Arango DB - Example

On Windows -

Download https://download.arangodb.com/arangodb33/Windows7/x86_64/ArangoDB3-3.3.5-1_win64.zip and unzip, locate arangod.exe and arangoimp.exe

1 Kick arangod.exe to launch graph db

2 Import CSV - VPN

```
D:\>arangoimp --file sample.csv --type csv --collection vpns
--create-collection true
Please specify a password:
Connected to ArangoDB 'http+tcp://127.0.0.1:8529', version 3.3.5,
database: '_system', username: 'root'
----------------------------------------
database:               _system
collection:             vpns
create:                 yes
source filename:        sample.csv
file type:              csv
quote:                  "
separator:
threads:                2
connect timeout:        5
request timeout:        1200
----------------------------------------
Starting CSV import...
2018-04-03T07:29:28Z [5980] INFO processed 32768 bytes (3%) of input
file
2018-04-03T07:29:28Z [5980] INFO processed 65536 bytes (15%) of input
file
2018-04-03T07:29:28Z [5980] INFO processed 98304 bytes (27%) of input
file
2018-04-03T07:29:28Z [5980] INFO processed 131072 bytes (40%) of input
file
2018-04-03T07:29:28Z [5980] INFO processed 163840 bytes (52%) of input
file
2018-04-03T07:29:28Z [5980] INFO processed 196608 bytes (65%) of input
file
2018-04-03T07:29:28Z [5980] INFO processed 229376 bytes (77%) of input
file
2018-04-03T07:29:28Z [5980] INFO processed 262144 bytes (90%) of input
file

created:          1000
warnings/errors:  0
updated/replaced: 0
ignored:          0
lines read:       1003

D:\>
```

3 Import - Lenel data

```
D:\>arangoimp --file woodcrest.csv --type csv --collection lenels
--create-collection true
```

```
Please specify a password:
Connected to ArangoDB 'http+tcp://127.0.0.1:8529', version 3.3.5,
database: '_system', username: 'root'
----------------------------------------
database:               _system
collection:             lenels
create:                 yes
source filename:        woodcrest.csv
file type:              csv
quote:                  "
separator:
threads:                2
connect timeout:        5
request timeout:        1200
----------------------------------------
Starting CSV import...
2018-04-03T07:30:31Z [7396] INFO processed 65536 bytes (3%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 98304 bytes (8%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 131072 bytes (11%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 163840 bytes (14%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 196608 bytes (17%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 229376 bytes (20%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 262144 bytes (23%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 294912 bytes (26%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 327680 bytes (29%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 360448 bytes (32%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 393216 bytes (35%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 425984 bytes (38%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 458752 bytes (41%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 491520 bytes (44%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 524288 bytes (47%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 557056 bytes (50%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 589824 bytes (53%) of input
file
```

```
2018-04-03T07:30:31Z [7396] INFO processed 622592 bytes (56%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 655360 bytes (59%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 688128 bytes (62%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 753664 bytes (65%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 786432 bytes (70%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 819200 bytes (73%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 851968 bytes (76%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 884736 bytes (79%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 917504 bytes (82%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 950272 bytes (85%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 983040 bytes (88%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 1015808 bytes (91%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 1048576 bytes (94%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 1081344 bytes (97%) of input
file
2018-04-03T07:30:31Z [7396] INFO processed 1109312 bytes (100%) of input
file

created:          5019
warnings/errors:  0
updated/replaced: 0
ignored:          0
```

```
lines read:       5021

D:\>
```

4. http://localhost:8529 shows UI - username root, password blank, database *system (dev lab password = 14edcd1c2b7ff3b42823eac1eeb5d456*)

(Note - careful on uploading csv, csv header expected no space for AQL to work) - AQL is similar to SQL in its defination but different syntax - few ex -

