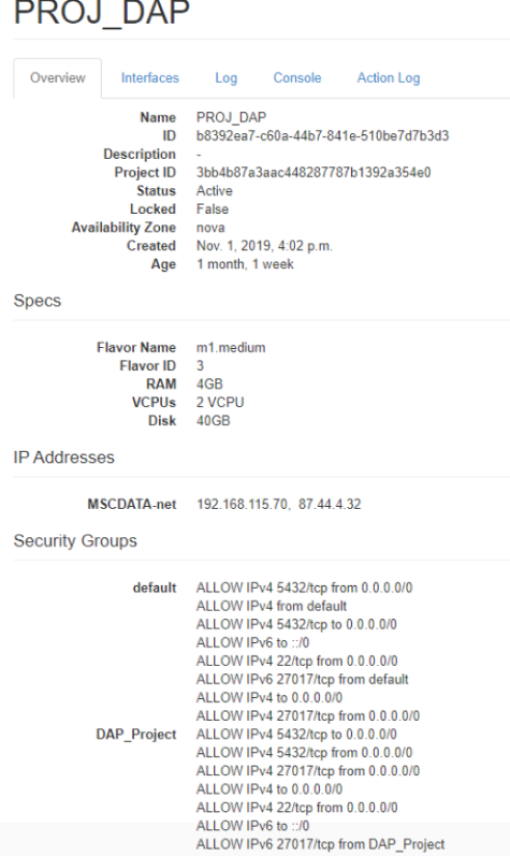
Architecture of Database Analytical Programming

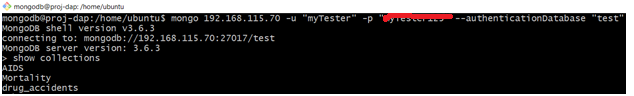
OpenStack, a cloud instance is used as Infrastructure as a Service (IaaS) provisioned for installation of MongoDB, Postgres and Spark. The cloud instance is provided by National College of Ireland. Security is ensured by authenticating the instance through a valid .pem or .ppk file.

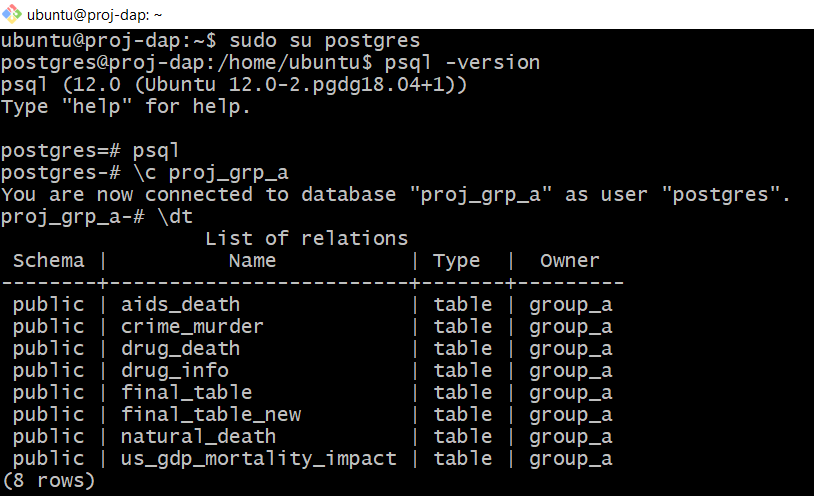
Machine Specification:

* Type – Ubuntu-Bionic 18.04.3 LTS
* RAM - 4 GB
* Virtual CPUs - 2
* Hard disk - 40 GB
* Availability Zone – nova
* Java version 8

