GraphDB installation using docker on Ubuntu 18.04

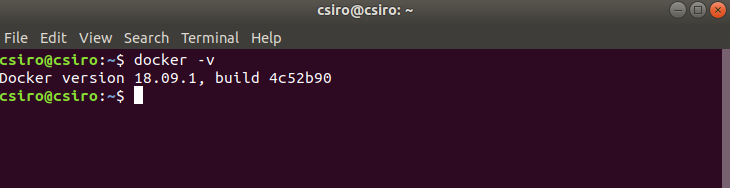
Steps for setting up ubuntu virtual machine:

1.) Install docker using the following link: <https://get.docker.com/> and execute the following commands

$ sudo apt-get update

$ curl -fsSL https://get.docker.com -o get-docker.sh

$ sh get-docker.sh



Link for the docker configuration file for GraphDB installation: <https://github.com/coderdecoder01/csiro-org-history-a/blob/master/Dockerfile>

2.) Running the GraphDB docker image inside docker:

a.) clone the repository using

$ git clone <https://github.com/coderdecoder01/csiro-org-history-a.git>

and change current directory using

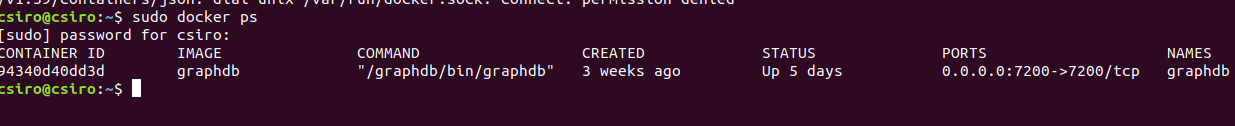
$ cd csiro-org-history-a

b.) Build docker image using docker file with

$ docker build –t graphdb .

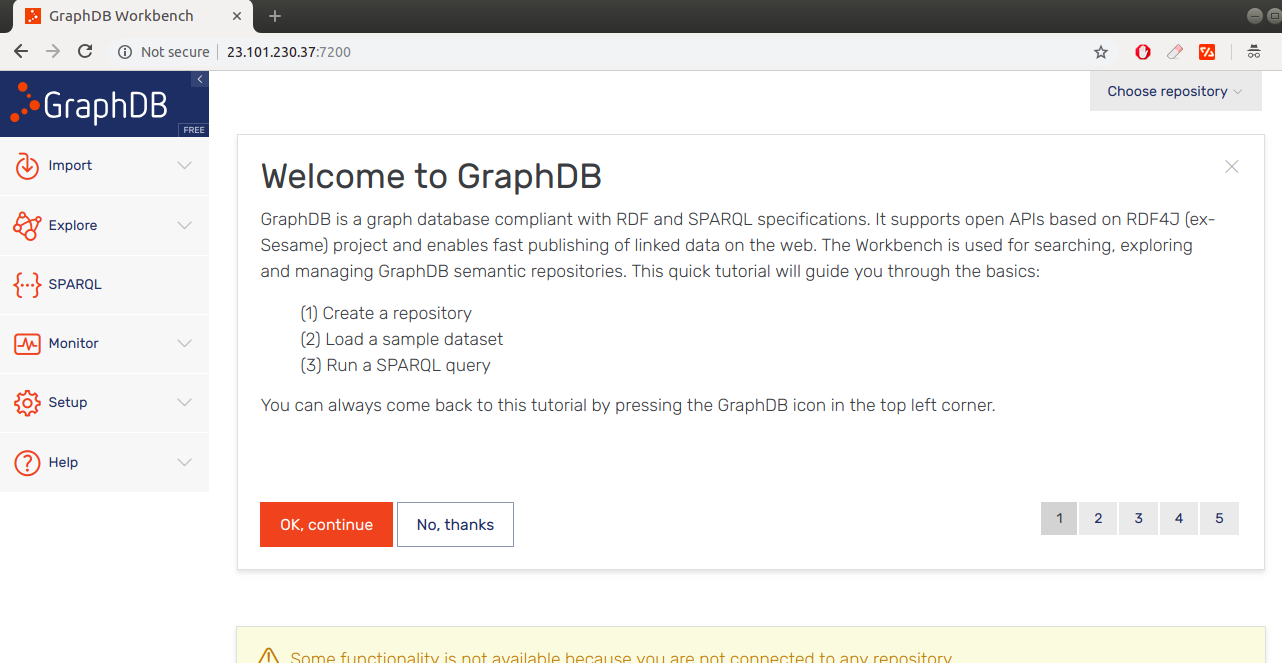
And run it inside docker with following parameters

