Fortinet Remote Software Developer, DevOps and Machine Learning Project

Virus-Total-Implementation

Isingizwe Didier Frank

July 25th, 2022

Contents

Chapter 1

Code Layout

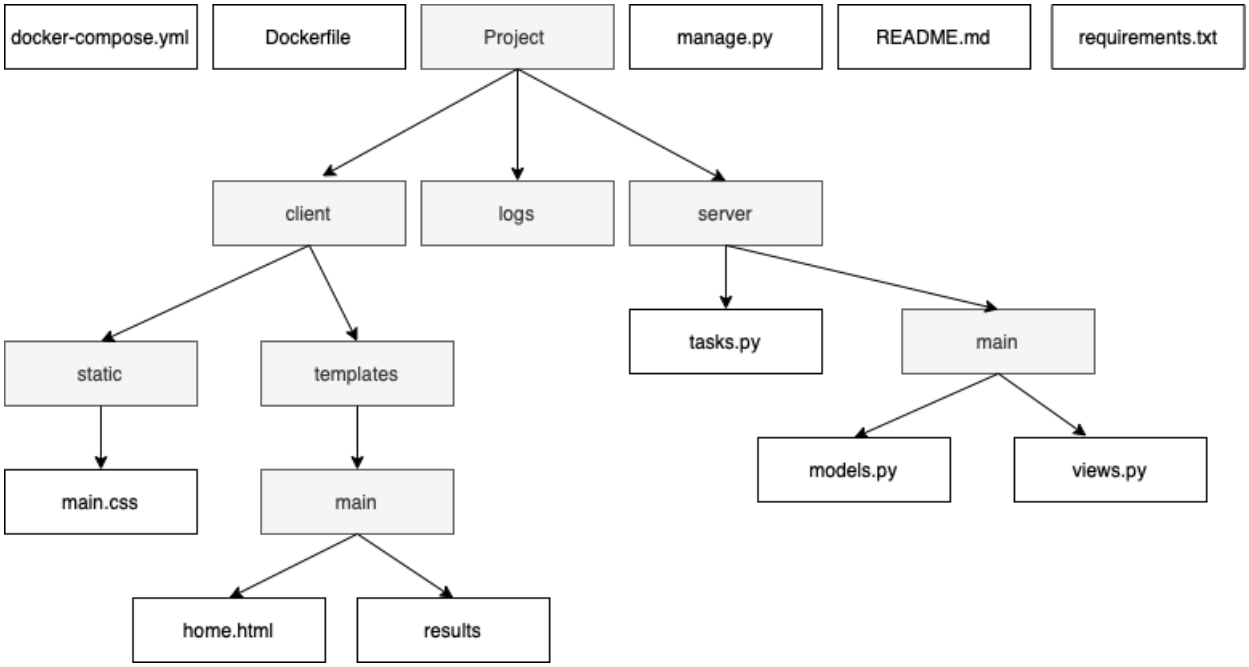

Figure 1.1: code layout

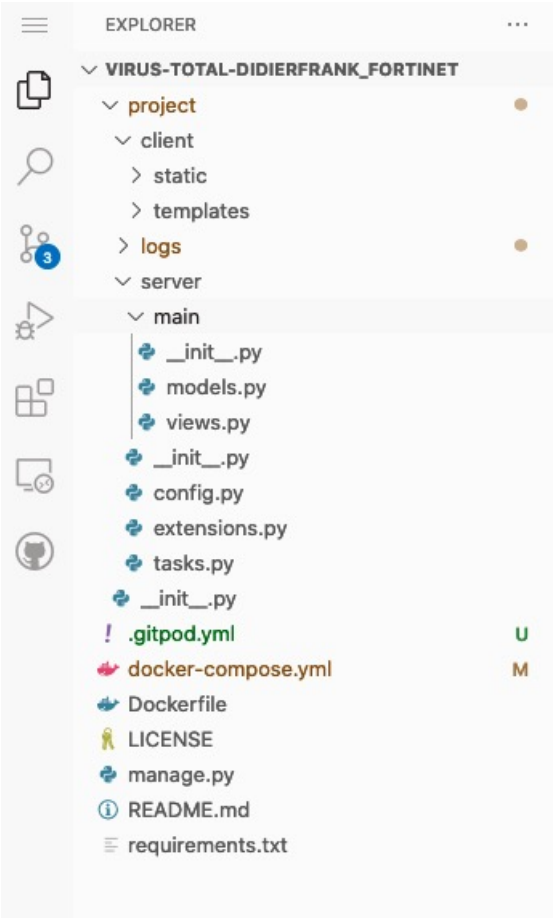

Figure 1.2: code layout2

Chapter 2

System Architecture

In this section, I designed a system architecture flow which i used to solve this project problem. Technologies used: Docker (to containerize the application), Flask (for web application) , Redis (for message broker) , celery (running asynchronous task), postgres (for database), Linux.

