

**Institute of Systems Science
National University of Singapore**

**GRADUATE CERTIFICATE
INTELLIGENT REASONING SYSTEMS**

Workshop Project Guide (2/5)

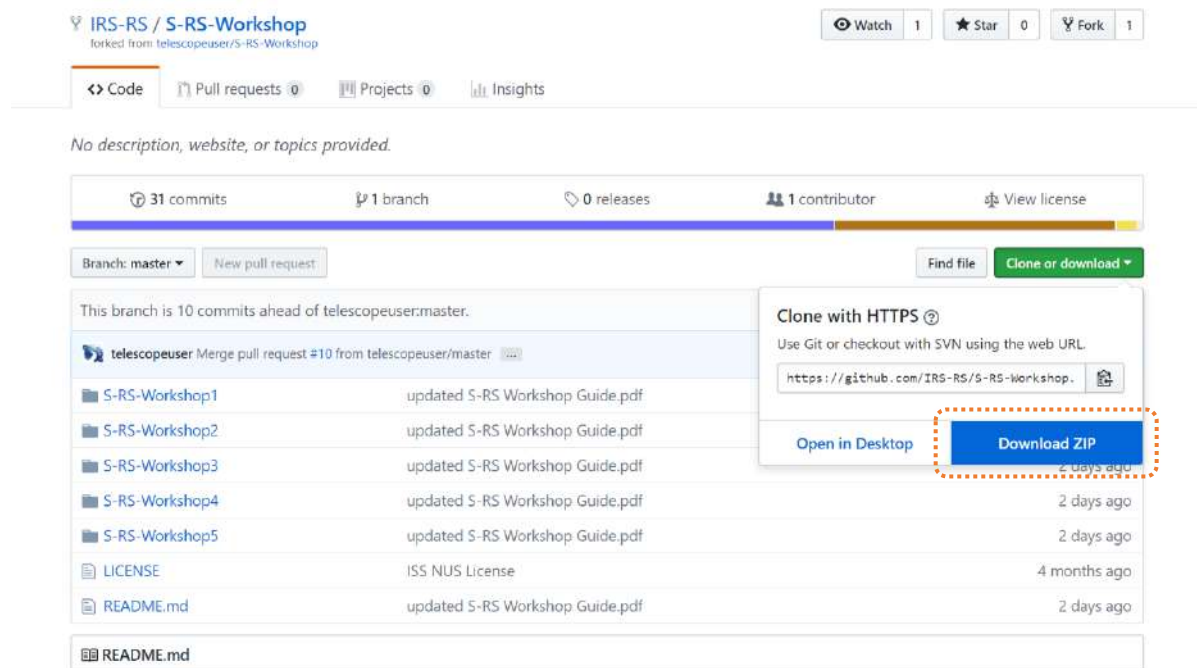
Subject: Reasoning Systems

© 2019 National University of Singapore All Rights Reserved.

The contents contained in this document may not be reproduced in any form or by any means, without the written permission of Institute of Systems Science, National University of Singapore other than for the purpose for which it has been supplied

*Institute of Systems Science, 25 Heng Mui Keng Terrace,
Singapore 119615*





Workshops & References <https://github.com/IRS-RS/S-RS-Workshop>



Virtual Machine Workstation <http://bit.ly/iss-vm>

Table of Contents

1. Workshop 2 – Search Reasoning.....	4
1.1. Cloud Balance Solver Deep Dive.....	5
1.1.1. Cloud Balance Solver [Java IDE].....	9
1.1.2. Cloud Balance Solver [KIE Workbench] v1.0.0	21
1.2. Cloud Balance Solver Enhancement.....	46
1.2.1. Cloud Balance Solver Enhancement [Data Objects].....	47
1.2.2. Cloud Balance Solver Enhancement [DRL Rule].....	49
1.2.3. Cloud Balance Solver Enhancement [Deploy] v2.0.0.....	53
1.2.4. Cloud Balance Solver Enhancement [API].....	54

1. Workshop 2 – Search Reasoning

WORKSHOP SEARCH REASONING

- **Cloud Balance Solver Deep Dive**
 - Cloud Balance Solver [Java IDE]
 - Cloud Balance Solver [KIE Workbench]
- **Cloud Balance Solver Enhancement**
 - GPU requirements; Data centre physical locations; Network latency, etc.

1.1. Cloud Balance Solver Deep Dive

{ Objective }

Analyse and execute cloud computer balancing system/solver using both Eclipse IDE and KIE Workbench (KIE Server / RESTful API).

