Application architecture

ITAPP

Software Security- POPBL

MONDRAGON UNIBERTSITATEA | MACCS

Index

Table of content

1. Input machine

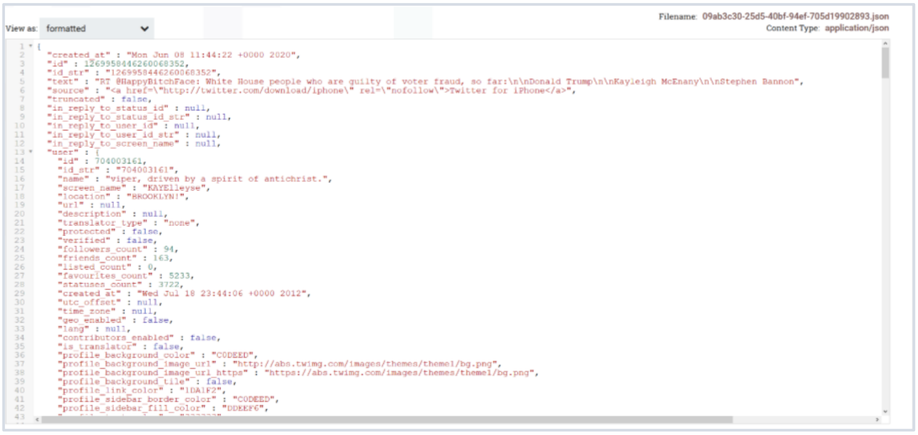
1.1 Infrastructure

The machine that receives the starting inputs is located in GCloud. It is running an Apache NiFi instance. The following security measures are set in place:

- Confidentiality
  - NiFi is secured with TLS, thus, the data that is sent to and sent by input.itapp.eus is completely confidential.
- Integrity
  - NiFi is secured with TLS, maintaining data integrity in all the connections from and to the input.itapp.eus
- Authentication
  - Not anyone can connect to NiFi to start and stop processors. The 8080 port on which NiFi runs only listens to localhost connections. You need to tunnel though SSH to the input.itapp.eus machine to be able to connect to NiFi. You need an SSH key to authenticate.

1.2 Entries

We have three data sources that send information in a variety of ways, through calls to some APIs and loading data from local, to the machine that collects the Entries, input.itapp.eus.


View as: formatted
Filename: 09ab3c30-25d5-40bf-94ef-705d19902893.json
Content Type: application/json

```
{
  "created_at" : "Mon Jun 08 11:44:22 +0000 2020",
  "id" : 1269958446260068352,
  "id_str" : "1269958446260068352",
  "text" : "RT @HappyBitchFace: White House people who are guilty of voter fraud, so far:\n\nDonald Trump\n\nKayleigh McEnany\n\nStephen Bannon",
  "source" : "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>",
  "truncated" : false,
  "in_reply_to_status_id" : null,
  "in_reply_to_status_id_str" : null,
  "in_reply_to_user_id" : null,
  "in_reply_to_user_id_str" : null,
  "in_reply_to_screen_name" : null,
  "user" : {
    "id" : 704003161,
    "id_str" : "704003161",
    "name" : "viper, driven by a spirit of antichrist.",
    "screen_name" : "KAYElleyse",
    "location" : "BROOKLYN!",
    "url" : null,
    "description" : null,
    "translator_type" : "none",
    "protected" : false,
    "verified" : false,
    "followers_count" : 94,
    "friends_count" : 163,
    "listed_count" : 0,
    "favourites_count" : 5233,
    "statuses_count" : 3722,
    "created_at" : "Wed Jul 18 23:44:06 +0000 2012",
    "utc_offset" : null,
    "time_zone" : null,
    "geo_enabled" : false,
    "lang" : null,
    "contributors_enabled" : false,
    "is_translator" : false,
    "profile_background_color" : "C0DEED",
    "profile_background_image_url" : "http://abs.twimg.com/images/themes/theme1/bg.png",
    "profile_background_image_url_https" : "https://abs.twimg.com/images/themes/theme1/bg.png",
    "profile_background_tile" : false,
    "profile_link_color" : "1DA1F2",
    "profile_sidebar_border_color" : "C0DEED",
    "profile_sidebar_fill_color" : "DDEEF6",
```

Illustration 1. Entries

These entries are JSON formatted.

