Loadsmart SRE Code Challenge

Your task is the following:

- Create an API in Python that implements the spec in the Swagger 2.0 file
- Provision a machine and deploy your code into it using AWS
- The machine should be reachable via port 80 and through the ELB called `default-elb`. Ideally, we should be able to manage it via the same API you built.

Consider your API production ready:

- Automate as much as possible
- Make your application containerized
- Provide integration tests

You are expected to provide*:

- Code for infrastructure creation
- Description of alerting rules
- A stress test measurement scenario
- Documentation
- Separate text file (5-6 lines max) explaining your solution

Be prepared to be questioned during the interview process on each choice you made.

* please send it on a zip file.

swagger-file.yaml

```yaml
swagger: '2.0'
info:
 description: SRE Test - Loadsmart
 version: 1.0.0
 title: Site Reliability Engineer Test
 contact:
   email: jobs@loadsmart.com
security:
 - basicAuth: []
paths:
 '/healthcheck':
   get:
     description: API health check
     responses:
       200:
         description: the service is up
 '/elb/{elbName}':
   get:
     operationId: listMachinesElb
     description: List machines attached to a particular load balancer
     produces:
       - application/json
     responses:
       '200':
         description: machines listed
         schema:
           type: array
           items:
             $ref: '#/definitions/MachineInfo'
       '404':
         description: the elb does not exist
   post:
     operationId: attachInstance
     description: Attach an instance on the load balancer
     consumes:
       - application/json
     produces:
       - application/json
     parameters:
       - in: body
         name: machineId
         description: instance identifier
         schema:
           $ref: '#/definitions/MachineId'
     responses:
       '201':
         description: instance added
         schema:
           $ref: '#/definitions/MachineInfo'
       '400':
         description: wrong data format
       '409':
```

```
            description: instance already on load balancer
    delete:
      description: Detach an instance from the load balancer
      parameters:
        - in: body
          name: machineId
          description: instance identifier
          schema:
            $ref: '#/definitions/MachineId'
      responses:
        '201':
          description: instance removed
          schema:
            $ref: '#/definitions/MachineInfo'
        '400':
          description: wrong data format
        '409':
          description: instance is not on load balancer
    parameters:
      - name: elbName
        in: path
        required: true
        description: pass the load balancer name
        type: string
securityDefinitions:
 basicAuth:
   type: basic
definitions:
 MachineInfo:
   type: object
   required:
     - instanceId
     - instanceType
     - launchDate
   properties:
     instanceId:
       type: string
       example: i-5203422c
     instanceType:
       type: string
       example: t1.micro
     launchDate:
       type: string
       example: '2016-08-29T09:12:33.001Z'
 MachineId:
   type: object
   required:
     - instanceId
   properties:
     instanceId:
       type: string
       example: i-5203422c
```