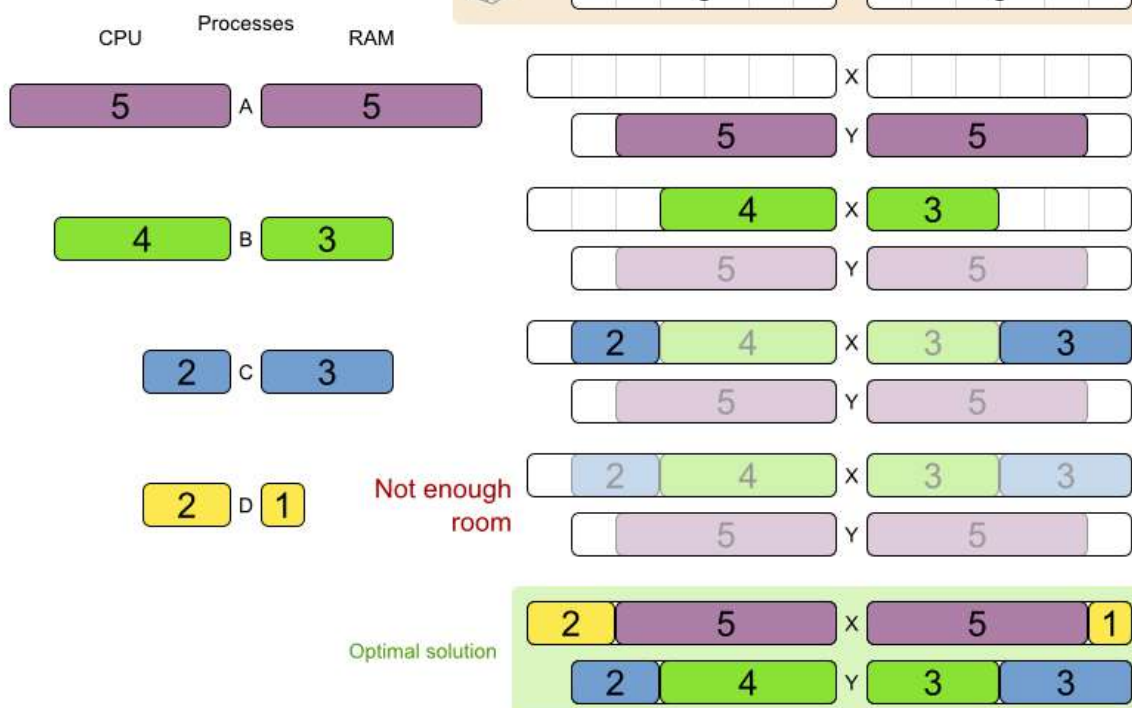
Decompose the end to end system solution, at system modelling level, to address the cloud balancing business resource optimization problem.

KIE OptaPlanner Deep Dive – Cloud Computer Balancing

- **Business Scenario / Problem Description**
- **A cloud service provider owns a number of cloud computers and needs to run a number of customers' processes on those computers. Assign each process to a computer.**

Cloud balance

Assign each process to a computer.



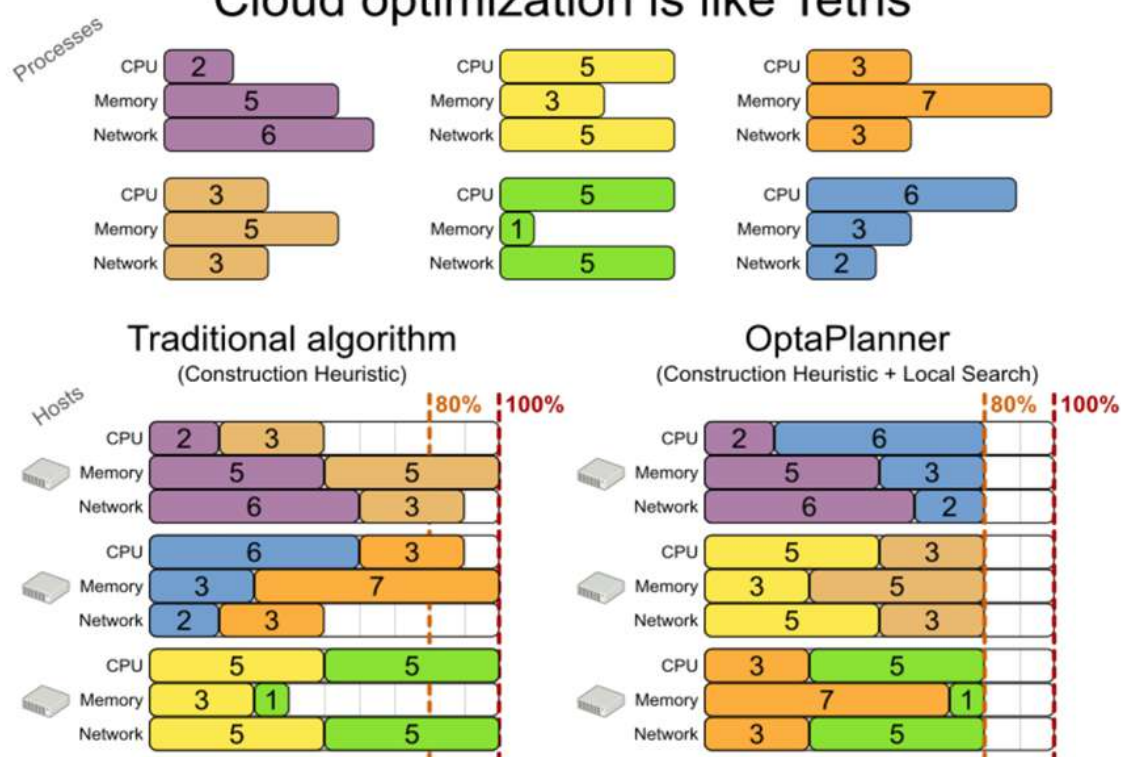
The following hard constraints must be fulfilled:

- Every computer must be able to handle the minimum hardware requirements of the sum of its processes:
 - **CPU capacity:** The CPU power of a computer must be at least the sum of the CPU power required by the processes assigned to that computer.
 - **Memory capacity:** The RAM memory of a computer must be at least the sum of the RAM memory required by the processes assigned to that computer.
 - **Network capacity:** The network bandwidth of a computer must be at least the sum of the network bandwidth required by the processes assigned to that computer.

The following soft constraints should be optimized:

- Each computer that has one or more processes assigned, incurs a maintenance cost (which is fixed per computer).
 - **Cost:** Minimize the total maintenance cost.

