22: Our Dashboard*



*Fig 23: Adding Widgets to Dashboard*



*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 rule/ security rule is violated. 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.