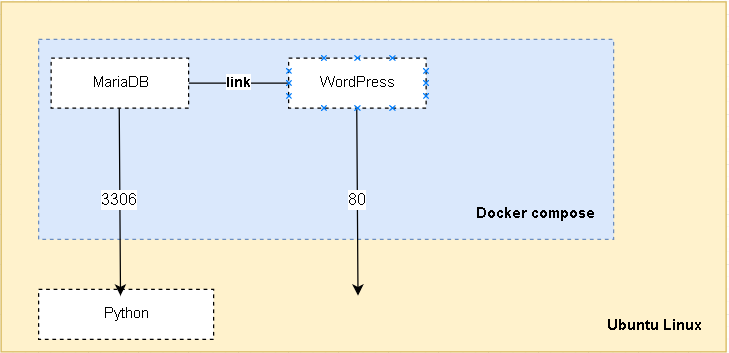
Solution architecture



Installation and Execution details

#### Install Docker Set CE - up the repository

1. Update the apt package index:

$ sudo apt-get update

1. Install packages to allow apt to use a repository over HTTPS:

$ sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

software-properties-common

1. Add Docker’s official GPG key:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

1. Set up the stable repository.

$ sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

1. Install the latest version of Docker CE

$ sudo apt-get install docker-ce

1. Configure Docker to start on boot

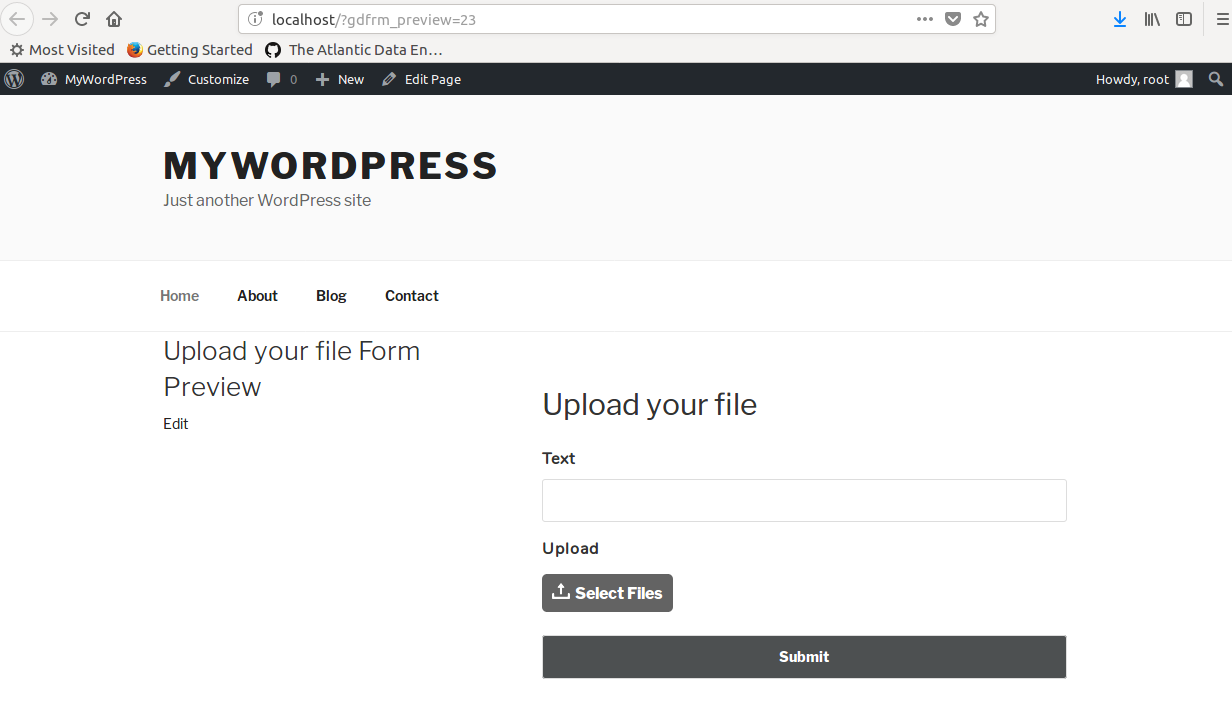
$ sudo systemctl enable docker

#### Install wordpress, mariaDB and python as a docker-compose

1. Install wordpress, mariaDB and python as a docker-compose



1. Install GrandWP Forms from wordpress, plugins > add new > search GrandWP Forms > install
   1. Create a form with a text box, upload file object and submit button.