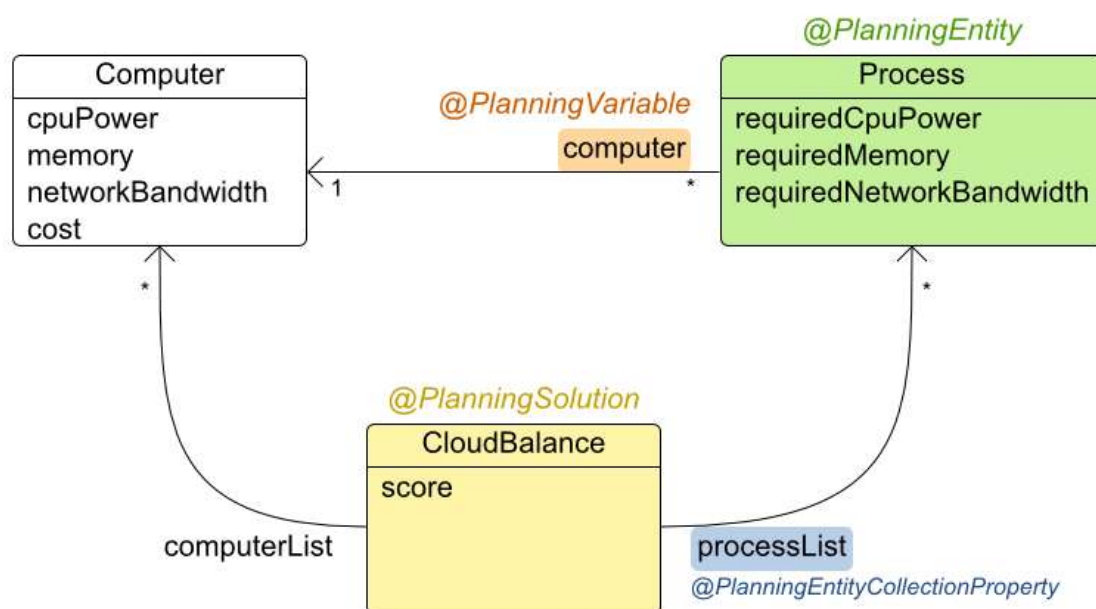
Cloud optimization is like Tetris

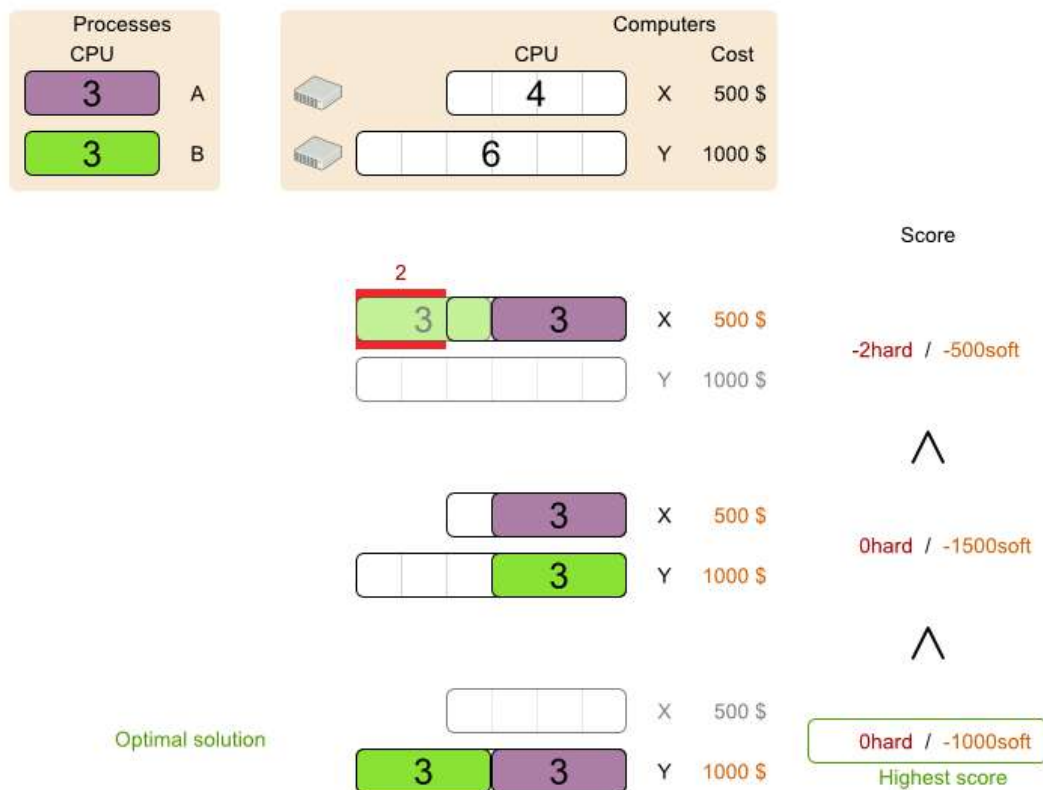


• Domain Modelling / Constraint Satisfaction

- To create a domain model, define all the objects that represent the input data for the problem. In this simple example, the objects are **processes** and **computers**.
- A separate object (Solution Class) in the domain model must represent a full data set of problem, which contains the input data as well as a solution. In this example, this object holds a list of **computers** and a list of **processes**. Each process is assigned to a computer; the **distribution of processes between computers** is the **solution**.