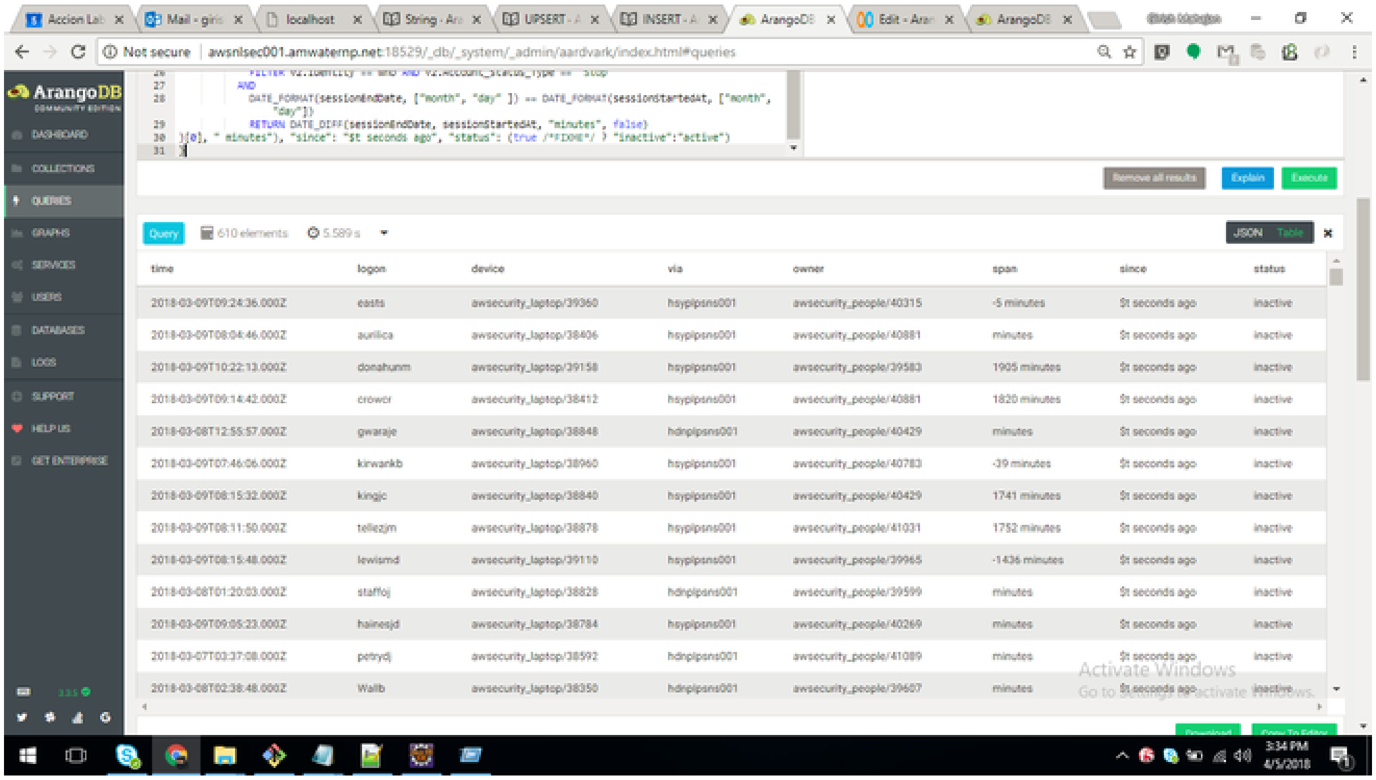
#1

```
FOR v in vpns
COLLECT Server = v.Server WITH COUNT INTO length
RETURN {
    "Server" : Server,
    "count" : length
  }


=> Answer for example

[
  {
    "Server": "hdnplpsns001",
    "count": 435
  },
  {
    "Server": "hsyplpsns001",
    "count": 565
  }
]
```

