Guardium Administration

Second Installment - Part II -

Maintaining a Balanced Environment

Handling UnBalanced Collectors

#3.2 : How to Detect Contributing Servers/Agent

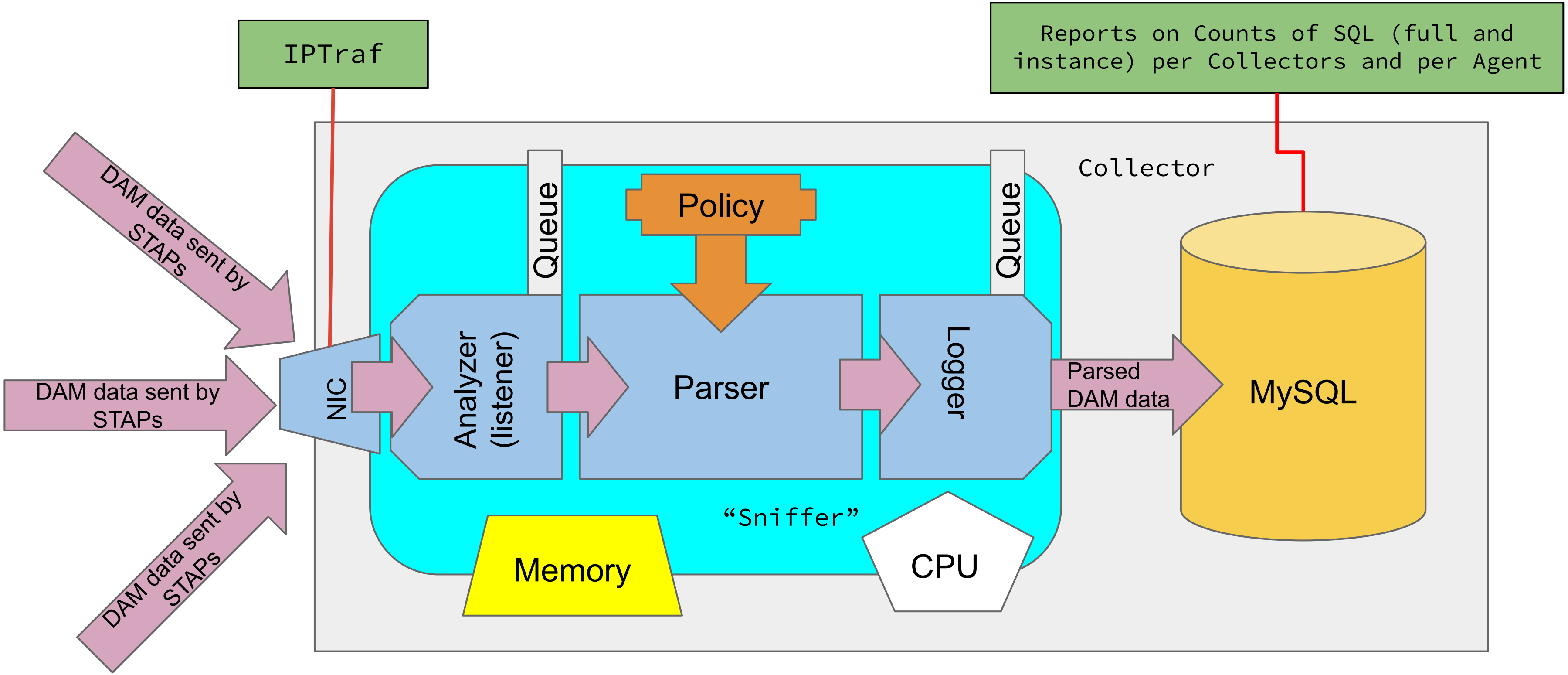
# **How to assess** the contribution of each Agent ?