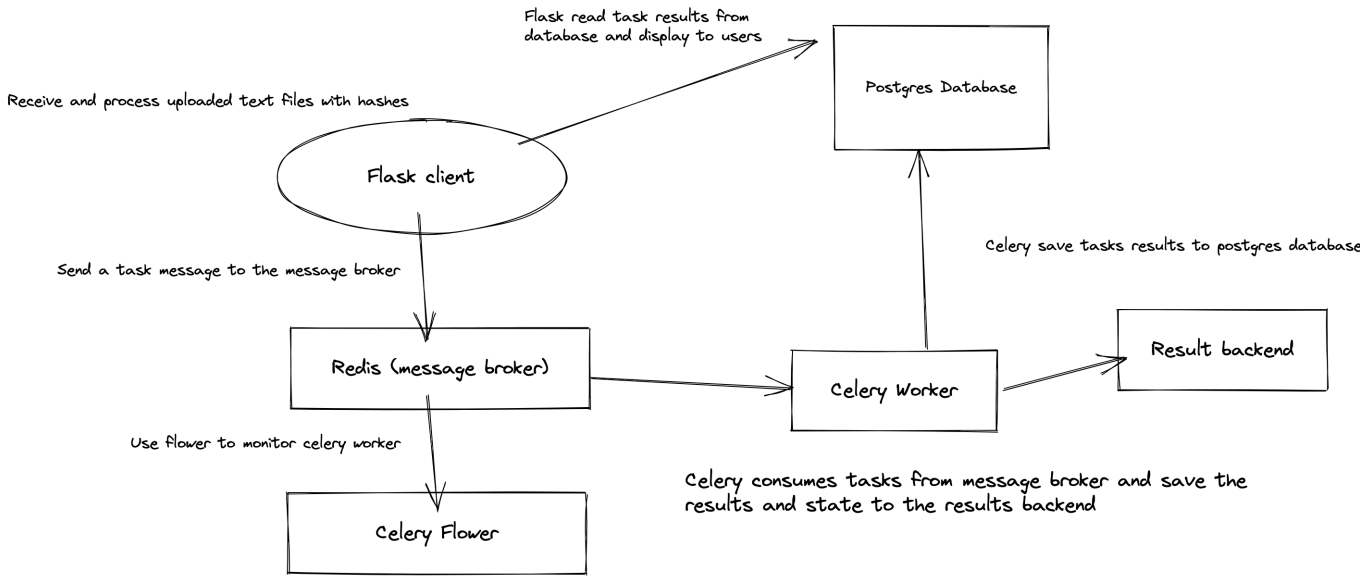

Figure 2.1: System flow

Chapter 3

Main Code Scope Explanation

In this section, I am explaining main part of the project which addresses the "Give some thoughts" section. Figure 3.1, the label A is responsible to check whether hash has been saved before and is less than one day, the label B is using existing data instead of using virus total api call and label C is using virus total api call if no data exist in database as well as handling different exceptions. Considering that virus total have api call limitations (QuotaExceededError), in label C, I am handling that problem by making some automatic retries after 24 hrs because every quote has 24 hrs.

```
@celery.task(name="fetch_file_info",bind=True)
def fetch_file_info(self,hash,job_id):
    task = Task.query.filter_by(hash=hash,job_id=job_id).first()

    # check if hash has been saved before and is less than one day
    existing_task = Task.query.filter(Task.results["data"].astext != "", Task.hash==hash, Task.created_at > datetime.now() - timedelta(days=1)).first()

    #Use existing data instead of api call
    if existing_task:
        task.results = existing_task.results
        db.session.commit()
        return task.results

    #use api call if no data exist in database
    try:
        results = fileinfo.info_file(hash)
        task.results = results
        db.session.commit()
        return results
    except virustotal3.errors.VirusTotalApiError as err:
        error_details = json.loads(err.message)
        if(error_details["error"]["code"] == "NotFoundError"):
            results = {"status": "error", "error_message":"File not found "}
        elif(error_details["error"]["code"] == "QuotaExceededError"):
            results = {"status": "error", "error_message":"API Quota exceeded will retry after 24 hours"}
            task.results = results
            db.session.commit()
            tomorrow = datetime.utcnow() + timedelta(days=1)
            raise self.retry(exc=err, countdown=tomorrow)
            return results
        else:
            results = {"status": "error", "error_message":"{}".format(error_details.message)}
        task.results = results
        db.session.commit()
        return results
    except Exception as err:
        results = {"status": "error", "error_message": "{}".format(err)}
        task.results = results
        db.session.commit()
        return results
```

A

B

C

Figure 3.1: Scope-code1

Figure 3.2, to perform asynchronous tasks with flask and celery, flask and celery runs in different containers.

```python
# This is required since flask and celery run in different containers
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = os.getenv("POSTGRES_URL")