Example #2 - with inner joins and date functions etc, customized json tuples

```
FOR v in awsecurity_vpns
FILTER v.Account_Status_Type == 'Start'
LET sessionCreationTime = DATE_ISO8601(DATE_FORMAT(SUBSTITUTE(CONCAT(
v.Date," ",v.Time), [ "AM", "PM"])   /*FIXME*/,
"%m/%d/%yyyy,%h:%ii:%ss"))
INSERT { "time": sessionCreationTime, "logon": v.Identity, "device":
CONCAT ("awsecurity_laptop/", (
    FOR a in awsecurity_laptops
    FILTER a.id == v.UPN
    RETURN DISTINCT a._key
)[0]), "via": v.Server, "owner": CONCAT ("awsecurity_people/", (
    FOR p in awsecurity_peoples
    FILTER v.First == p.first
    RETURN DISTINCT p._key
)[0]), "span": CONCAT ((
    LET sessionStartedAt = (
        FOR v2 in awsecurity_vpns
        FILTER v2.Identity == v.Identity AND v2.Account_Status_Type ==
'Start'
        COLLECT sessionDate= v2.Date, sessionTime = v2.Time WITH COUNT
into length
        LET anydate = SUBSTITUTE( CONCAT (sessionDate, " ",
sessionTime), [ "AM", "PM"])   /*FIXME*/
        RETURN DATE_ISO8601(DATE_FORMAT(anydate,
"%m/%d/%yyyy,%h:%ii:%ss"))
    )[0]
    FOR v1 in awsecurity_vpns
    FILTER v1.Identity == v.Identity AND v1.Account_Status_Type ==
'Stop'
        COLLECT who= v1.Identity, sessionDate= v1.Date, sessionTime =
v1.Time WITH COUNT into length
        LET anydate = SUBSTITUTE( CONCAT (sessionDate, " ",
sessionTime), [ "AM", "PM"], "")
        LET sessionEndDate = DATE_ISO8601(DATE_FORMAT(anydate,
"%m/%d/%yyyy,%h:%ii:%ss"))
            FOR v2 in awsecurity_vpns
            FILTER v2.Identity == who AND v2.Account_Status_Type ==
'Stop'
          AND
            DATE_FORMAT(sessionEndDate, ["month", "day" ]) ==
DATE_FORMAT(sessionStartedAt, ["month", "day"])
            RETURN DATE_DIFF(sessionEndDate, sessionStartedAt,
"minutes", false)
)[0], " minutes"), "since": "$t seconds ago", "status": (true /*FIXME*/
? "inactive":"active")
} INTO awsecurity_sessions
```

Above returns like =>



awsnlsec001.amwatemp.net:18529/_db/_system/_admin/aardvark/index.html#queries

```
            FILTER v2.identity == [illegible] AND v2.Account_status_type == "Stop"
          AND
            DATE_FORMAT(sessionEndDate, ["month", "day" ]) == DATE_FORMAT(sessionStartedAt, ["month",
                "day"])
            RETURN DATE_DIFF(sessionEndDate, sessionStartedAt, "minutes", false)
})[0], " minutes"), "since": "$t seconds ago", "status": (true /*FIXME*/ ? "inactive":"active")
}
```

610 elements 5.589 s

time	logon	device	via	owner	span	since	status
2018-03-09T09:24:36.000Z	eastis	awsecurity_laptop/39360	hsyplpsns001	awsecurity_people/40315	-5 minutes	$t seconds ago	inactive
2018-03-09T08:04:46.000Z	aurilica	awsecurity_laptop/38406	hsyplpsns001	awsecurity_people/40881	minutes	$t seconds ago	inactive
2018-03-09T10:22:13.000Z	donahunm	awsecurity_laptop/39158	hsyplpsns001	awsecurity_people/39583	1905 minutes	$t seconds ago	inactive
2018-03-09T09:14:42.000Z	crowcr	awsecurity_laptop/38412	hsyplpsns001	awsecurity_people/40881	1820 minutes	$t seconds ago	inactive
2018-03-08T12:55:57.000Z	gwaraje	awsecurity_laptop/38848	hdnplpsns001	awsecurity_people/40429	minutes	$t seconds ago	inactive
2018-03-09T07:46:06.000Z	kirwankb	awsecurity_laptop/38960	hsyplpsns001	awsecurity_people/40783	-39 minutes	$t seconds ago	inactive
2018-03-09T08:15:32.000Z	kingjc	awsecurity_laptop/38840	hsyplpsns001	awsecurity_people/40429	1741 minutes	$t seconds ago	inactive
2018-03-09T08:11:50.000Z	tellezjm	awsecurity_laptop/38878	hsyplpsns001	awsecurity_people/41031	1752 minutes	$t seconds ago	inactive
2018-03-09T08:15:48.000Z	lewismd	awsecurity_laptop/39110	hsyplpsns001	awsecurity_people/39965	-1436 minutes	$t seconds ago	inactive
2018-03-08T01:20:03.000Z	staffoj	awsecurity_laptop/38828	hdnplpsns001	awsecurity_people/39599	minutes	$t seconds ago	inactive
2018-03-09T09:05:23.000Z	hainesjd	awsecurity_laptop/38784	hsyplpsns001	awsecurity_people/40269	minutes	$t seconds ago	inactive
2018-03-07T03:37:08.000Z	petrydj	awsecurity_laptop/38592	hdnplpsns001	awsecurity_people/41089	minutes	$t seconds ago	inactive
2018-03-08T02:38:48.000Z	Wallb	awsecurity_laptop/38350	hdnplpsns001	awsecurity_people/39607	minutes	$t seconds ago	inactive