- **Unfortunately** the BUM gives ONLY global statistics -
- **Only 2 places : NIC and MySQL (see diagram)**
  - At the NIC level : IPTraf from CLI
  - At the MySQL Level : Statistical Reports counting the number of SQLs

# **What to do ?**

- Do nothing is **rarely an option** in this case
- **Re-Assign** some Agents to underloaded collectors to reach a more balanced environment
- **Potentially** Activate the **E**nterprise **L**oad **B**alancing, but be careful, this too requires close monitoring and speedy reaction in case of trouble

# 3.2 : Assessing Contributing Agents (Not in the BUM)

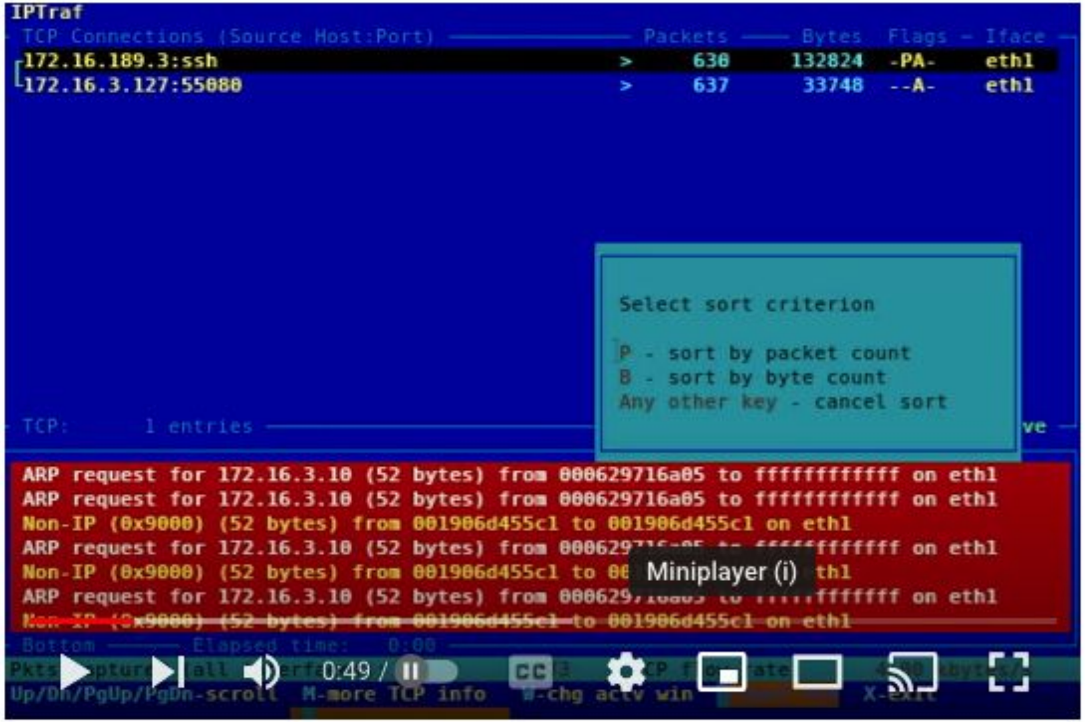


Author: Frederic Petit

Option #1 : IPTraf

In CLI, just type in :

```
>iptraf
```



```
IPTraf
 TCP Connections (Source Host:Port) ------------ Packets ---- Bytes  Flags - Iface -
┌172.16.189.3:ssh                          >       638    132824  -PA-    eth1
└172.16.3.127:55080                        >       637     33748  --A-    eth1

                                              Select sort criterion

                                              P - sort by packet count
                                              B - sort by byte count
                                              Any other key - cancel sort
 TCP:       1 entries ------------------                                      ve -
 ARP request for 172.16.3.10 (52 bytes) from 000629716a05 to ffffffffffff on eth1
 ARP request for 172.16.3.10 (52 bytes) from 000629716a05 to ffffffffffff on eth1
 Non-IP (0x9000) (52 bytes) from 001906d455c1 to 001906d455c1 on eth1
 ARP request for 172.16.3.10 (52 bytes) from 00062971      ffffffffff on eth1
 Non-IP (0x9000) (52 bytes) from 001906d455c1 to 00                 th1
 ARP request for 172.16.3.10 (52 bytes) from 000629            ffffffff on eth1
 Non-IP (0x9000) (52 bytes) from 001906d455c1 to 001906d455c1 on eth1
 Bottom ------ Elapsed time:   0:00
 Up/Dn/PgUp/PgDn-scroll  M-more TCP info   W-chg actv win        X-exit
```