Cloud balance class diagram





Drools score calculation

- Constraints in Drools Rule Language (DRL)
 - Declarative (like SQL, regular expression)
- Integration opportunities
 - Drools Workbench
 - Decision tables

Drools score calculation: facts

- Facts in DRL loaded from
 - `@ProblemFact(Collection)Property`
 - `@PlanningEntity(Collection)Property`

DRL hard constraint: CPU power

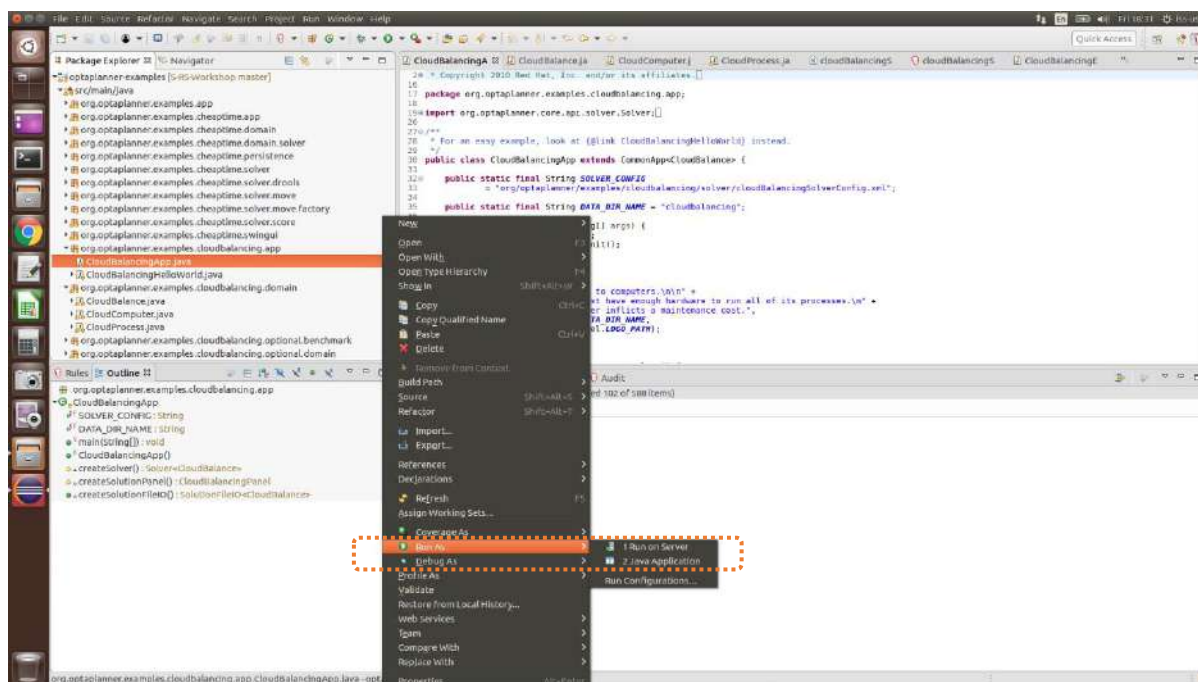
```
rule "requiredCpuPowerTotal"
when
  // there is a computer
  $s : Computer($cpu : cpuPower)
  // with too little cpu for its processes
  accumulate(
    Process(computer == $s, $requiredCpu : requiredCpuPower);
    $total : sum($requiredCpu);
    $total > $cpu
  )
then
  // lower hard score by the excessive CPU usage
  scoreHolder.addHardConstraintMatch(kcontext,
    $cpu - $total);
end
```

DRL soft constraint: computer cost

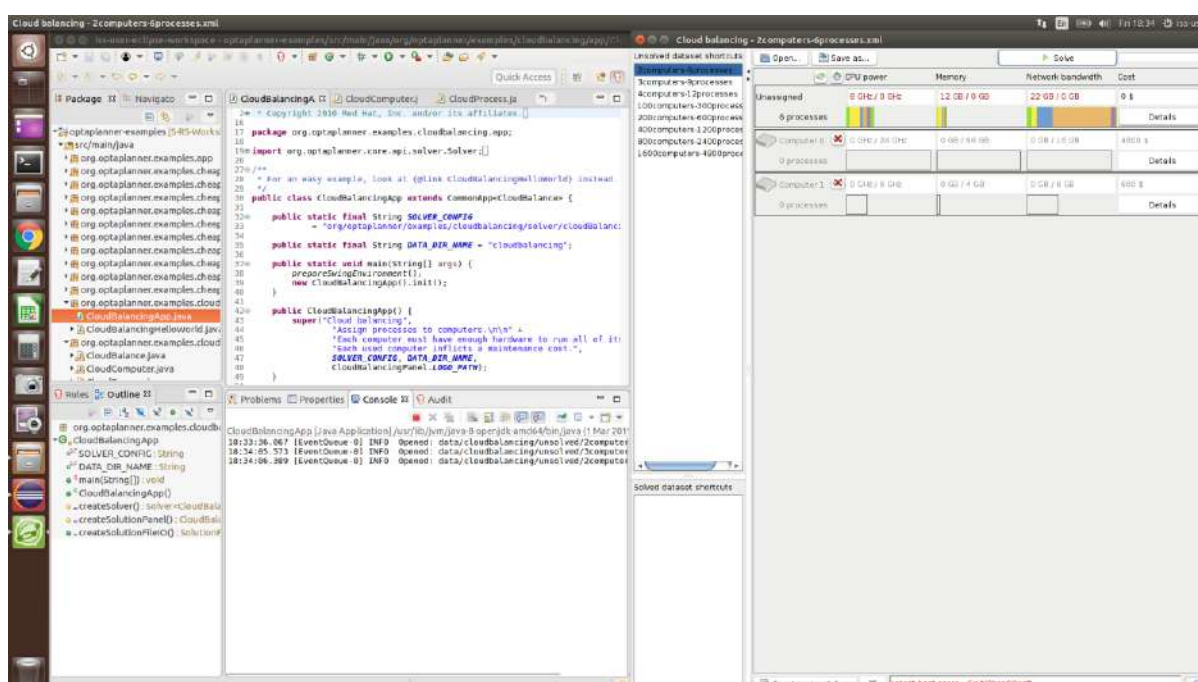
```
rule "computerCost"
when
  // there is a computer
  $s : Computer($c : cost)
  // there is a processes on that computer
  exists Process(computer == $s)
then
  // lower soft score by the maintenance cost
  scoreHolder.addSoftConstraintMatch(kcontext, - $c);
end
```