Installing on Linux -

Installation - $ sudo rpm -ivh arangodb3-3.3.5-1.x86_64.rpm

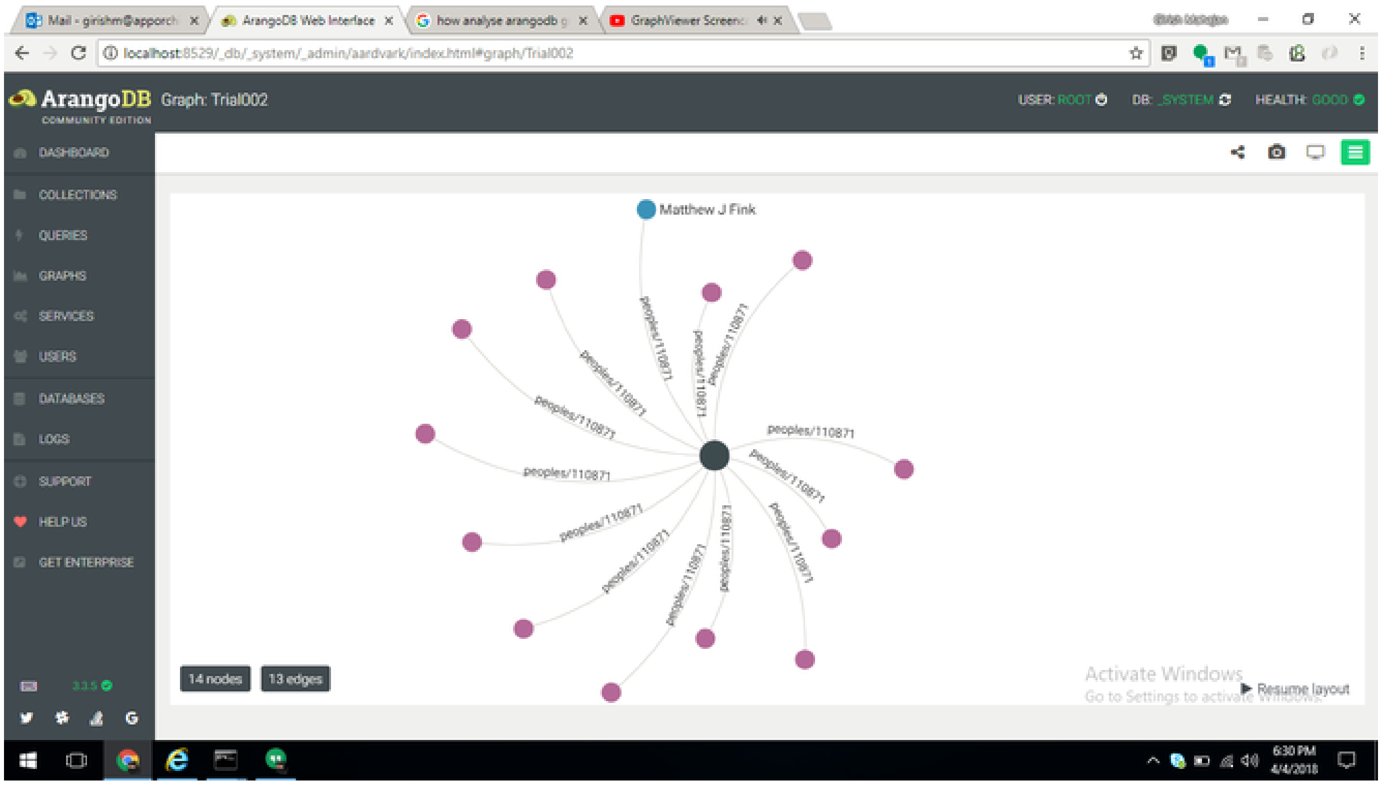
Security Dev Lab // password == 14edcd1c2b7ff3b42823eac1eeb5d456