Excellent Tutorial video on IPTraf

https://youtu.be/D91hg8sEcOw

Option #2 : SQLs Recorded into MySQL - By Product of DAM

This is the tricky part :

- Requires having centralized/concentrated the DAM data into an ELK instance
- Or you have the DAM Traffic on the Collector only

Our Solution :

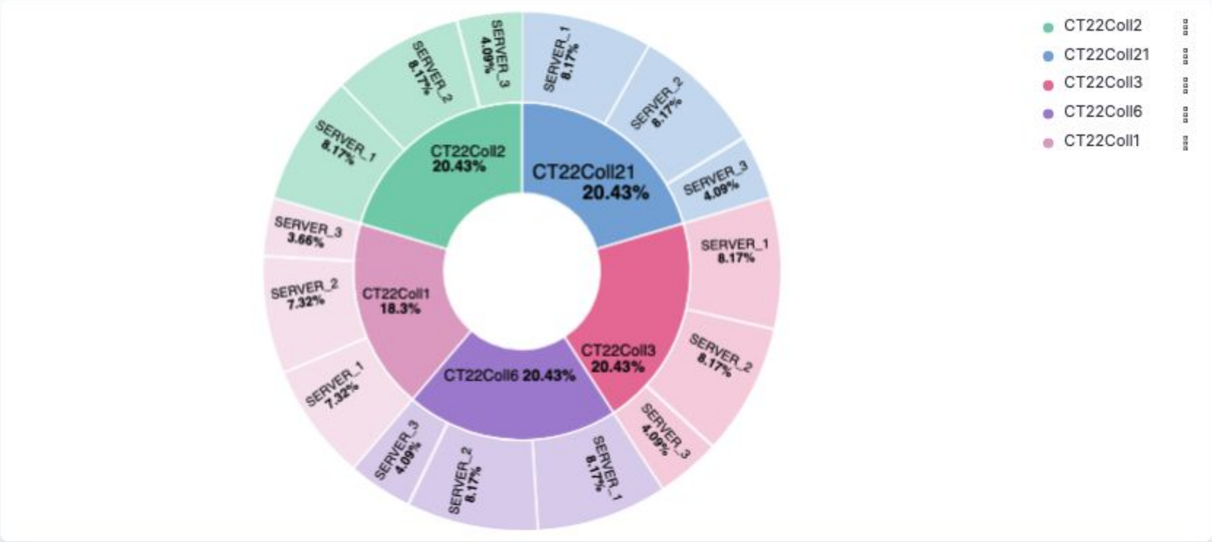
- Export the DAM Traffic and send them to a Central ELK instance thru the **CT22T Enrichment process (next slides)**

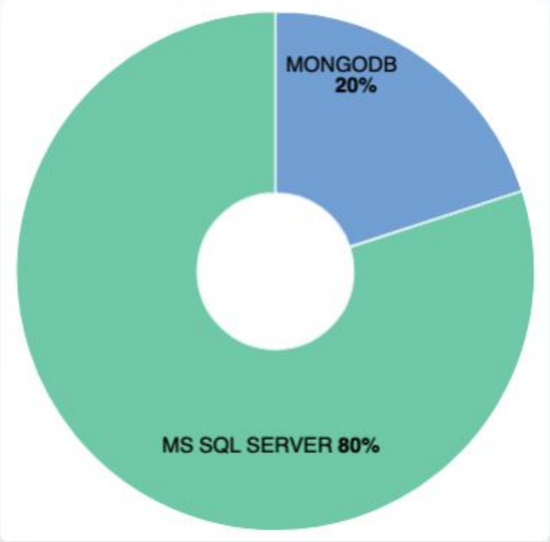
If on Collector’s MySQL only:

