IPFS Infrastructure Usage Logs (3.22 GB)
cid: bafybeiftyvcar3vh7zua3xakxkb2h5ppo4giu5f3rkpsqgcfh7n7axxnsa

1 Read log data

In [2]:

```python
import pandas as pd
import re
pd.set_option('max_columns', None)
pd.options.display.max_colwidth = 100
```

In [2]:

```python
# read log data
path = 'nginx-01-02-2021-bank2-sv15_encrypted.log'
file = open(path)
lines = file.readlines()
```

In [3]:

```python
# split by '"'
s = pd.Series(lines)
df = s.str.split('"', expand=True)

# remove rows with missing value
df['check1'] = df.apply(lambda x: len(x[1].split(' ')), axis=1)
df['check2'] = df.apply(lambda x: len(x[2].split(' ')), axis=1)
df = df[(df['check1'] == 3) & (df['check2'] == 9)].drop(['check1', 'check2'], axis=1)
```

In [4]:

```python
# create dataframe
df[['encryptedIP','timestamp']] = df[0].str.split(' ', expand=True).drop([1,2,4], axis=1)
df['timestamp'] = df['timestamp'].str.strip('[]')

df[['method','path','version']] = df[1].str.split(' ', expand=True)

df[['response', 'bytes_returned', 'request_length', 'request_time', 'upstream_response_time',
df['bytes_returned'] = df['bytes_returned'].astype(int)

df[['referrer','agent']] = df[[3,5]]

df[['server_name','http_host','http_schema']] = df[6].str.split(' ', expand=True).drop([0], ax
df['http_schema'] = df['http_schema'].str.strip('\n')

df = df.drop([0,1,2,3,4,5,6], axis=1)

df.shape
```

Out[4]:

```
(7169922, 17)
```

In [5]:

```python
# keep only GET requests
df = df[df['method']=='GET']
# filter valid agents
df = df[df['agent']!='yes']
df['check'] = df.apply(lambda x: re.search("^[a-zA-Z]", str(x['agent']))==None, axis=1)
df = df[df['check']==False].drop(['check'], axis=1)
# exclude path '/'
df = df[df['path']!='/']

df.shape
```

Out[5]:

```
(6866353, 17)
```

2 Extract CID

In [6]:

```
df['cid'] = ''
# extract CID from path
df['http_host'] = df['http_host'].str.strip(':443')
df.loc[df['path'].str.startswith('/ipfs'), 'cid'] = df['path'].str[6:]
df.loc[df['http_host'].str.endswith('.ipfs.dweb.link'), 'cid'] = df['http_host'].str[:-15]
df['cid'] = df.apply(lambda x: x['cid'].split('/')[0], axis=1)
df.head()
```

Out[6]:

	encryptedIP	timestamp	method	p
0	gAAAAABh-Vo0VtoLZ4C9ouT9ixNPqG74tCLkzKEaCTJvLR...	2022-01-02T00:00:38+00:00	GET	/ipfs/QmewCrTqsMECeYcX2etcuRAi2G37yNrL1QBsjxj
1	gAAAAABh-Vo0Ru3gCTTvtKzrLyYHguwPqaqVBUCBnnHBIT...	2022-01-02T00:00:38+00:00	GET	/ipfs/QmSoLuCB7xeFD5vf8pYnzoBhRFfnnM41nPy4zBn
2	gAAAAABh-Vo0qdpIKr_Kw7VH1HM8dFfqyAMCdHA8vpi0Q-...	2022-01-02T00:00:38+00:00	GET	/dan
3	gAAAAABh-Vo0YvaJZfSGDoelpTg6_0dJFIM6NcwD-4w9f6...	2022-01-02T00:00:38+00:00	GET	/dan
4	gAAAAABh-Vo0B03dW6C0_w9-_RnBaeCJia2kavg1IvelAD...	2022-01-02T00:00:38+00:00	GET	/ipfs/QmewCrTqsMECeYcX2etcuRAi2G37yNrL1QBsjxj

In [46]:

```
# remove nan
df = df[df['cid'].isna()!=True]
# filter valid cid
df['check'] = df.apply(lambda x: bool(re.match("^[A-Za-z0-9]*$", str(x['cid']))) and len(str(x
df = df[df['check']==True].drop(['check'], axis=1)
df.shape
```

Out[46]:

```
(6645871, 18)
```

In [49]:

```
df = df[['timestamp', 'bytes_returned', 'agent', 'cid']]
df.head()
```

Out[49]:

	timestamp	bytes_returned	agent	cid
0	2022-01-02T00:00:38+00:00	423	axios/0.17.1	QmewCrTqsMECeYcX2etcuRAi2G37yNrL1QBsjxjAgZSwfy
1	2022-01-02T00:00:38+00:00	185936	Mozilla/5.0 (Linux; U; Android 11; zh-cn; V2066A Build/RP1A.200720.012) AppleWebKit/537.36 (KHTM...	QmSoLuCB7xeFD5vf8pYnzoBhRFfnnM41nPy4zBnSqmjH7J
2	2022-01-02T00:00:38+00:00	464368	Mozilla/5.0 (Linux; Android 11; V2046A; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 C...	bafybeifyvews52mcsuqfbeoxxlzv5lewk37jc43b5tpbd3gzs3rvcktpaa
3	2022-01-02T00:00:38+00:00	1630912	Mozilla/5.0 (iPhone; CPU iPhone OS 14_7_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko...	bafybeifqhn5mwknicly5hb72bgs4m2674xu24kxjt7j25ebw2tej5wiiqy
4	2022-01-02T00:00:38+00:00	412	axios/0.17.1	QmewCrTqsMECeYcX2etcuRAi2G37yNrL1QBsjxjAgZSwfy

In [50]:

```
df.to_csv('data.csv') # 1.51 GB
```

3 Group by user

```
In [51]: df = pd.read_csv('data.csv', index_col=0)
         df.shape
```

```
Out[51]: (6645871, 4)
```

```
In [52]: df_groupby_user = df[['bytes_returned','agent']].groupby('agent').agg(['sum','count'])
         df_groupby_user.columns = df_groupby_user.columns.get_level_values(0) + '_' + df_groupby_user.
         df_groupby_user = df_groupby_user.reset_index()
         df_groupby_user = df_groupby_user.rename(columns={
                                  "bytes_returned_sum": "request_sum",
                                  "bytes_returned_count": "request_count",
                                 })
         df_groupby_user.shape
```

```
Out[52]: (21264, 3)
```

```
In [53]: df_groupby_user.head()
```

```
Out[53]:
```

	agent	request_sum	request_count
0	AVProMobileVideo/6.1.7.39280 (Linux;Android 10) ExoPlayerLib/2.15.0	6629429	1
1	AccompanyBot	244764	22
2	ActionExtension/3 CFNetwork/1220.1 Darwin/20.3.0	1586273	5
3	AirPlay/2.0 (App/30.172.0) MFi_AirPlay_Device (MFiModelGroup/257872-0020)	64108028	101
4	AirPlay/2.0 (App/30.172.0) MFi_AirPlay_Device (MFiModelGroup/ElVU8BViYT0YUCNRKu1tWQNNxfpQUqz5a9U...	525377961	413

```
In [54]: df_groupby_user.to_csv('data_groupby_user.csv') # 4 MB
```