Apart from manipulating data scnearios, stumble upon a fact that arango UI (creds - root // 14edcd1c2b7ff3b42823eac1eeb5d456) can also provides quick REST ful service which exposes cooked repositories and UI app as followings - (section = Services)

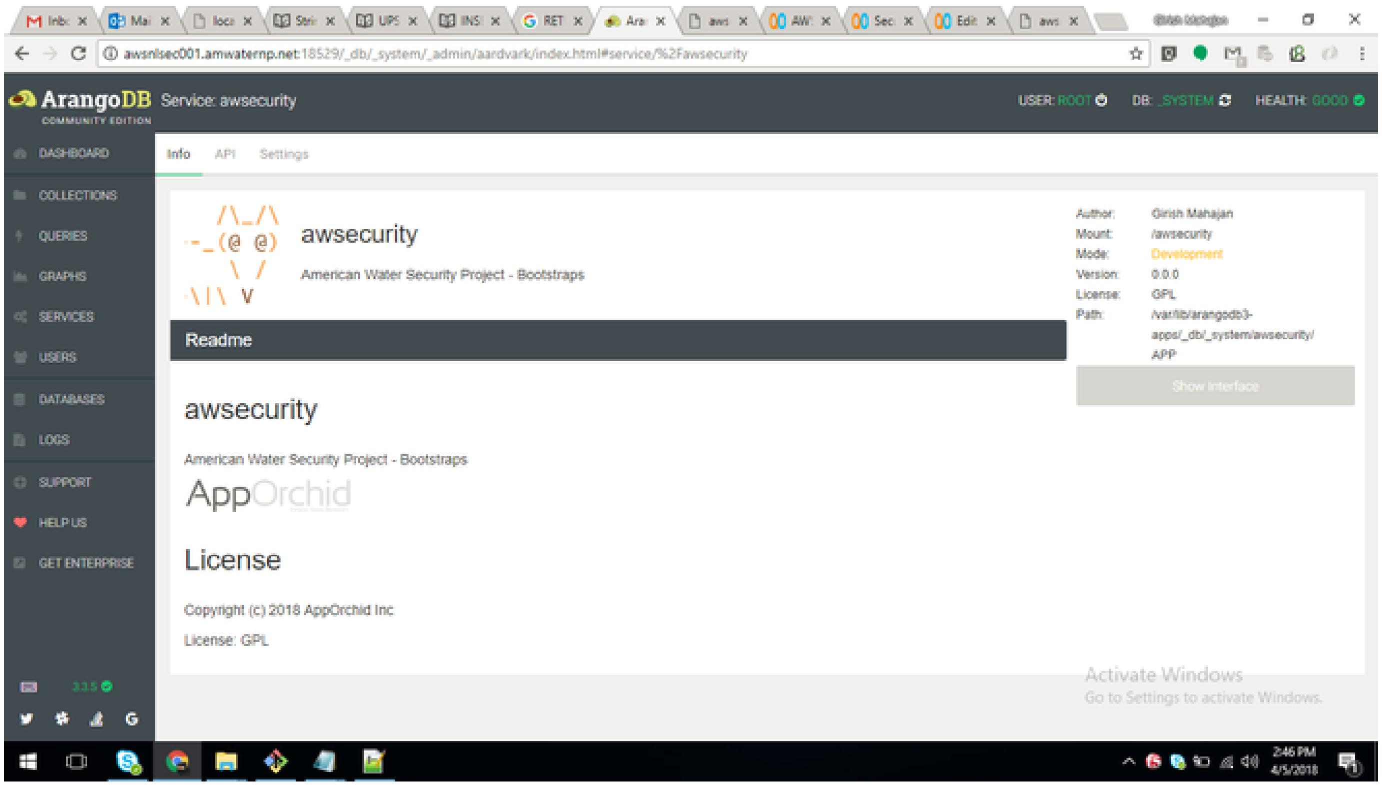
App - Arango awsecurity restful front-end app