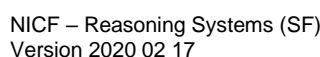

1.1.1. Cloud Balance Solver [Java IDE]

- 1) Expand project to **CloudBalancingApp.java**; right click mouse to call menu; Select **Run As : 2 Java Application**;



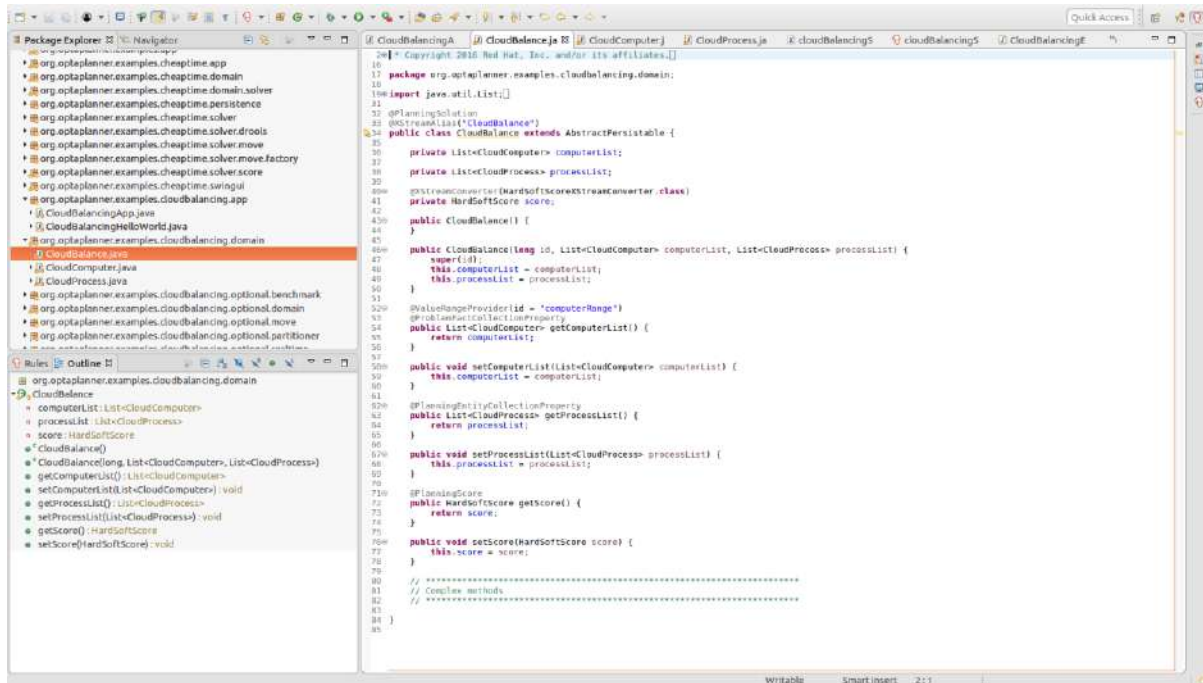
- 2) Select **2 computers – 6 processes**; Click **Solve**;