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db.init_app(app)
```

Figure 3.2: Flask-celery-container

Figure 3.3, models.py file is handling postgres database tasks.

project > server > main > models.py

```python
from sqlalchemy.dialects.postgresql import JSON
from flask_hashids import HashidMixin, Hashids
from project.server.extensions import db
from datetime import datetime

# hashids = Hashids()

class Job(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    tasks = db.relationship('Task', backref='job')
    created_at = db.Column(db.DateTime, default=datetime.now)

class Task(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    hash = db.Column(db.String(80), nullable=False)
    job_id = db.Column(db.Integer, db.ForeignKey("job.id"), nullable=False)
    results = db.Column(JSON, nullable=True)
    created_at = db.Column(db.DateTime, default=datetime.now)

    def __init__(self, hash, job_id, results={}):
        self.hash = hash
        self.job_id = job_id
        self.results = results
```

Figure 3.3: Postgres: DB Column definition

Figure 3.4, considering that while counting number of engines I encountered some null values, so, this section is responsible for substracting number of engines with null values in order to get an accurate number of engines.

project > client > templates > main > results.html > ...

```
12  <table class="table">
13    <thead>
14      <tr>
15        <th>ID</th>
16        <th>Status</th>
17        <th>Fortinet detection name </th>
18        <th>Number of engines detected  </th>
19        <th>Scan Date</th>
20      </tr>
21    </thead>
22    <tbody id="tasks">
23      {% for task in results %}
24      <tr>
25        <td>{{ task.hash }}</td>
26        {% if task.results and task.results["data"] %}
27          <td>Completed</td>
28          <td>{{task.results["data"]["attributes"]["last_analysis_results"]["Fortinet"]["result"]}}</td>
29          <td>
30            {% set count = namespace(value=0) %}
31            {% for key, value in task.results["data"]["attributes"]["last_analysis_results"].items() %}
32            {% if value["result"] != "null" %}
33              {% set count.value = count.value +1 %}
34            {% endif %}
35            {% endfor %}
36
37            {{count.value}}
38          </td>
39        {% elif task.results and task.results["status"] == "error" %}
40          <td>Error</td>
41          <td>{{task.results["error_message"]}}</td>
42          <td>N/A</td>
43        {% else %}
44          <td>Pending</td>
45          <td>N/A</td>
46          <td>N/A</td>
47        {% endif %}
48        <td>{{task.created_at.strftime('%d-%m-%Y %H:%M:%S')}} </td>
49      </tr>
50      {% endfor %}
51    </tbody>
```