$ docker run –name graphdb –d –p 7200:7200 graphdb



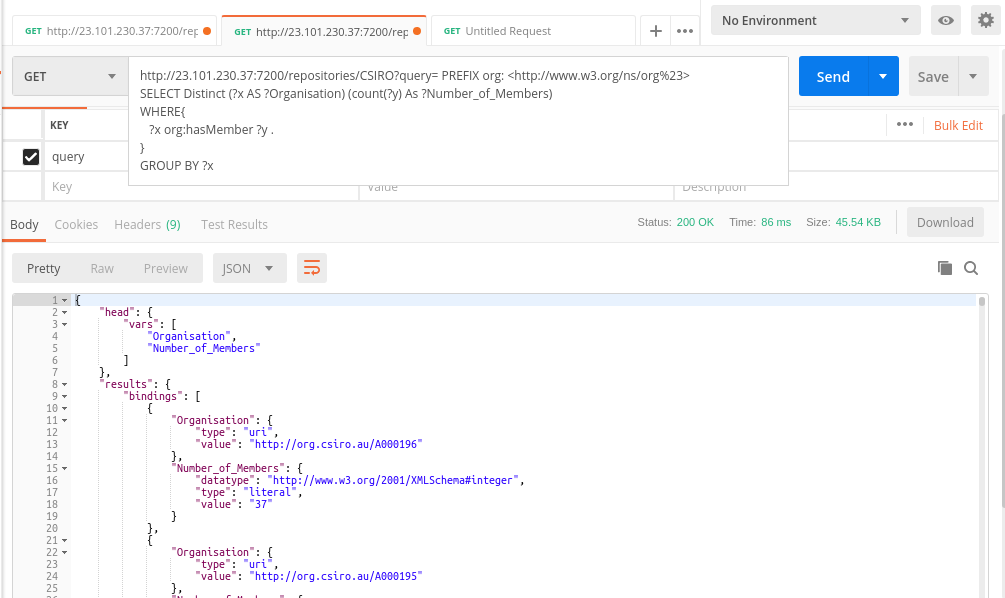
This would make the GraphDB workbench available through <ip address of virtual machine>:7200

In our case VM is hosted at 23.101.230.37, so graphDB work bench can be accessed at http://23.101.230.37:7200



It exposes a SPARQL REST interface to query data on the same link.

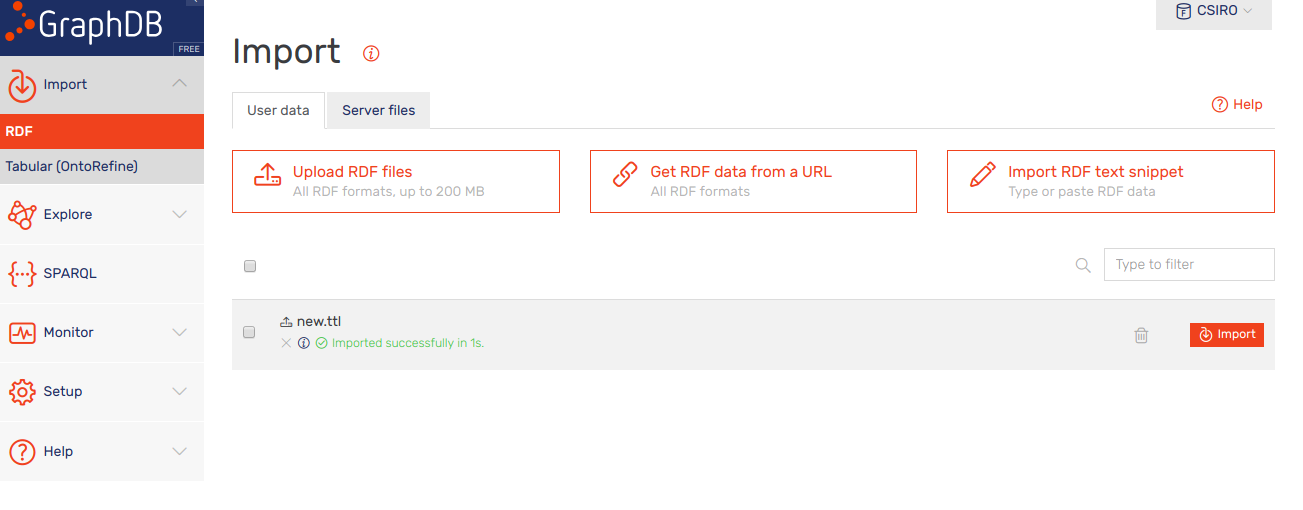
Here is an example showing a REST request done using POSTMAN to the SPARQL interface