- Write a report mimicking our CT22T solution, but it won’t get the same display

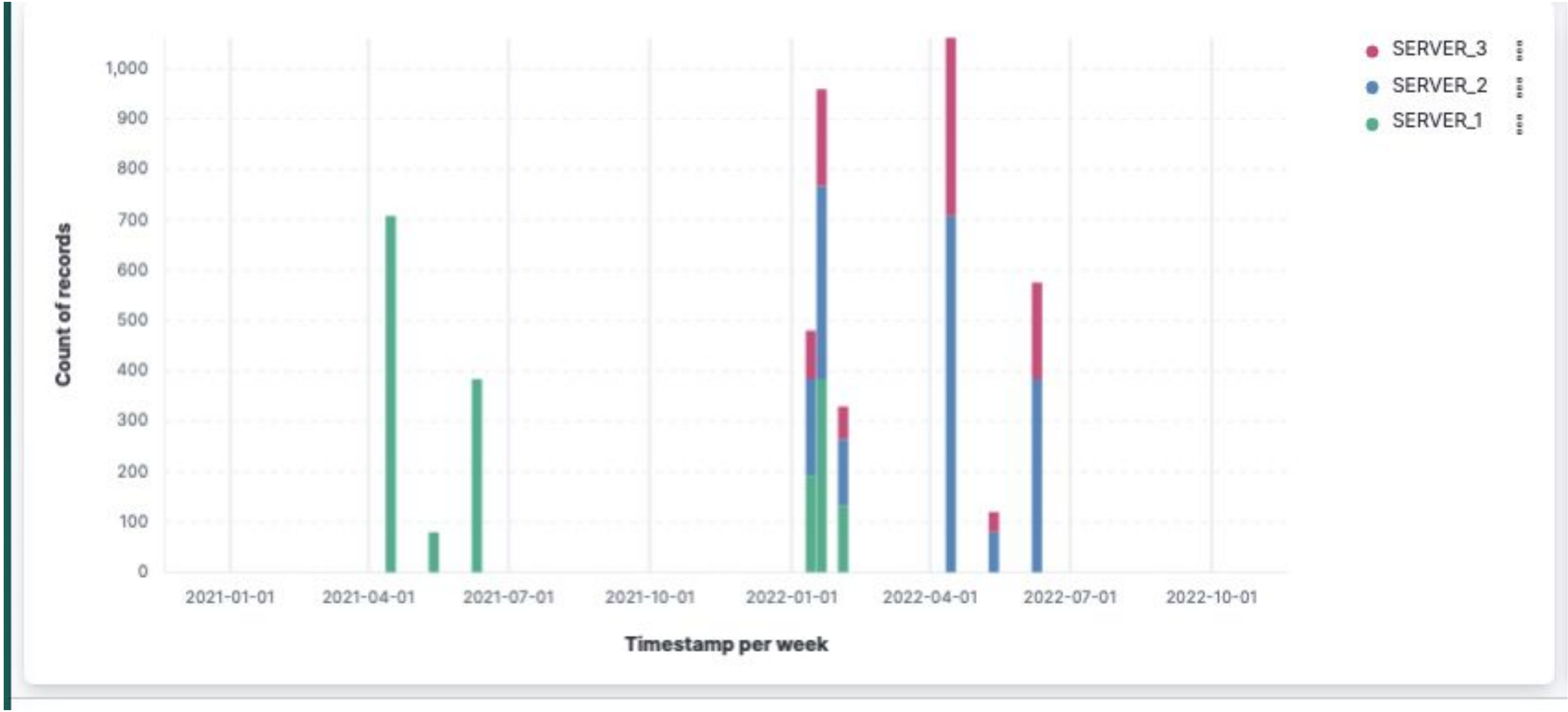
Kibana Screen Shots based on CT22T Process