Figure 3.4: Handling Null Values

Chapter 4

Project Screenshot

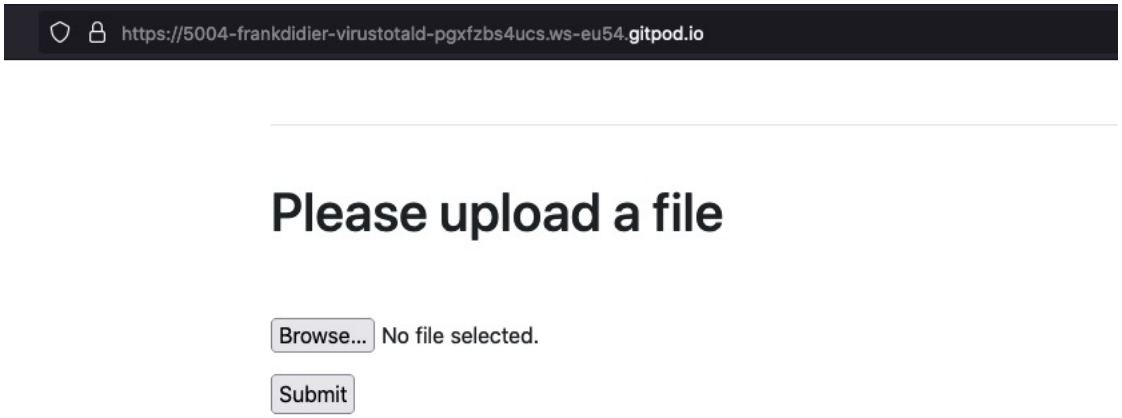

Figure 4.1: HomePage

https://5004-frankdidier-virustotald-pgxfzbs4ucs.ws-eu54.gitpod.io/results/2

Task Status

ID	Status	Fortinet detection name	Number of engines detected	Scan Date
0496f4962d3dce3caa849f605749f7f2	Completed	W32/Kryptik.EIWS!tr	73	25-07-2022 13:21:19
e85463d19104cacd79a25cacb0b57c1d	Completed	None	72	25-07-2022 13:21:19
ad04e313410dd865916b720e03e6b77e	Completed	W32/Injoker.ST!tr	73	25-07-2022 13:21:19
73d45bfefdef3a8b379887cf582a6105	Completed	W32/Banload.WQI!tr.dldr	75	25-07-2022 13:21:19
2090a5cf258c81d08c284f4ca0e367a7	Completed	W32/Malicious_Behavior.VEX	75	25-07-2022 13:21:19
f0810b0186d0bfa77b42f8e30e9a966a	Completed	W32/Agent.XMY!tr	75	25-07-2022 13:21:19
d8f1db6500c04dbfd9f231958d43125b	Completed	W32/Injector.CRIZ!tr	75	25-07-2022 13:21:19
e47789e7bf6cb9214479c1a44d48226f	Completed	W32/Generic.AC.45A1B9!tr	75	25-07-2022 13:21:19
bf360f859f5683d9c3c53cc112c54087	Completed	MSIL/Banload.FG!tr.dldr	75	25-07-2022 13:21:19
9d602778db57e86abb51a0fc9f908eba	Completed	W32/Autoit.BXM!tr	74	25-07-2022 13:21:19
30fddae663c97d23d79fd732daef2276	Completed	W32/Farfli.BAL!tr	75	25-07-2022 13:21:19
4f67cfcda368be58a8b9532b037f95cb	Completed	W32/Scar.MBZK!tr	75	25-07-2022 13:21:19
40d07e92e530a2ee8394ddcdd3995cb0	Completed	W32/Rovnix.AJ!tr	75	25-07-2022 13:21:19
e12358080d94f8e6481069349559f825	Completed	W32/Carbanak.A!tr	75	25-07-2022 13:21:19
221ad45d39c066fe698aff4d89fc2435	Completed	W32/Banload.WQS!tr.dldr	75	25-07-2022 13:21:19