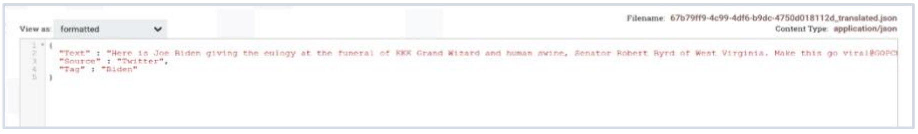
- Validation & Sanitization

- We ensure that the file that NiFi receives is a JSON.
- JSONs that do not have the fields we need are discarded.
- The JSON is parsed through a whitelist that filters characters we do not allow.

1.3 Outputs

The machine input.itapp.eus sends the JSONs through HTTP via a Remote Process Group in NiFi once they have been treated and enriched to the gateway machine, gateway.itapp.eus.

- Validation & Sanitization
    - Before sending the JSON we check its structure and characters once again.
        - We ensure the enrichment we have done is there.
        - We ensure the flowfiles those JSON create are not empty.



Filename: 67b79ff9-4c99-4df6-b9dc-4750d018112d_translated.json
Content Type: application/json

```
{
  "Text" : "Here is Joe Biden giving the eulogy at the funeral of KKK Grand Wizard and human swine, Senator Robert Byrd of West Virginia. Make this go viral@GOPCH
  "Source" : "Twitter",
  "Tag" : "Biden"
}
```

Illustration 2. Outputs

2. Gateway Machine

2.1 Infrastructure

The machine that receives data from input.itapp.eus, gateway.itapp.eus, is located in GCloud. It is running an Apache NiFi instance. The following security measures are set in place:

- Confidentiality
    - NiFi is secured with TLS, thus, the data that is sent to and sent by gateway.itapp.eus is completely confidential.
- Integrity
    - Nifi is secured with TLS, maintaining data integrity in all the connections from and to the gateway.itapp.eus
- Authentication
    - Not anyone can connect to NiFi to start and stop processors. The 8080 port on which NiFi runs only listens to localhost connections. You need to tunnel though SSH to the gateway.itapp.eus machine to be able to connect to NiFi. You need an SSH key to authenticate.
    - Not anyone can send data to Gateway's NiFi. The iptables configuration in gateway.itapp.eus blocks all incoming traffic except for the input.itapp.eus machine.

2.2 Entries

We have only one input channel communication where data comes from the input.itapp.eus machine.

The entries are JSON formatted.

- Validation
    - We ensure that the file that NiFi receives is a JSON.

2.3 Outputs

The machine gatway.itapp.eus sends the JSONs through HTTP via a Remote Process Group in NiFi once they have been treated to the cloud NiFi cluster (node1.itapp.eus, node2.itapp.eus, node3.itapp.eus).

Once we have checked the input for gateway is a JSON file we trust its origin and just send it to cloud.itapp.eus.

3. Cloud Machine (Nifi-Cluster)

3.1 Infrastructure

The cluster is composed by three machines that receive data from the gateway machine. The cluster is located in GCloud. It is running an Apache NiFi Cluster. The following security measures are set in place:

- Confidentiality
    - NiFi Cluster is secured with TLS, thus, the data that is sent to and sent by all the nodes is completely confidential.
- Integrity
    - Nifi is secured with TLS, maintaining data integrity in all the connections from and to all the nodes.
- Authentication
    - Not anyone can connect to NiFi to start and stop processors. The 8080 port on which NiFi runs only listens to localhost connections. You need to tunnel though SSH to all the nodes to be able to connect to NiFi. You need an SSH key to authenticate.
    - Not anyone can send data to Cloud's NiFi. The iptables configuration in cloud.itapp.eus blocks all incoming traffic except for the gateway.itapp.eus machine.

3.2 Entries

We have only one input channel communication where data comes from the gateway.itapp.eus machine.

The entries are JSON formatted.

- Validation
    - We ensure that the file that NiFi receives is a JSON file.
    - Once we convert it to JSON we rerun all the checks from input.itapp.eus.

3.3 Outputs

The cluster sends the JSONs once they have been treated and enriched to the Hadoop Cluster, which is located in the same machines.

- Validation & Sanitization

        - Before sending the JSON we check its structure and characters once again.
            - We ensure the enrichment we have done is there.

The JSONs are grouped into groups of 10 to lower the number of files on HDFS.

