



# Cloud & Green Computing (**LAB**)

**HANDS ON**  **HEROKU**

# Cosa faremo?

Deploy di una web app su **Heroku** utilizzando alcuni *add-ons*.

Necessario:

1. Scaricare il codice dell'app (da completare)
  - `app.py` e le directories `static`, `templates`
2. Account GitHub, Heroku e AWS (con utente IAM)

# Web App - Specs

- Selezionare un'immagine
- Generare un *QR code* per visualizzarla
- Applicare il *QR code* come watermark sull'immagine stessa
- Visualizzare l'immagine finale (originale + watermark)

# Step 1

- Arricchire la web app con le funzionalità richieste, senza doverle implementare, ma utilizzando gli add-ons (*chiamate REST ad un servizio già esistente*), forniti da Heroku:
  - **Qrackajack:** creare il *QR code* per l'immagine caricata sul bucket
    - <https://devcenter.heroku.com/articles/qrackajack>
  - **Watermarker:** applicare il *QR code* all'immagine
    - <https://devcenter.heroku.com/articles/watermarker>

## Step 2 – Completare il file app.py

```
@app.route('/watermark', methods=['POST'])
def apply_watermark():
    bucket_name = "" # INSERT YOUR BUCKET NAME

    filename = request.form['filename']
    path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    r1 = s3_client.upload_file(path, bucket_name, filename, ExtraArgs={'ACL': 'public-read'})

    # GENERATE REQUEST FOR QRACKAJACK
    qr_req_url = ""

    qr_name = f"qr_{filename}"
    qr_path = request_and_save(qr_req_url, qr_name)

    r2 = s3_client.upload_file(qr_path, bucket_name, qr_name, ExtraArgs={'ACL': 'public-read'})

    # GENERATE REQUEST FOR WATERMARKER
    watermark_req_url = ""

    watermark_name = f"watermark_{filename}"
    request_and_save(watermark_req_url, watermark_name)

    print("watermark done")

    # clean bucket
    s3_client.delete_object(Bucket=bucket_name, Key=qr_name)

    return render_template("upload.html", filename=watermark_name)
```

Completare la funzione  
`apply_watermark()`

N.B.: sono presenti funzioni di  
utilità, all'interno del codice, che  
potrete utilizzare.

# Step 3 – Host su Heroku

1. Assicurarsi di avere **tutti i file necessari** per il deploy di un'app Python su Heroku.
  - : è necessario definire le dipendenze dell'app? Istruzioni aggiuntive per l'esecuzione?
2. scegliere ***Deploy con GitHub***
3. affinché l'app fornisca funzioni, inserire le credenziali dell'utente IAM come **variabili d'ambiente su Heroku**, con i seguenti nomi:
  - `AWS_ACCESS_KEY_ID`
  - `AWS_SECRET_ACCESS_KEY`

# File da consegnare

- `report_GX.pdf`, **uno per gruppo**
- `jorunal_name_surname.pdf`, **uno per studente**