1. Install Pip on Ubuntu 14.04 LTS

From root user Install:

apt-get -y install python-pip

pip install wheel

pip install pandas

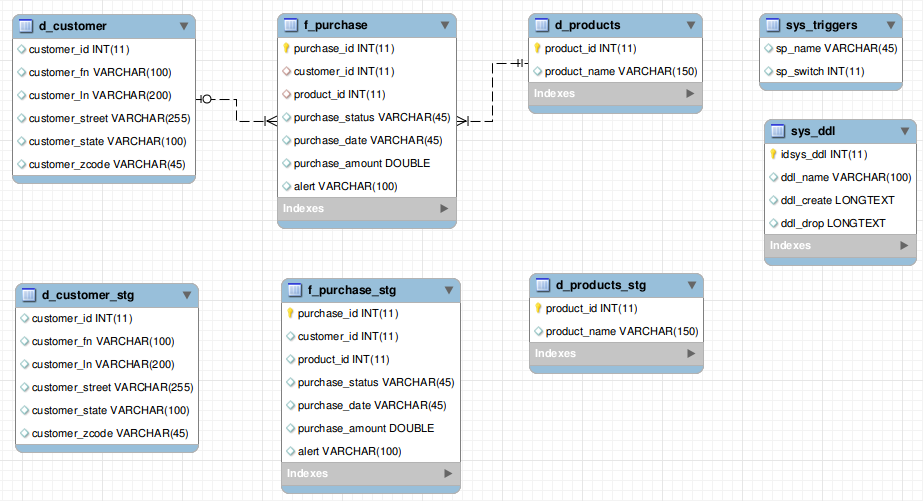
pip install flask\_sqlalchemy

pip install sqlalchemy

apt-get install python3-pymysql

1. Database design





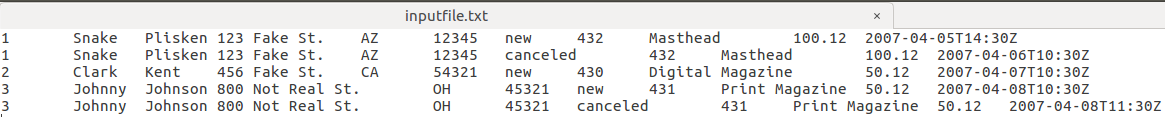
1. Create file **datapipeline\_run.py** to be executed by below command:

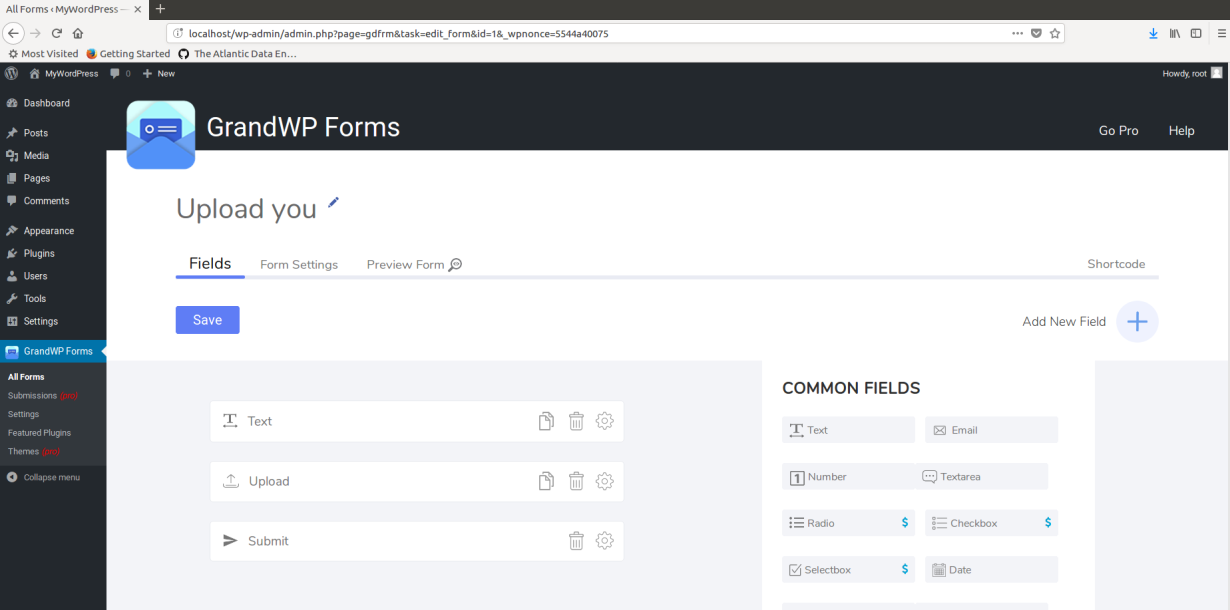
$ python datapipeline\_run.py

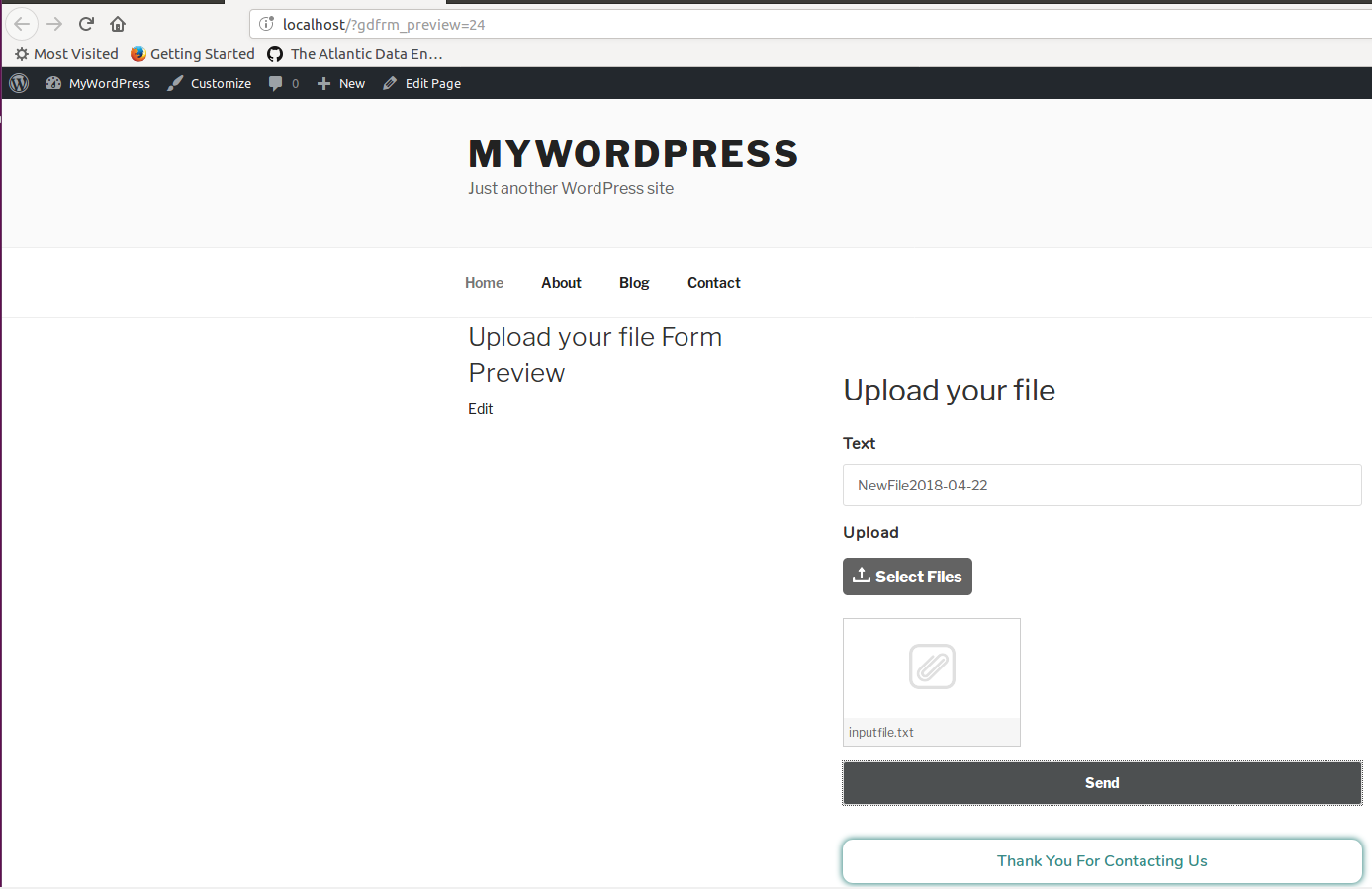


Test

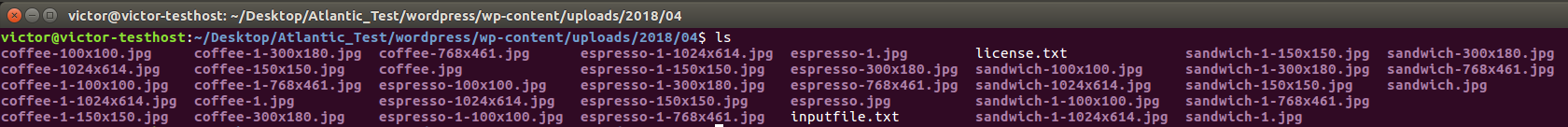
1. Upload inputfile.txt in WordPress - GrandWP Forms