4. Hadoop cluster

3.4 Infrastructure

The cluster is composed by three machines that receive data from the NiFi cluster. The cluster is located in GCloud. It is running a Hadoop Cluster. The following security measures are set in place:

- Confidentiality
  - Hadoop Cluster is secured with TLS, thus, the data that is sent to and sent by all the nodes is completely confidential.
- Integrity
  - Hadoop is secured with TLS, maintaining data integrity in all the connections from and to all the nodes.
- Authentication
  - Not anyone can connect to Hadoop to start and stop processors. You need an SSH key to authenticate yourself in the machine and stop Hadoop.
  - Not anyone can write data to Hadoop. Only processes which are owned by users that are in the supergroup group of the Hadoop Cluster can write data in it. Thus, only our NiFi should be able to write there. LEAST PRIVILEGE

3.5 Entries

We have only one input channel communication where data comes from the NiFi Cluster.

- The entries are JSON formatted.
- There are no security checks for the files in Hadoop.

3.6 Outputs

The cluster stores all the data in specific folders.

5. Spark Cluster

5.1 Infrastructure

The cluster is composed by three machines located in AWS. It is running a Spark Cluster.

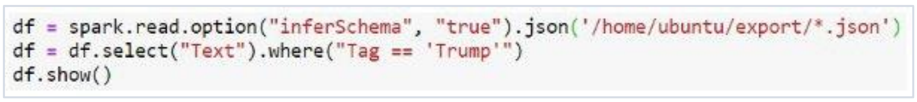
The following security measures are set in place:

- Confidentiality
  - The Spark Cluster is secured with TLS, thus, the data that is sent to and sent by all the nodes is completely confidential.
- Integrity
  - Spark is secured with TLS, maintaining data integrity in all the connections from and to all the nodes.
- Authentication
  - Not anyone can connect to Spark to execute notebooks. We use Spark's Log4j implementation to enable username/password based logins to the web UI (still under construction).

5.2 Entries

The notebooks we execute for the training fetch data from the Hadoop cluster. This data is trusted and thus it is not verified.

- Authentication
  - To be able to read from the Hadoop cluster the user that executes the Spark process needs to be added to the Hadoop's cluster supergroup.



```
df = spark.read.option("inferSchema", "true").json('/home/ubuntu/export/*.json')
df = df.select("Text").where("Tag == 'Trump'")
df.show()
```

Illustration 3. Entries Authentication

5.3 Outputs

The cluster will create the model that will be used in the project which we do not check on save.

6. WEB APP

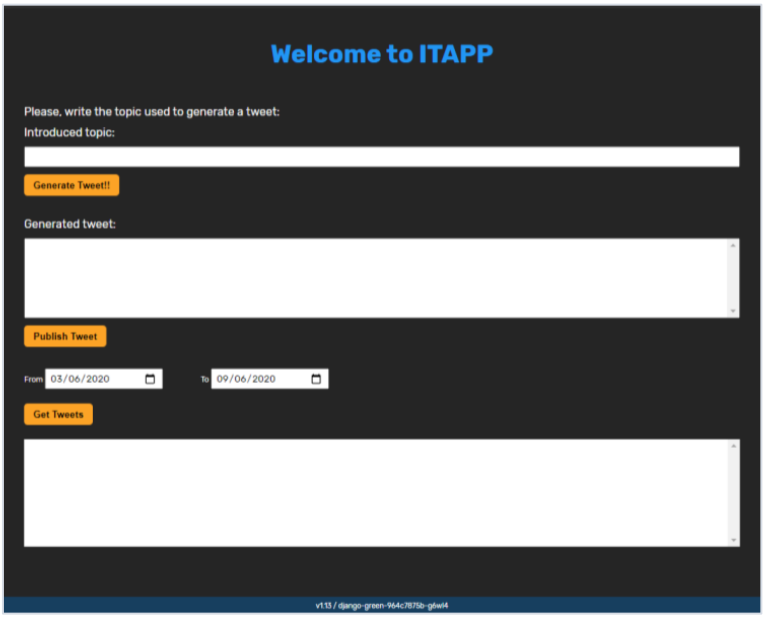
6.1 Infrastructure

The Django based web application runs over a machine located on Google Cloud. Anybody on the internet can access to it.

- Confidentiality
  - The web application is secured with TLS, thus, the data exchanges that happen with users are completely confidential.
- Integrity
  - The same way, TLS maintains the integrity of the messages that are exchanged with the web application.
- Authentication
  - Maybe we will add a login screen via Key Cloak, for now the web app allows anyone to interact with it.