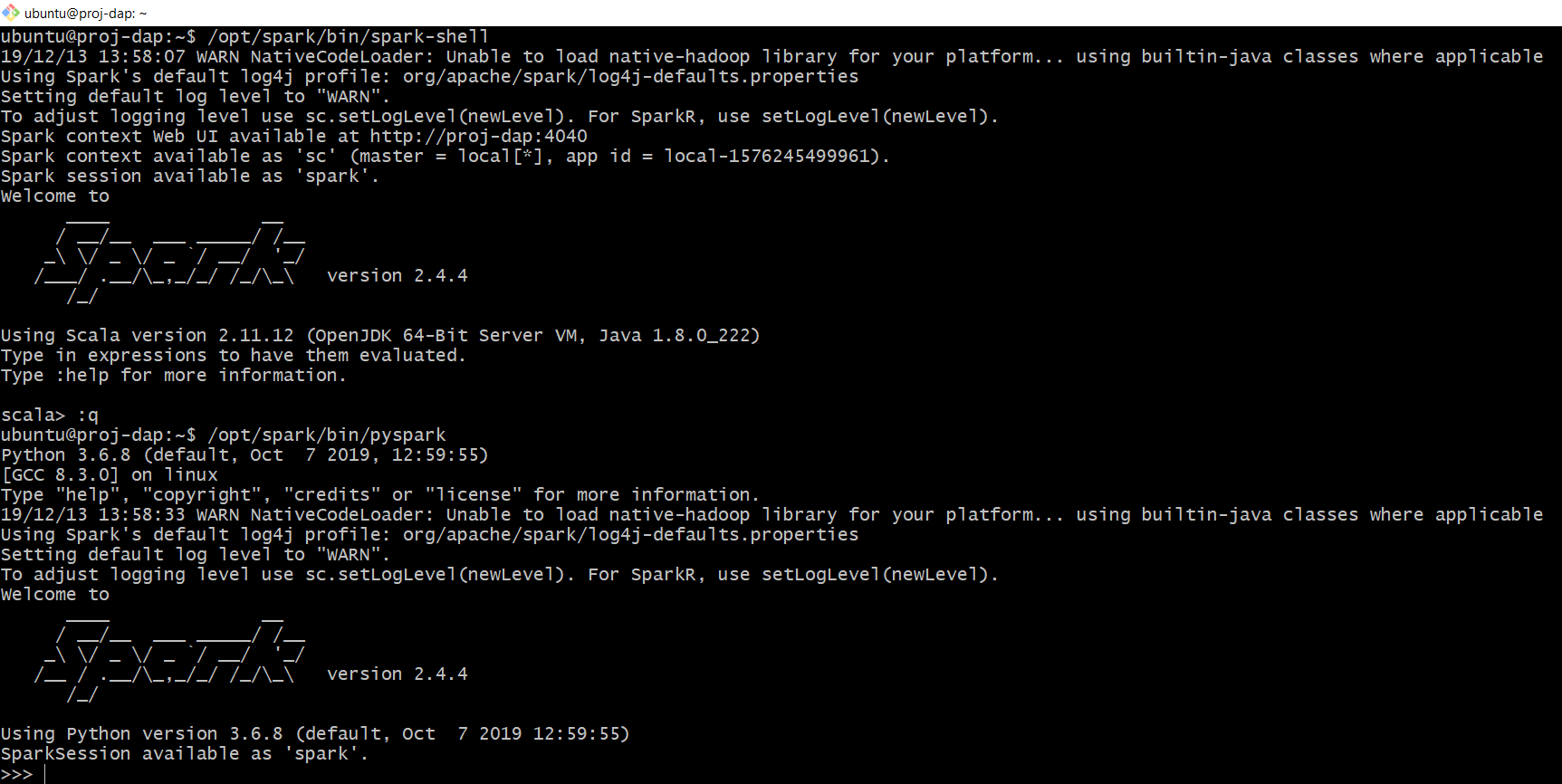


The above figure represents an overview of the machine specification. New security (DAP\_Project) is created for opening the ports 27017 and 5432. 27017 is for MongoDB and 5432 is for Postgres. Two super users **mongodb** and **postgres** were created to access NoSQL and SQL databases respectively.

MongoDB version **3.6.3** is installed and **myTester** user is created and Read-Write permission is provided to access it from python with password authentication. We have created a database **test** and store all our collections.

PostgreSQL version **12.0** is installed and **group\_a** useris created with Read-Write access, and it is accessed from python using password authentication. **Proj\_grp\_a** database is created, and the tables are stored in the database.

Apache Spark version 2.4.4 is also installed in the OpenStack instance. Three key reason to choose Spark are **simplicity, speed and support.** It can be easily integrated with any services and scale the data. Data can be processed in-memory as well as in disk. As we are processing the data in-memory it performs faster than Hadoop MapReduce and other services. It supports language such as Java, Python, Scala and R. Scala version is 2.11.12 and Python version is 3.6.8 is installed in the instance. For our analysis, **Scala** is used for processing the data from postgres using JDBC connection and combine the tables and load back into a new table. **Spark-SQL** function such as filter, join, groupBy, count and sum are used for joining and processing the dataset. The advantage of using Spark is schema-less. It processes the data in-memory and loads data into postgres without specifying its type and there is no need of creating a table, unlike python.



In the firewall settings, we have opened the ports 22, 5432 and 27017 for OpenSSH, PostgreSQL and MongoDB respectively to connect from other machines.

Python is used for Extraction, Transformation and Loading (ETL) the dataset into respective databases. Also pre-processing, cleaning of the dataset is carried out in python’s library such as Pandas, NumPy, etc. Regex, map and lambda functions are also used to transform the datasets. For extracting the API data from websites, we used JSON imports of python.

MongoClient from pyMongo for MongoDB, psycopg2 for postgres is used to connect the databases from python. A user-defined function (UDF) is created for exceptional handling in the code. The tables are loaded to postgres from CSV file using sqlio of Pandas library. The visualizations are performed using Plotly library of python. Various Plotly objects such as express, graph\_objects, chlorepth maps, seaborn chart, heat map, histogram, bubble chart, pie-charts, donut charts, sun-burst chart and widgets drop down boxes are used for interactive visualization.