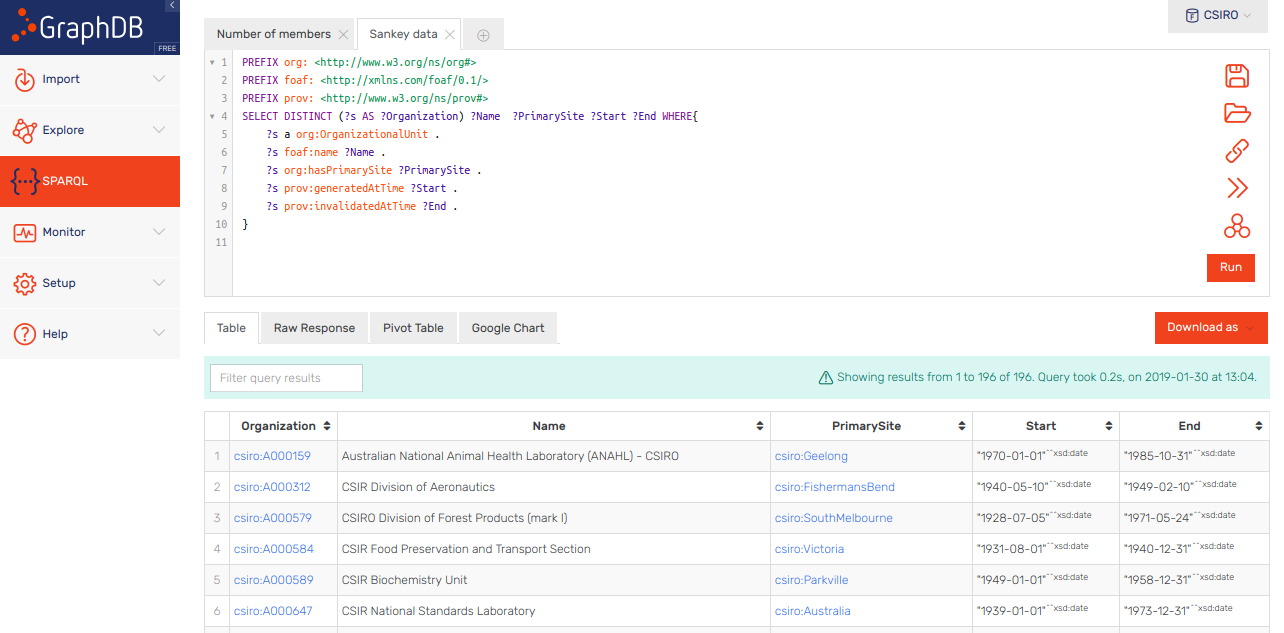
Importing CSIRO data and Querying

GraphDB can readily import rdf files of type .ttl , .owl, .n3 , .rdf, .n3, .nq and zip files as well.

After uploading the file, clicking the import button will import data into the current repository.



A SPARQL query interface is available where data inside the repository can be queried and useful queries can be saved. The result of the query can then be downloaded in formats such as CSV, JSON or XML



Front end Interface

Web interface for querying and visualizing the data is also hosted on the same virtual machine using Nginx and PHP and can be viewed live at <http://23.101.230.37>

Source code for the website is available at https://github.com/coderdecoder01/csiro-org-history-a