6.2 Entries

The web application has three main data entries/inputs. Those are the imports and loading of the model the backend does, the user input fields on the application and the tweet loading it does from elastic search.


Welcome to ITAPP

Please, write the topic used to generate a tweet:

Introduced topic:

Generated tweet:

From 03/06/2020 To 09/06/2020

v1.13 / django-green-964c7875b-g6wl4

Illustration 4. Itapp application

- Input testing

        - All the inputs present in the web application will be fuzz tested to avoid undesirable behaviors.
- Validation & Sanitization
    - All the input fields have limited characters you can input to them.
    - Once an input is sent to the backend it is filtered comparing it against regular expressions.
    - The data received from elastic search is checked:
        - It should be a JSON.
        - The dates of the first and last tweet should coincide with what the user inputted.

6.3 Outputs

The web application has two main data outputs. It posts the curated tweets to a Twitter account we have set up and saves those exact tweets on elastic search.

- The user's edition of the generated tweet is limited to known characters.
- Once an input is sent to the backend it is filtered comparing it against regular expressions.
- A JSON is formed with that tweet that respects the JSON structure we are using in elastic search.
    - We ensure the generated tweet has that structure.
- The new tweet is sent to twitter, elastic search and HDFS.

7. Secure coding practices

7.1 Code security

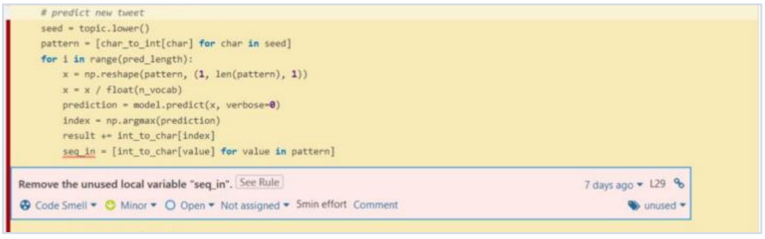
The development of the code of the web application Itapp will follow the following Secure Software Design Principles in order to create the most secure software possible. Those design principles will be checked against SonarQube, the static code analysis tool we have used.

7.2 Code styling, zero smells, zero bugs

Itapp has been developed according to the needs we established and the warnings SonarQube was giving us. That way the code has ended up with no code smells nor bugs.

These SonarQube scans have been automatized to pass with the pipeline's execution, although they cannot stop the pipeline itself. The scans are manually checked after each execution.

To exemplify some of the changes made to the code here are some unused variable warnings that have been fixed:

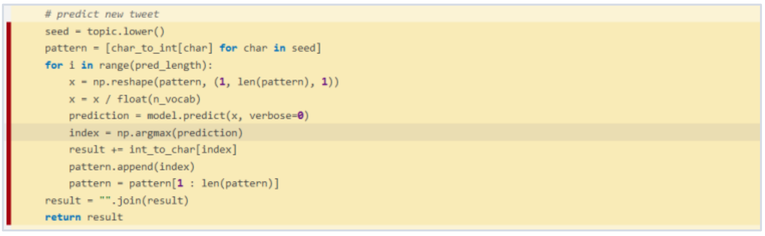


```
# predict new tweet
seed = topic.lower()
pattern = [char_to_int[char] for char in seed]
for i in range(pred_length):
    x = np.reshape(pattern, (1, len(pattern), 1))
    x = x / float(n_vocab)
    prediction = model.predict(x, verbose=0)
    index = np.argmax(prediction)
    result += int_to_char[index]
    seq_in = [int_to_char[value] for value in pattern]
```

Remove the unused local variable "seq_in". See Rule   7 days ago ▾ L29

Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 5min effort Comment   unused ▾

Illustration 5. Before fixed



```
# predict new tweet
seed = topic.lower()
pattern = [char_to_int[char] for char in seed]
for i in range(pred_length):
    x = np.reshape(pattern, (1, len(pattern), 1))
    x = x / float(n_vocab)
    prediction = model.predict(x, verbose=0)
    index = np.argmax(prediction)
    result += int_to_char[index]
    pattern.append(index)
    pattern = pattern[1 : len(pattern)]
result = "".join(result)
return result
```

Illustration 6. After fixed

7.3 Economy of mechanism:

The code will be segmented into different parts that work in unison, each of the parts having a concrete goal and being the smallest and simplest possible. This translates to using the least lines of code possible and following the guidelines on code complexity set by SonarQube.

Code complexity is measured in the amount of nested checks a piece of code has.