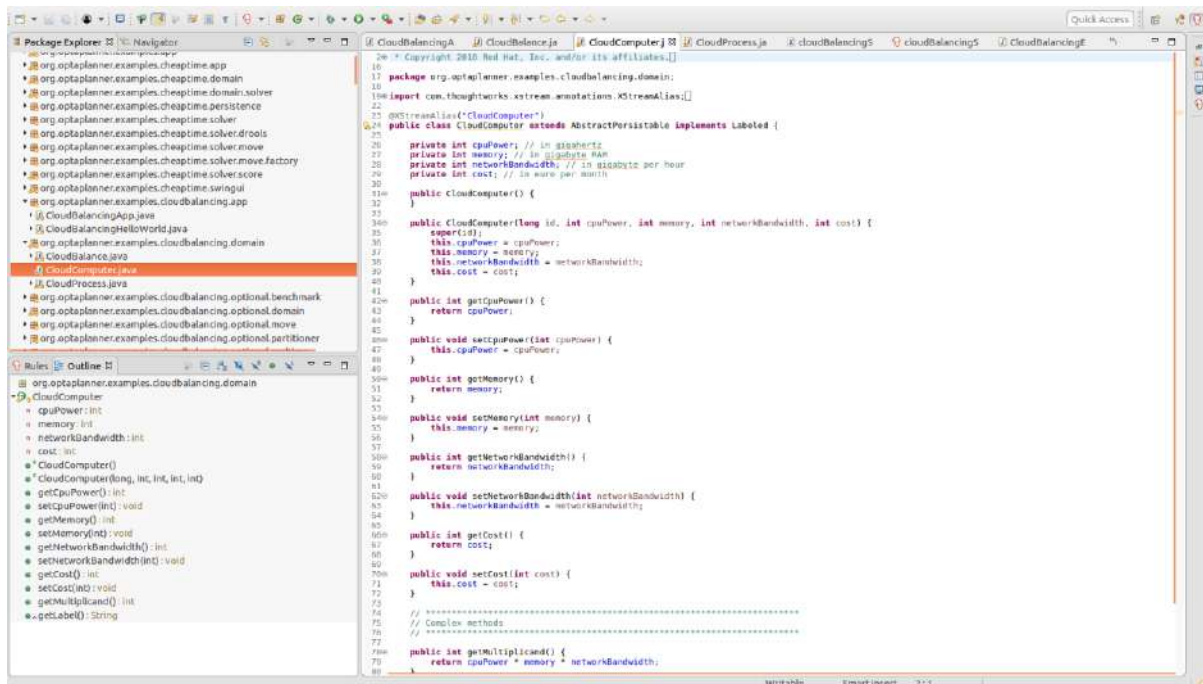


3) Review class diagram; Map against Java classes/objects

CloudBalance.java



CloudComputer.java



CloudProcess.java

```

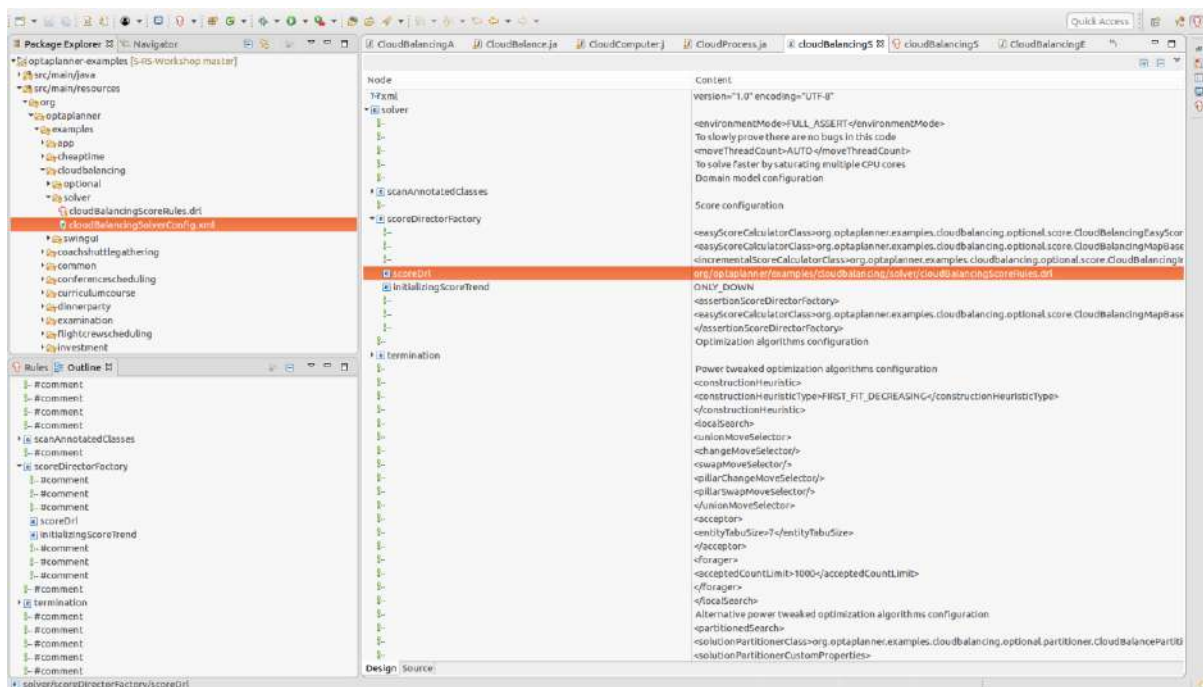
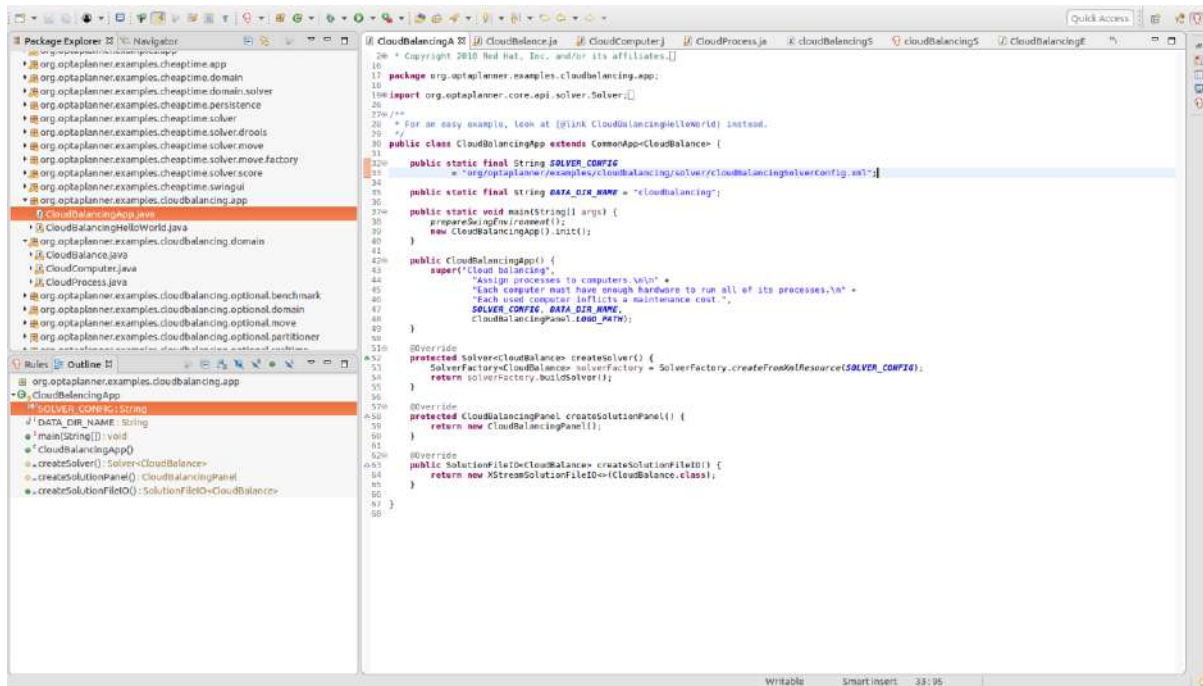
24  * Copyright 2010 Red Hat, Inc. and/or its affiliates.
25
26  package org.optaplanner.examples.cloudbalancing.domain;
27
28  import com.thoughtworks.xstream.annotations.XStreamAlias;
29
30  @PlanningEntity(difficultyComparatorClass = CloudProcessDifficultyComparator.class)
31  @XStreamAlias("CloudProcess")
32  public class CloudProcess extends AbstractPersistable {
33
34      private int requiredCpuPower; // in gigahertz
35      private int requiredMemory; // in gigabyte RAM
36      private int requiredNetworkBandwidth; // in gigabyte per hour
37
38      // planning variables: changes during planning, between score calculations.
39      private CloudComputer computer;
40
41      public CloudProcess() {
42      }
43
44      public CloudProcess(long id, int requiredCpuPower, int requiredMemory, int requiredNetworkBandwidth) {
45          super(id);
46          this.requiredCpuPower = requiredCpuPower;
47          this.requiredMemory = requiredMemory;
48          this.requiredNetworkBandwidth = requiredNetworkBandwidth;
49      }
50
51      public int getRequiredCpuPower() {
52          return requiredCpuPower;
53      }
54
55      public void setRequiredCpuPower(int requiredCpuPower) {
56          this.requiredCpuPower = requiredCpuPower;
57      }
58
59      public int getRequiredMemory() {
60          return requiredMemory;
61      }
62
63      public void setRequiredMemory(int requiredMemory) {
64          this.requiredMemory = requiredMemory;
65      }
66
67      public int getRequiredNetworkBandwidth() {
68          return requiredNetworkBandwidth;
69      }
70
71      public void setRequiredNetworkBandwidth(int requiredNetworkBandwidth) {
72          this.requiredNetworkBandwidth = requiredNetworkBandwidth;
73      }
74
75      @PlanningVariable(valueRangeProviderRefs = {"computerRange"},
76          strengthComparatorClass = CloudComputerStrengthComparator.class)
77      public CloudComputer getComputer() {
78          return computer;
79      }
80
81      public void setComputer(CloudComputer computer) {
82          this.computer = computer;
83      }
84
85      // *****
86      // Complex methods
87      // *****