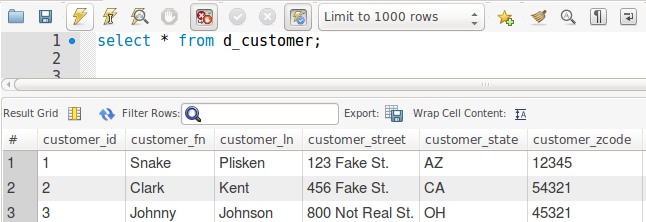
1. Verify the location of the file uploaded by the client

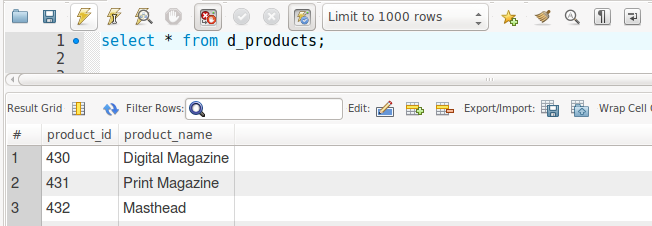


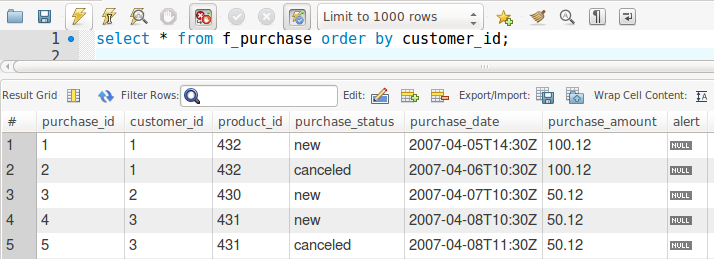
1. Execute the python script datapipeline\_run.py