For example, our code had the following code complexity issue solved:

8. Resumen y reflexión

Finalmente, todo el proceso se resume en el siguiente diagrama, en el que podemos observar los pasos a seguir en el momento de querer hacer un cambio:

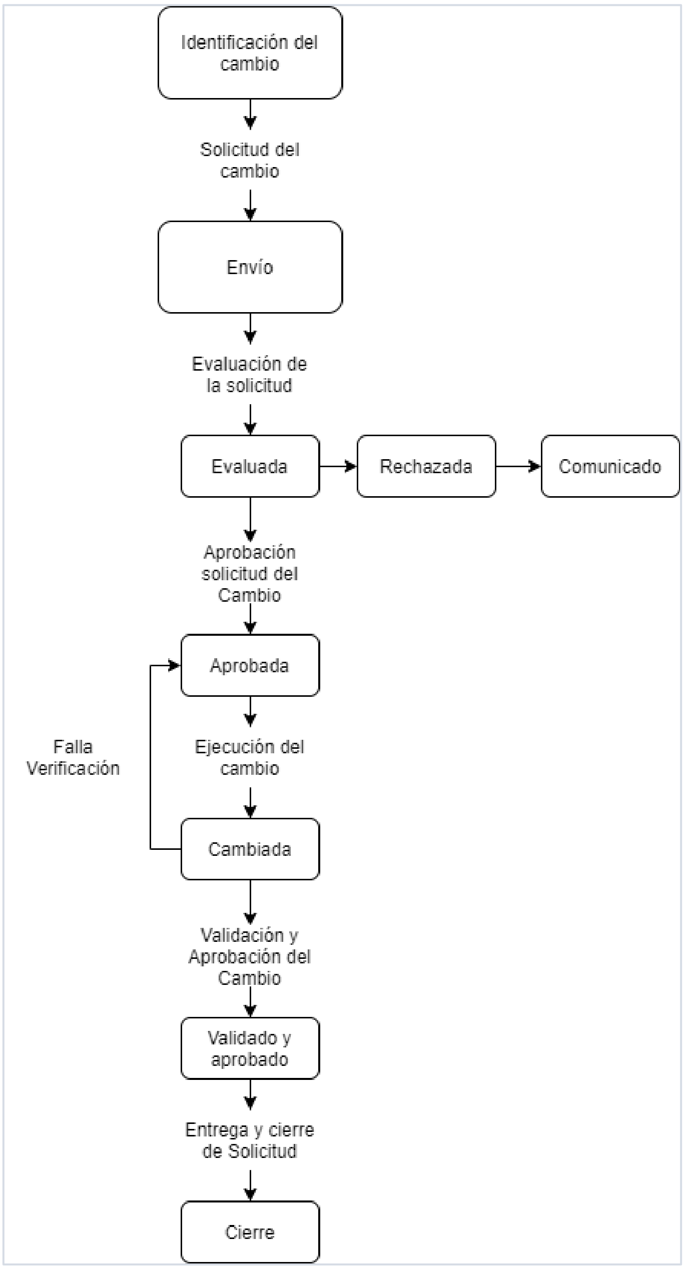

Identificación del cambio
Solicitud del cambio
Envío
Evaluación de la solicitud
Evaluada
Rechazada
Comunicado
Aprobación solicitud del Cambio
Aprobada
Falla Verificación
Ejecución del cambio
Cambiada
Validación y Aprobación del Cambio
Validado y aprobado
Entrega y cierre de Solicitud
Cierre

Illustration 7. Diagram

Finamente, a modo de reflexión, es interesante indicar que debido a que se está tratando con clientes/integrantes del equipo de desarrollo, es posible que encuentre cierta resistencia cuando se intente implementar un proceso de gestión de cambios.

Aunque con el tiempo, se comprueba que este control de cambios es mutuamente beneficioso ya que aclara qué pueden esperar los clientes y cuáles serán las implicaciones del cambio propuesto. Dado que esto incluye el coste y la calidad final, el proceso de control de cambios se realiza no solo en interés de su negocio sino también del cliente.

Los clientes pueden ser reacios a preparar la solicitud inicial para la documentación de cambios, pero es posible prepararlos en base a una conversación y luego enviarlos al cliente para determinar si ha entendido correctamente lo que se requiere. Una vez que el cliente está dispuesto a aceptar formalmente que la solicitud de cambio se ha capturado con precisión, el proceso puede avanzar sin problema.