Figure 4.2: Results

https://5556-frankdidier-virustotald-pgxfzbs4ucs.ws-eu54.gitpod.io/tasks

Flower Dashboard Tasks Broker

Show 10 entries Search:

Name	UUID	State	args	kwargs	Result	Received	Started	Runtime	Worker
fetch_file_info	37789803-a333-4f14-9fb5-0fb667852f97	SUCCESS	('0496f4962d3dce3caa849f605749f7f2', 1)	{}	{'data': {'attributes': {'type_description': 'Win32 EXE', 'tlsh': 'T10D24C035F7A0E9E7EDA084F01FADC2E1B639973013528893378 44BAE66342E5077A757', 'vhash': '025076655d151715156171z4002600c27ze7z52z620607bz', 'trid': [...], 'creation_date': 1450255186, 'names': [...], 'signature_info': {...}, 'last_modification_date': 1656425972, 'type_tag': 'peexe', 'times_submitted': 32, 'total_votes': {...}, 'size': 223744, 'popular_threat_classification': {...}, 'authentihash': 'bd1f3c5939ec0350407ed83b7d5ca5f962644977c910c819769697909c281462', 'last_submission_date': 1600886916, 'meaningful_name': 'TrueCrypt.exe', 'sandbox_verdicts': {...}, 'sha256': 'e8d5648612de5523fa245e31a6f5e79541ba524faf8ae1527269b2fd1f953e08', 'type_extension': 'exe', 'tags': [...], 'last_analysis_date': 1631486829, 'unique_sources': 24, 'first_submission_date': 1450257469, 'sha1': '4d35984d0a3a1c4281cfb4c72b9d6cfd9ff3adfd', 'ssdeep': '3072:r76fl6artVYwWHD/QxFHv51JjgS5/a3smrYA6zPiW8Hi3Fpdjz/Lq2ePQ:f6dG/TS5f06rOiV/ZcQ', 'md5': '0496f4962d3dce3caa849f6057...', ...}}}	2022-07-25 13:20:42.732	2022-07-25 13:20:42.736	0.737	celery@9c2
fetch_file_info	4431ce1f-1b70-45cb-abc2-834097531 7f9	SUCCESS	('82a02a0864447d51bb8c18ab4554a77e', 1)	{}	{'status': 'error', 'error_message': 'File not found '}	2022-07-25 13:20:42.734	2022-07-25 13:20:42.737	0.704	celery@9c2
fetch_file_info	df1b787b-f24c-400d-bdeb-2566f37d8641	SUCCESS	('ad04e313410dd865916b720e03e6b77e', 1)	{}	{'data': {'attributes': {'type_description': 'Win32 EXE', 'tlsh': 'T1B1F412126507E463F4170BB2D6CC42F40D3EBC83FF90A5EFAE457E2EB4B65A25442968', 'vhash': '075066551d1d15751015z91zb15kz12fz', 'trid': [...], 'creation_date': 1446913320, 'names': [...], 'signature_info': {...}, 'last_modification_date': 1647642762, 'type_tag': 'peexe', 'times_submitted': 10, 'total_votes': {...}, 'size': 765952, 'popular_threat_classification': {...}, 'authentihash': '724c6103d9f093d45c58cb69eee5696096deac7cee20b83b4cdba6befa8e6f4a', 'last_submission_date': 1600871250, 'meaningful_name': 'invoice.exe', 'sandbox_verdicts': {...}, 'sha256': '33f2fa32b29e8fecff9b92da47d8b154a44d3fe3d6c96ca4ac50c7cc6627b064', 'type_extension': 'exe', 'tags': [...], 'last_analysis_date': 1647635365, 'unique_sources': 8, 'first_submission_date': 1447663336, 'sha1': '3046a1e74ab923b67c92c5c3be84b70202cd9bc6', 'ssdeep': '12288:YCbtfVBEYP3/ceVY2v0Ig2MUnyqWwwytlR4GRvmf+VNQPjt97Xx3osCH:xpEYP3UevY28lNMUnyq/4GReWN2t2', 'packers': {...}, 'md5': 'ad04e313410dd8659...', ...}}}	2022-07-25 13:20:42.739	2022-07-25 13:20:42.741	0.761	celery@9c2