Sample swagger json - awsecurity => swagger-awsecurity.json

Few glimpse - (graph where one persons N relation with sessions and 1 laptop can seen as below) -



Embedded REST service could use based on foxx framework -



awsecurity

American Water Security Project - Bootstraps

Author:	Girish Mahajan
Mount:	/awsecurity
Mode:	Development
Version:	0.0.0
License:	GPL
Path:	/var/lib/arangodb3-apps/_db/_system/awsecurity/APP

Readme

awsecurity

American Water Security Project - Bootstraps

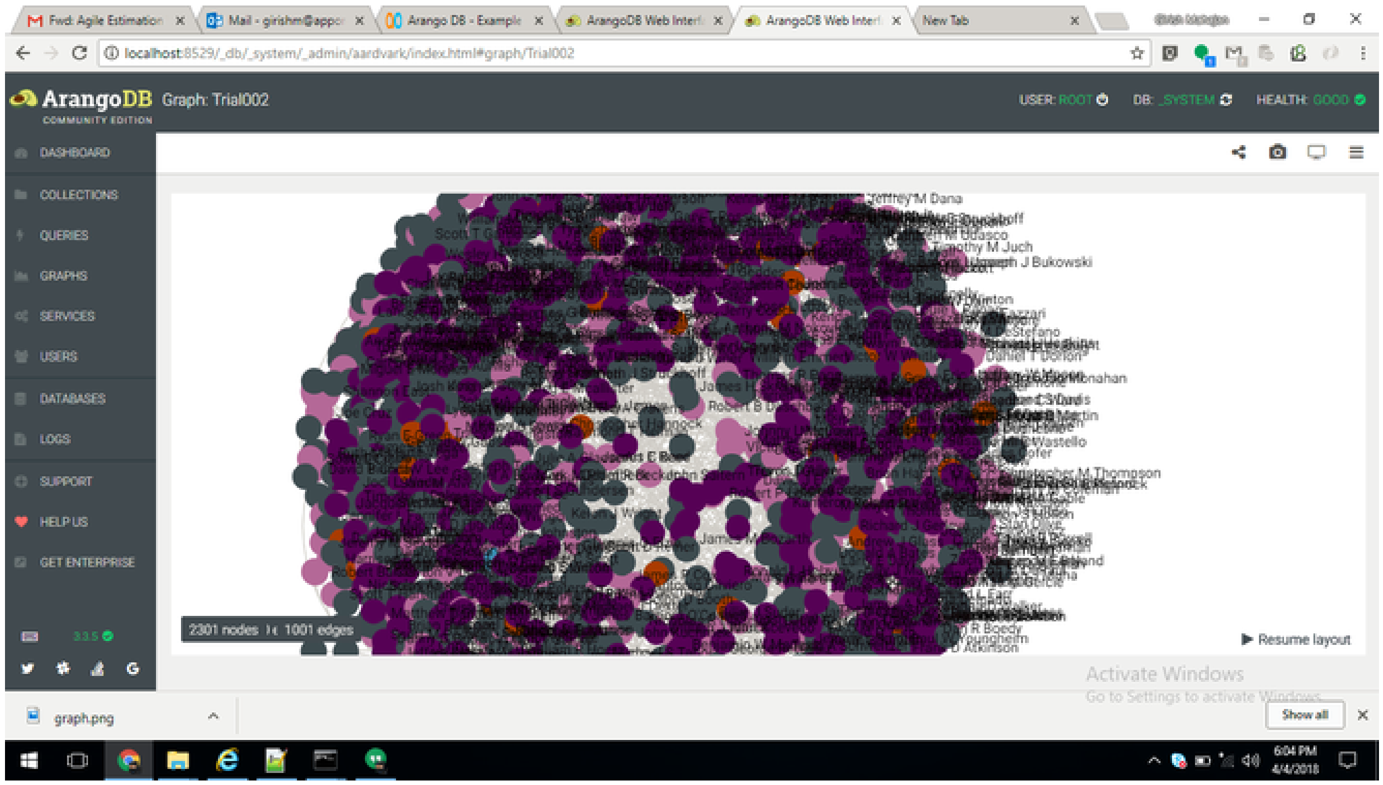
AppOrchid

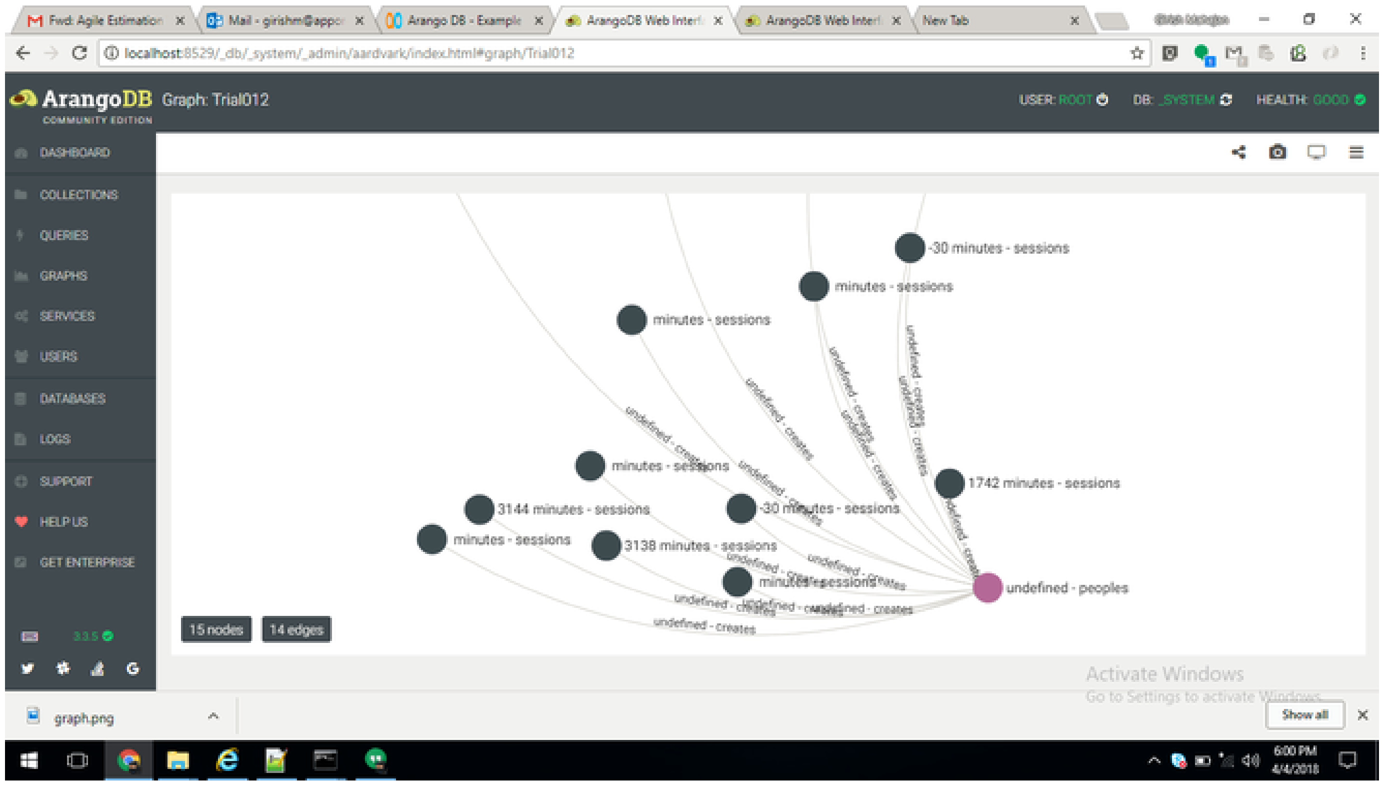
License

Copyright (c) 2018 AppOrchid Inc

License: GPL

Another graph - Can't make a meaning out of this though - but its a bursting all models in one -





