Hack9 2019

High performance in the cloud

Hack9 of 2019 has a focus on the cloud. The goal of each team is to build an application in the cloud of your choice. The emphasis will be on the performance of the solution, offset by the costs it incurs. Hack Niners are encouraged to utilize all managed services the cloud of their choice offers, to cut down cost and effort of implementation.

The task

The task is to build a REST service for telecom accounting. The service will support basic functions, like registering a call that was made and reporting.
The service is initialized by the CSV file, which you can find here.
Configuration file consists of records, each describing one price entry. A price entry has the dialing prefix, a point in time it is valid from, initial seconds, rounding increment and the actual price of the call, per minute. For a given dialed number and moment of the call the price entry is selected according to these rules:

1. Find rules with the longest prefix that matches the dialed number
2. Among those, select the rule with most recent “valid from” time, still before time-of-call

The cost of the call is calculated as the price per minute divided by 60 and effective seconds. Effective number of seconds is calculated as: initial seconds plus actual seconds, rounded to the upper bound of increment. Mathematical formula is:

$$effective\ duration = \ CEILING\left(\frac{initial + reported\ duration}{increment}\right) * increment$$

$$call\ cost = effective\ duration * \frac{price\ per\ minute}{60}$$

All financial numerical values are rounded to 2 fractional digits.

For example:

Prefix	Valid from	Initial	Increment	Price
+381	2019-01-01T00:00:00	10	10	4.2
+38121	2019-01-01T00:00:00	20	5	4.0
+38121	2019-06-01:00:00:00	20	5	3.0
+38121123	2019-08-01:00:00:00	25	15	1.0

These would be the prices and costs of calls, for 5-minute call:

Number called	Time	Price	Cost	Explanation
+38121654321	2019-04-01T12:30:00	4.0	320s * 4 / 60 = 21.33	Price entry #2
+38121123456	2019-08-20T14:23:18	1.0	330s * 1 / 60 = 5.5	Price entry #4

+38121123456	2019-07-01T12:02:28	3.0	320s * 3 / 60 = 16	Price entry #3

REST service needs to support endpoints described by the OpenAPI YAML, provided here. We advise you to use Swagger Editor or similar tool to visualize YAML file.

Cloud and budget

Each team has chosen a cloud and Hack9 crew has provided accounts for those clouds.
Hack Niners are advised to consider the price of managed services they plan to use. The right choice of services can actually lower the cost of the solution, while wrong choice can get the team over budget. Budget per team is $75 (the amount is the same for Azure and AWS).

Testing

Each team must register their solution with the testing panel provided on URL https://hack9judge.levi9.com/. After any team member logs in (using domain credentials), they can provide URL pointing to the solution they are building. The dashboard will run a judging application: "Judge Thread", which will run tests on the provided solution. We encourage teams to register early on, to monitor the progress of their work.
Judge Thread will run a battery of functional tests to validate the implementation correctness. The endpoints which pass the functional tests will be candidates for performance tests. Results will be stored and used for scoring.
Please note that Judge Thread runs in these regions of the cloud providers:

Cloud provider	*Region*
Amazon Web Services	Eu-west-1
Microsoft Azure	West Europe