fetch_file_info	59cc7ac8-65b6-4b4f-91c9-28201fd9f55f	SUCCESS	('e85463d19104cacd79a25cacb0b57c1d', 1)	{}	{'data': {'attributes': {'type_description': 'Win32 EXE', 'tlsh': 'T1D2132A59AAAD44E1E2B2C17D89D30906E2B2F4605F2387CF6164D25F0F337E59E35312', 'vhash': '0440665d151515151cz221z11z103bz4fz', 'trid': [...], 'creation_date': 1414544583, 'names': [...], 'signature_info': {...}, 'last_modification_date': 1658301538, 'type_tag': 'peexe', 'times_submitted': 44, 'total_votes': {...}, 'size': 41472, 'type_extension': 'exe', 'authentihash': 'cbb687cb3a7b881a807b1830fec51f070246f1c9c3f59b1fa63fd1e6cc15289e', 'last_submission_date': 1645284002, 'known_distributors': {...}, 'meaningful_name': 'lpkinstall.exe', 'sha256': '71c660fd9bdc1b0677cc14930b5450b2489727526918551c39510025be2ca183', 'tags': [...], 'last_analysis_date': 1645284002, 'unique_sources': 9, 'first_submission_date': 1421477862, 'sha1': '4857300b1fd6da73c9b76d869290b4d44f885c78', 'ssdeep': '768:CUQ2PMOlKgXlr2/rhq20VVU9Swzg/hJBBWd5dy++2ZSBt:CnsRlKgXlJ2h9SwzgJhskZmSB', 'md5': 'e85463d19104cacd79a25cacb0b57c1d', 'pe_info': {...}, 'magic': 'PE32+ executable for MS...', ...}}}	2022-07-25 13:20:42.736	2022-07-25 13:20:42.742	1.050	celery@9c2
					{'data': {'attributes': {'type_description': 'Win32 EXE', 'tlsh': 'T1C8F412993396B4CFC827C8355AE11D54FB6625BF27A3861BA053132D9F2E993CB044F2', 'vhash': '275036750514808 5c30025', 'trid': [...], 'crowdsourced_yara_results': [...], 'creation_date': 1451938857, 'names': [...], 'dot_net_guids': {...}, 'last_modification_date': 1637066229, 'type_tag': 'peexe', 'times_submitted': 6, 'total_votes': {...}, 'size': 747008, 'popular_threat_classification': {...}, 'authentihash':				

Figure 4.3: Celery Flower Dashboard

Chapter 5

Comments and Difficulties

I faced difficulties while using virus total API platform due to API calls limitations (i reached the api limit after using the api for a couple of times), there is no way to access to a development version of the virustotal which will not block the api limit; most big companies like paypal and stripe have api for development that can be used without any limit just to test stuff instead of using your real api details.

I am still setting up the Live demo using digital ocean platform which will be live on Tuesday, 26th Vancouver time.