Overview of the Amount of SQLs recorded per Coll & Server

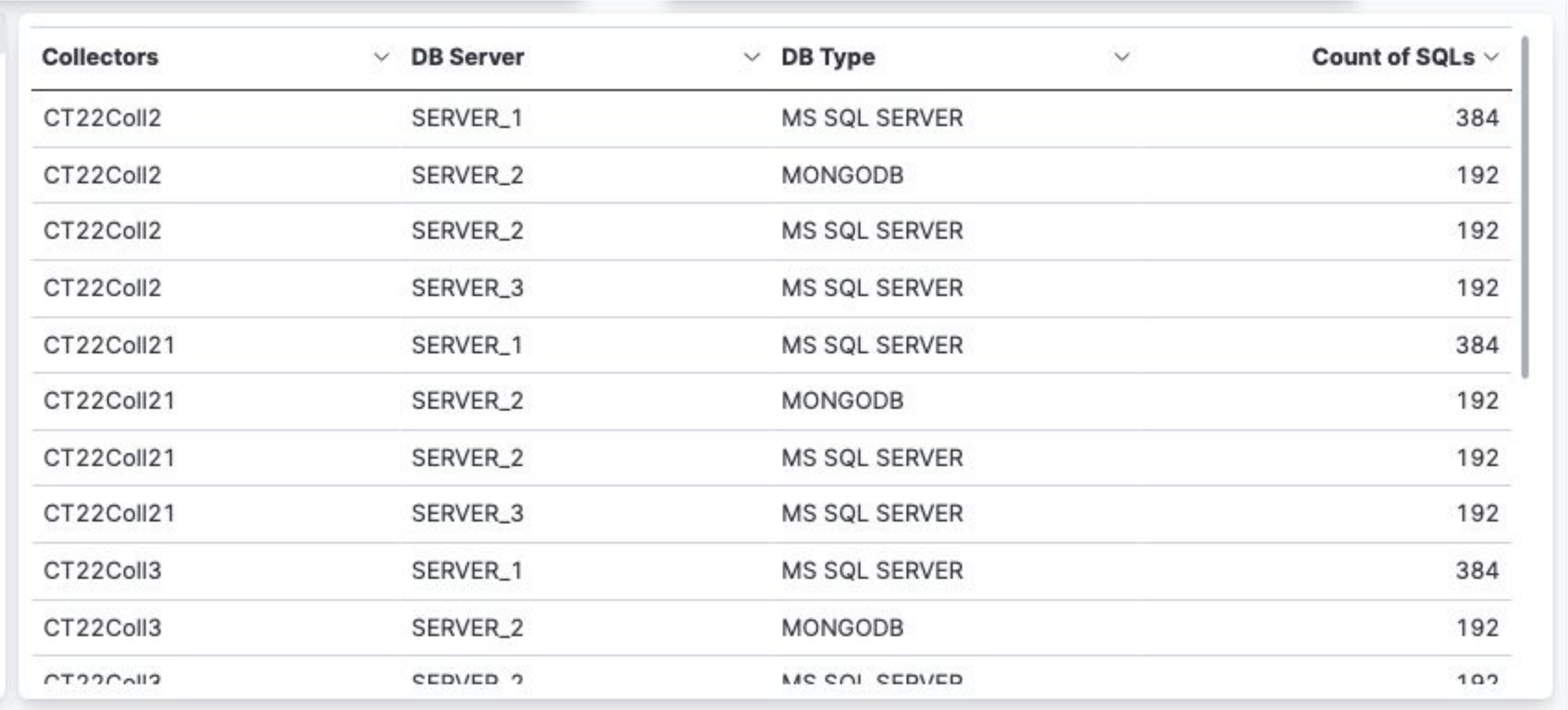




Overview of the Amount of SQLs over Time

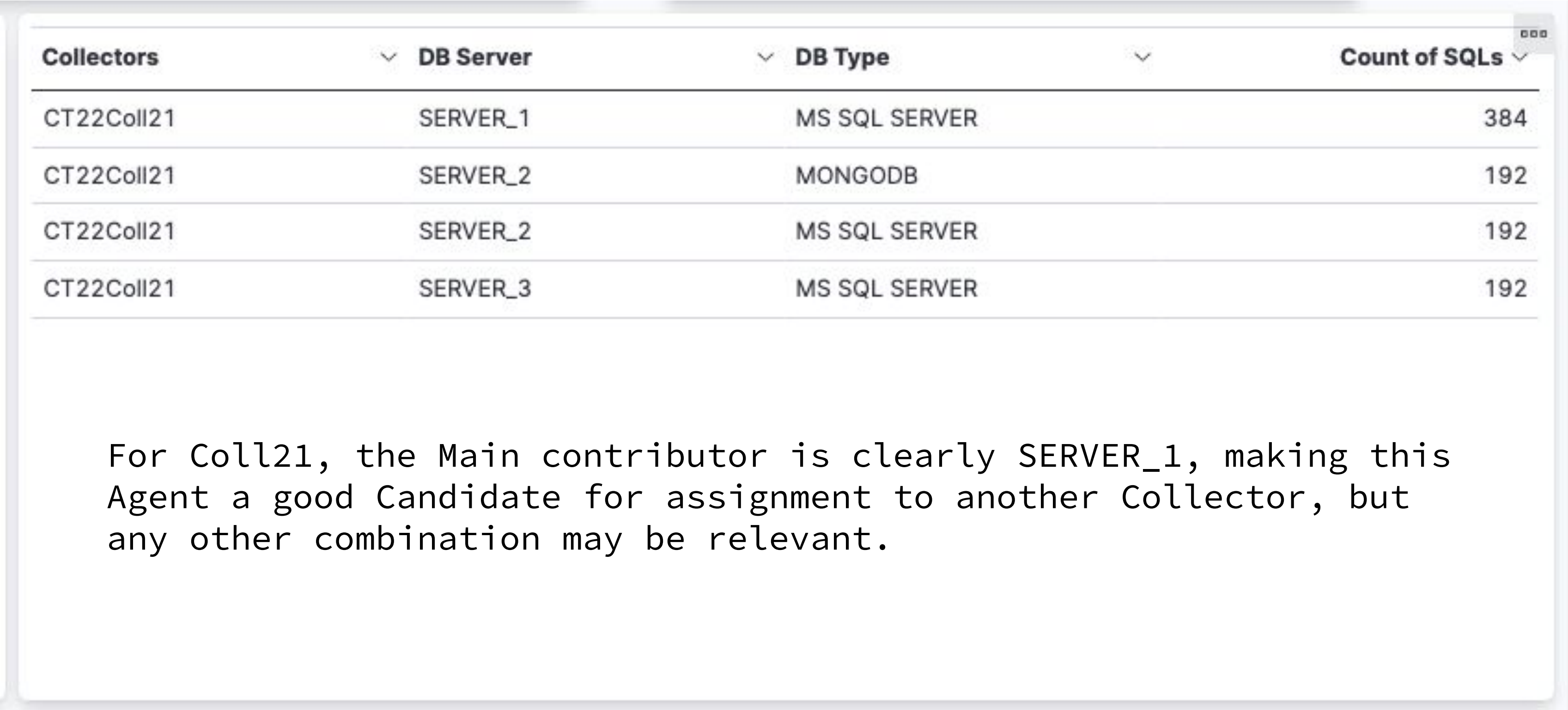


Overview of the Amount of SQLs in Tabular Format



Collectors	DB Server	DB Type	Count of SQLs
CT22Coll2	SERVER_1	MS SQL SERVER	384
CT22Coll2	SERVER_2	MONGODB	192
CT22Coll2	SERVER_2	MS SQL SERVER	192
CT22Coll2	SERVER_3	MS SQL SERVER	192
CT22Coll21	SERVER_1	MS SQL SERVER	384
CT22Coll21	SERVER_2	MONGODB	192
CT22Coll21	SERVER_2	MS SQL SERVER	192
CT22Coll21	SERVER_3	MS SQL SERVER	192
CT22Coll3	SERVER_1	MS SQL SERVER	384
CT22Coll3	SERVER_2	MONGODB	192
CT22Coll3	SERVER_2	MS SQL SERVER	192

Amount of SQLs For 1 Specific Collector in Tabular Format



Collectors	DB Server	DB Type	Count of SQLs
CT22Coll21	SERVER_1	MS SQL SERVER	384
CT22Coll21	SERVER_2	MONGODB	192
CT22Coll21	SERVER_2	MS SQL SERVER	192
CT22Coll21	SERVER_3	MS SQL SERVER	192

For Coll21, the Main contributor is clearly SERVER_1, making this Agent a good Candidate for assignment to another Collector, but any other combination may be relevant.

The End