Description:

Our goal is to find the network coverage of our customers.
To reach this goal we will use the data regarding their environment and their endpoints and map the possible coverage gaps and attacks.

Your goal is to :

1. Build a service that has 2 REST endpoints:
    - /connections - which will get a host_id as a query parameter and return a JSON list of the hosts ids that can potentially reach it
    - /statistics - which will return service statistics in a JSON format:
      number of hosts in the environment,
      number of requests to all endpoints and average request processing time in milliseconds.
      Statistics should be from process startup.
2. Write at least 2-3 automatic tests that will verify the functionality (at least 1 positive test and 1 negative test) and at least 1 performance test for this service.
3. Add logging and readme file on how to run the service

Example of using the attack endpoint:
$curl ‘http://localhost/api/v1/connections?host_id=host-123abc’
[“host-poi123”]

$curl ‘http://localhost/api/vi/statistics’
{“host_count”:2, “request_count”: 1125896, “average_request_time”:0.002589715}

More Details:

The input for your service is a JSON document describing the customer's network environment.
Environment is described using 2 types of objects: Hosts and firewall rules.
The structure of the cloud environment JSON is:

```
{
        “hosts”: [ hosts ],
        “fw_rules”: [ rules ]
}
```

A host has the following structure:

```
{
        “host_id”: “host-xxxx”,
        “name”: “ server A”,
        “label”: [“label1”,...]
}
```

host_id - an identifier that uniquely identifies a host
name - a user friendly display name
label - a list of zero or more label strings

To allow hosts to communicate with each other we will have to define it in the FW rule.
Firewall rules have the following structure:

```
{
        “fw_rule_id”: “fw-xxx”,
        “src_label”: “label1”,
        “dst_label”: “label2”
}
```

fw_rule_id - an identifier that uniquely identifies a firewall rule
src_label - a string that represents the source label of a traffic
dst_label - a string that represents the destination label of a traffic

In the example above, all traffic from hosts that have “label1” is allowed to access hosts that have “label2”.
A few notes:

1. Hosts with the same label can not access each other unless there is a firewall rule explicitly allowing that
2. The firewall rules are not transitive, meaning that if A can access B and B can access C, it doesn't mean that A is able to access C.