Below collections (worked out in attached queries-_system-root_arangodb.json) for modelling data - from source lenels and vpns set. So far -

These are actor we're considering with the data modelling so far -

actors	detail
peoples	knowldge of employees in an organization
laptops	knowledge of devices / laptops / server credentials / vpn keys/ badges given for ^^people etc

locations	knowledge of geo locations for organization
premises	knowledge of premises inherits form ^^location
badging_stations	knowldge of badging stations and door infos
sessions	knowledge of nothing but vpn sessions or net surf sessions or badging sessions we can identify by color let say
vpn_servers	knowledge of intra servers deploys in organization as cyber entry - ie. above output, inherits from ^^sessions
are_in	Edge relation
creates	Edge relation
owns	Edge relation

In order to getting started a fresh, we need to import a source data into arangodb using arangoimp utility like above and then create collection names and use AQL to form a json structure to populate rest of documents and edges.

Sample bash function to wipe clean slate back in arango - Click here to expand...

```
# wipe out collections and recreate, except source lenels & vpns data keeping untouched -


function truncate_arangodbs {

for collection in vpn_servers peoples premises sessions badging_stations laptops locations; do curl -u "root:14edcd1c2b7ff3b42823eac1eeb5d456" -X DELETE http://localhost:8529/_api/collection/awsecurity_$collection; done
for collection in vpn_servers peoples premises sessions badging_stations laptops locations; do echo "{ \"name\": \"$(echo "awsecurity_$collection")\" }" > $collection.txt; done
for collection in vpn_servers peoples premises sessions badging_stations laptops locations; do curl -u "root:14edcd1c2b7ff3b42823eac1eeb5d456" -XPOST --data @$collection.txt http://localhost:8529/_api/collection ; done
for collection in vpn_servers peoples premises sessions badging_stations laptops locations; do rm -rf $collection.txt; done


for collection in creates owns are_in; do curl -u "root:14edcd1c2b7ff3b42823eac1eeb5d456" -X DELETE http://localhost:8529/_api/collection/awsecurity_$collection; done
for collection in creates owns are_in; do echo "{ \"name\": \"$(echo "awsecurity_$collection")\", \"type\": 3 }" > $collection.txt; done
for collection in creates owns are_in; do curl -u "root:14edcd1c2b7ff3b42823eac1eeb5d456" -XPOST --data @$collection.txt http://localhost:8529/_api/collection ; done
for collection in creates owns are_in; do rm -rf $collection.txt; done


# unload few

for collection in premises locations; do curl -u "root:14edcd1c2b7ff3b42823eac1eeb5d456" -XPUT http://localhost:8529/_api/collection/awsecurity_$collection/unload; done

}
```

Step 1 - Follow Arango DB - Example Linux arangodbimp statements to import raw records (wiz creates two of collections, named lenels & vpns), then
Step 2 - Run above function to (re)create stuff needed in raw in bash

```
$ truncate_arangodbs
```

Step 3 - Import a json query from attached confluence page
Step 4 - Login to UI and run step by step each queries to see the data modelling out of Step 1 raw data

Step 5 - data is ready

REST Endpoints are exposes here -

http://awsnlsec001.amwaternp.net/awsecurity/peoples

http://awsnlsec001.amwaternp.net/awsecurity/sessions

http://awsnlsec001.amwaternp.net/awsecurity/badging_stations

http://awsnlsec001.amwaternp.net/awsecurity/owns

http://awsnlsec001.amwaternp.net/awsecurity/vpn_servers

http://awsnlsec001.amwaternp.net/awsecurity/laptops

Raw log - http://awsnlsec001.amwaternp.net/awsecurity/vpns

Raw log - http://awsnlsec001.amwaternp.net/awsecurity/lenels

Note - On java - https://github.com/arangodb/arangodb-java-driver driver may use with current codes.