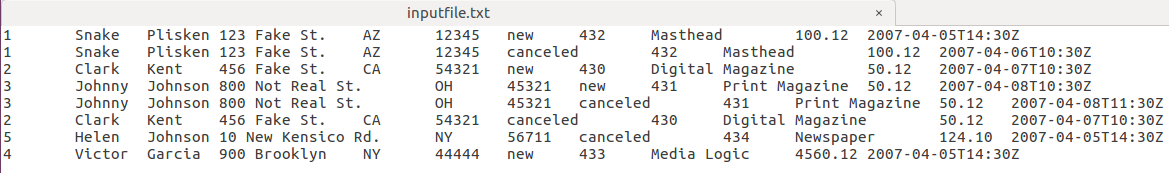
1. Verify the tables in the database



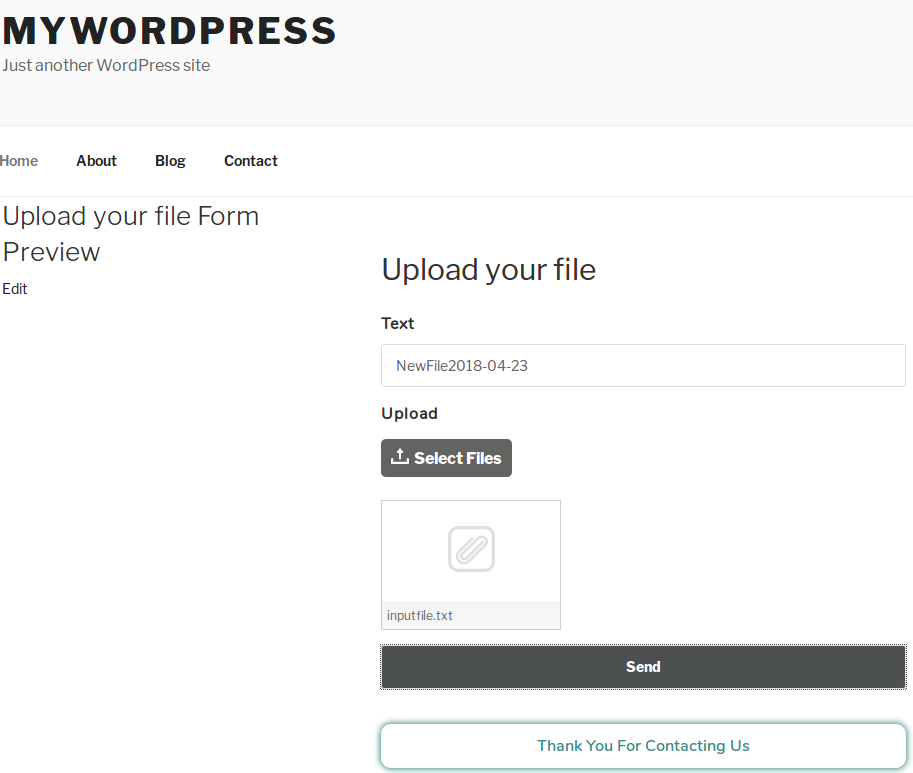




1. Add rows into a new inputfile.txt, including the next cases
   1. Add a row with status “canceled” of existing customer\_id = 2, product\_id = 430
   2. Add a new purchase customer\_id = 5, product\_id = 434 with status “canceled” (This **should show an alert**)
   3. Add a new purchase customer\_id = 4, product\_id = 433



Add a new file from client in WordPress Form

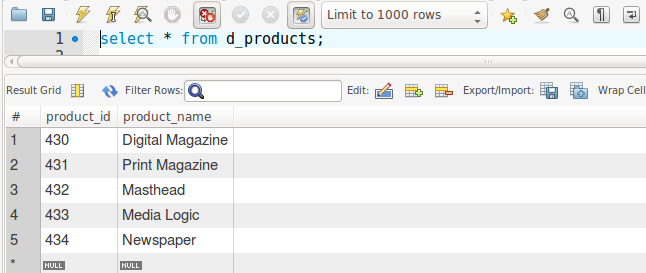


1. Execute the python script datapipeline\_run.py again

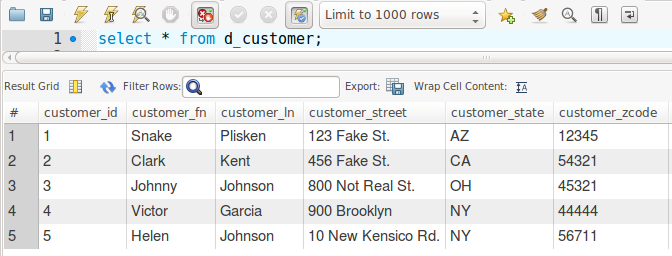


1. Verify the tables in the database

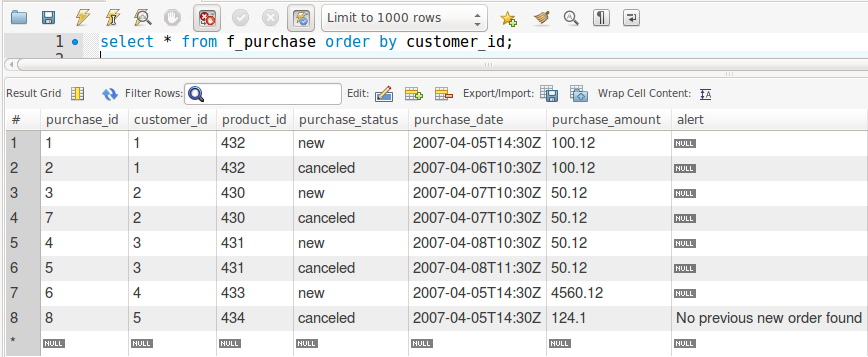
Two new products have been added to the d\_products table



Two new customers have been added to the d\_customer table



* 1. Add a row with status “canceled” of existing customer\_id = 2, product\_id = 430
     1. A new purchase has been added with the status “canceled”
  2. Add a new purchase customer\_id = 5, product\_id = 434 with status “canceled” (This **should show an alert**)
     1. A new purchase has been added with the status “canceled” but, it shows an alert because it has not previous new purchase.
  3. Add a new purchase customer\_id = 4, product\_id = 433
     1. A new purchase has been added with the status “new”



* 1. Add a new purchase customer\_id = 4, product\_id = 433 but changing the address of the customer and status “canceled”.
     1. After running the process again we get a new purchase has been added with the status “canceled” and the customer address has change as well.

