

Given we have n number of docker instances or n number of ec2 instances that consists of a framework called optimization solver that we have purchased a single license costing over 20k USD that limits each execution of solve to 5 parallel executions. How can we design or extend the python/java/nodejs/c# package so that whenever we try to invoke solve we honor the contractual agreement ensuring that only 5 docker instances/processes/threads can use this function.

### Constraints

- Ensure you cater that each process should not hold the license forever this should be time configurable. Number of licenses can be increased.
- Cater for runtime exceptions in mid processing.
- We can cluster batches of licenses example stg/dev/prod
- Implementation should be able to run as a standalone package deployable into a web app or plain console java app
- Assignment should be sent over to the interview manager CC'd to the HR preferable a link to GIT hub repo
- Assignment is time bound to be done next day before 12pm

# Internal library below

Class OptimizationSolver(object)

def solve(data):

Return someresult

### Usage

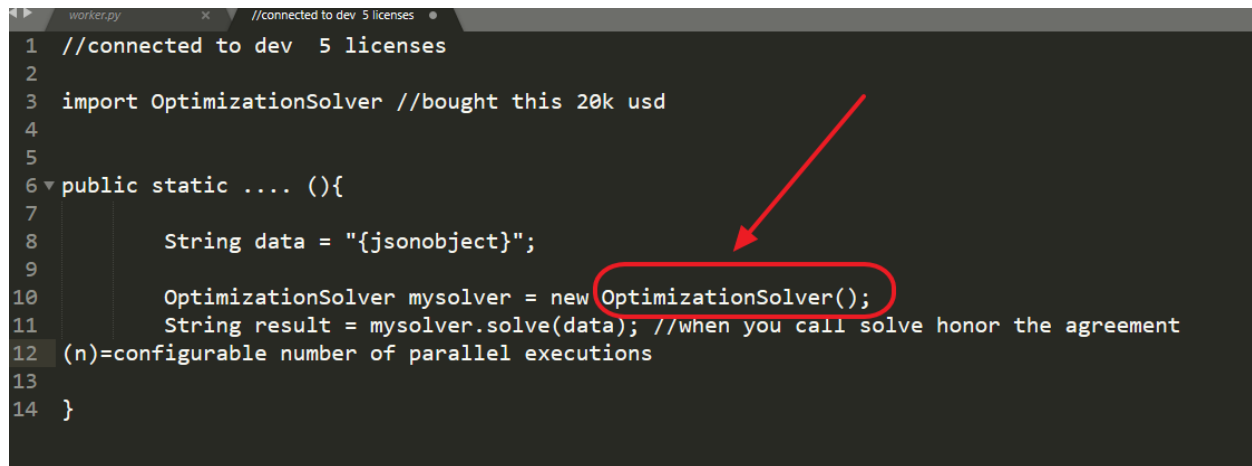
def myfunction():

opt\_solver = OptimizationSolver()

result = opt\_solver.solve(data)

print("result={}".format(result))

If this is answered in java main function should be as below if other language should reflect the same concept



```
1 //connected to dev 5 licenses
2
3 import OptimizationSolver //bought this 20k usd
4
5
6 public static .... (){
7
8     String data = "{jsonobject}";
9
10    OptimizationSolver mysolver = new OptimizationSolver();
11    String result = mysolver.solve(data); //when you call solve honor the agreement
12    (n)=configurable number of parallel executions
13
14 }
```

**Interview Manager(s):**

timothy.mugayi@mbwww.com

PohSoon.Yap@mbwww.com