```

Physical file's location:

Name	Size	Type	Modified	Accessed	Owner	Group	Permissions	MIME Type	Location
5-WS-Workshop	5 items	Folder	17:08	17:08	root	vboxsf	drwxrwx--	inode/directory	/media
5-WS-Workshop1	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
reference	3 items	Folder	17:08	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
optaplanner-examples	4 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
data	22 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
src	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
main	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
java	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
org	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
optaplanner	1 item	Folder	17:08	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
examples	24 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
app	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
cheaptime	5 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
cloudbalancing	5 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
domain	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
CloudBalance.java	2.8 kB	Text	17:09	17:09	root	vboxsf	-rw-rw-r--	text/x-java	/media
CloudComputer.java	2.4 kB	Text	17:09	17:09	root	vboxsf	-rw-rw-r--	text/x-java	/media
CloudProcess.java	3.2 kB	Text	17:09	17:09	root	vboxsf	-rw-rw-r--	text/x-java	/media
optional	6 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
persistence	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
swingui	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
coachshuttleclustering	5 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
common	6 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
conferencescheduling	4 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
curriculumcourse	5 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
dinnerparty	5 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
examination	4 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
flightcrew scheduling	4 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
investment	5 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media
machineassignment	5 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory	/media

4) Review solver configuration file (XML): ***org.optaplanner/examples/cloudbalancing/solver/cloudBalancingSolverConfig.xml***




```

<?xml version="1.0" encoding="UTF-8"?>
<solver>
  <!--<environmentMode>FULL_ASSERT</environmentMode>--><!-- To slowly prove there are no bugs
in this code -->
  <!--<moveThreadCount>AUTO</moveThreadCount>--><!-- To solve faster by saturating multiple
CPU cores -->

  <!-- Domain model configuration -->
  <scanAnnotatedClasses>
    <packageInclude>org.optaplanner.examples.cloudbalancing</packageInclude>
  </scanAnnotatedClasses>

  <!-- Score configuration -->
  <scoreDirectorFactory>
    <!--
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingEasyScoreCalculator</easyScoreCalculatorClass>-->
    <!--
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBasedEasyScoreCalculator</easyScoreCalculatorClass>-->
    <!--
<incrementalScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingIncrementalScoreCalculator</incrementalScoreCalculatorClass>-->

    <scoreDrl>org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl</scoreDrl>
    <initializingScoreTrend>ONLY_DOWN</initializingScoreTrend>
    <!--<assertionScoreDirectorFactory>-->
    <!--
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBasedEasyScoreCalculator</easyScoreCalculatorClass>-->
    <!--</assertionScoreDirectorFactory>-->
    </scoreDirectorFactory>

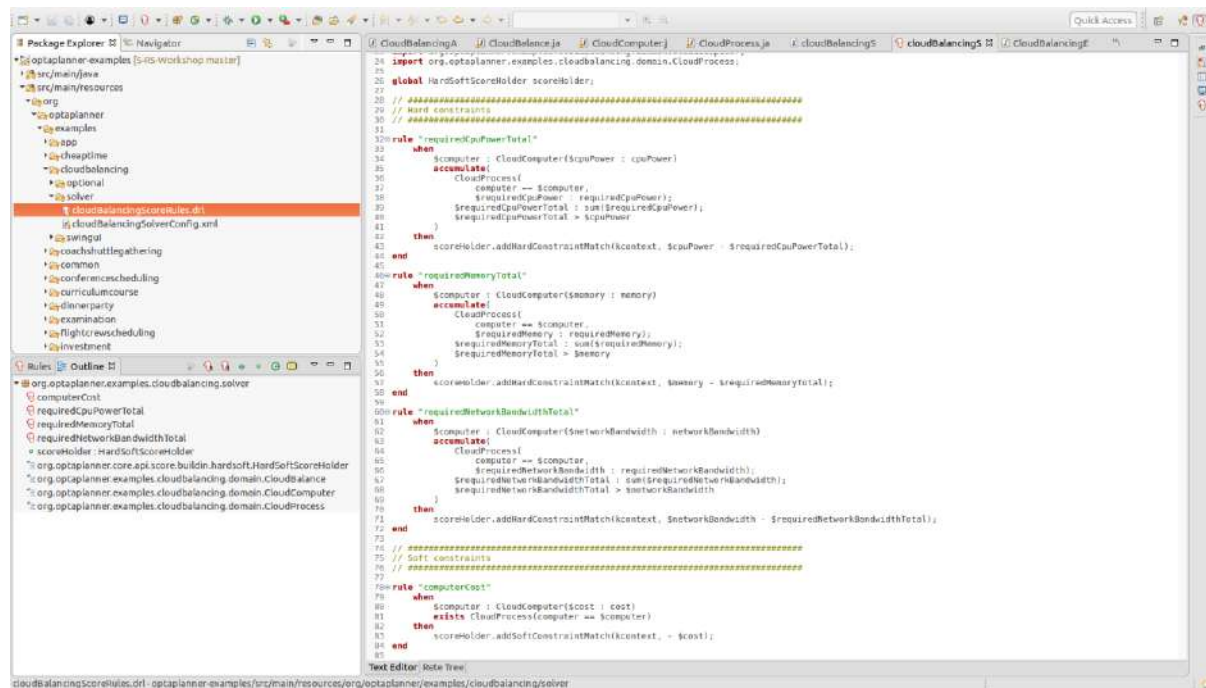
  <!-- Optimization algorithms configuration -->
  <termination>
    <minutesSpentLimit>2</minutesSpentLimit>
  </termination>

  <!-- Power tweaked optimization algorithms configuration -->
  <!--<constructionHeuristic>-->
    <!--<constructionHeuristicType>FIRST_FIT DECREASING</constructionHeuristicType>-->
  <!--</constructionHeuristic>-->
  <!--<localSearch>-->
    <!--<unionMoveSelector>-->
      <!--<changeMoveSelector/>-->
      <!--<swapMoveSelector/>-->
      <!--<pillarChangeMoveSelector/>-->
      <!--<pillarSwapMoveSelector/>-->
    <!--</unionMoveSelector>-->
    <!--<acceptor>-->
      <!--<entityTabuSize>7</entityTabuSize>-->
    <!--</acceptor>-->
    <!--<forager>-->
      <!--<acceptedCountLimit>1000</acceptedCountLimit>-->
    <!--</forager>-->
  <!--</localSearch>-->

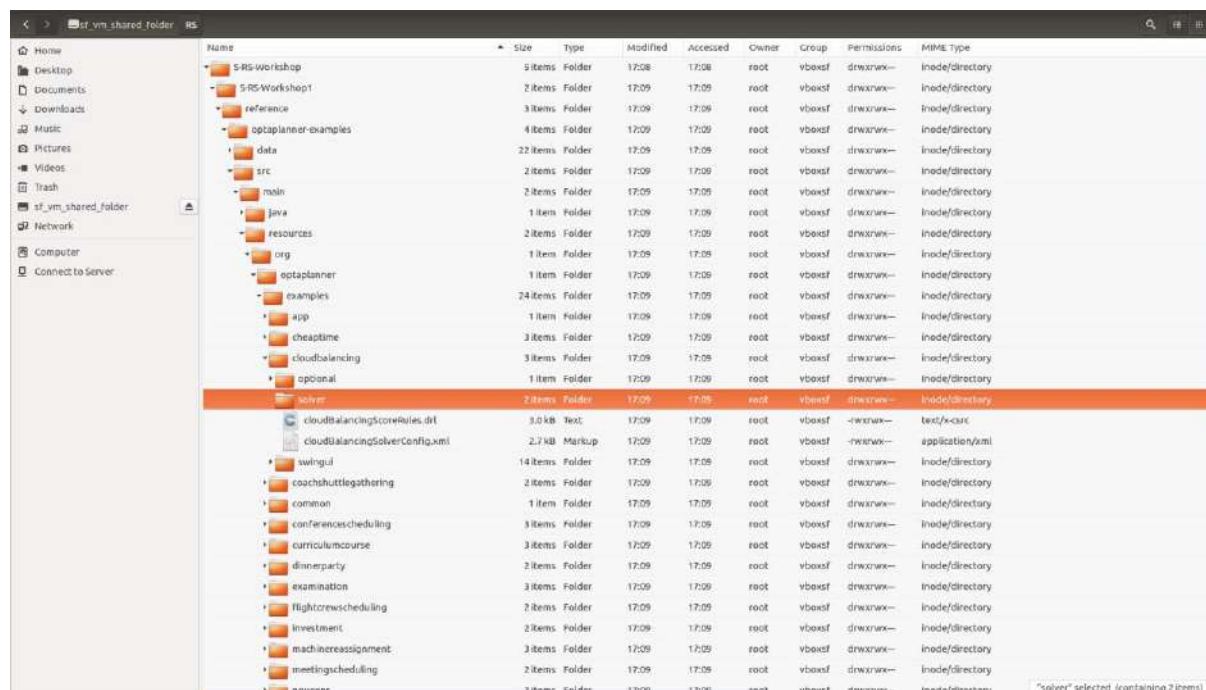
  <!-- Alternative power tweaked optimization algorithms configuration -->
  <!--<partitionedSearch>-->
  <!--
<solutionPartitionerClass>org.optaplanner.examples.cloudbalancing.optional.partitionner.CloudBalancePartitioner</solutionPartitionerClass>-->
    <!--<solutionPartitionerCustomProperties>-->
      <!--<partCount>4</partCount>-->
      <!--<minimumProcessListSize>300</minimumProcessListSize>-->
    <!--</solutionPartitionerCustomProperties>-->
  <!--</partitionedSearch>-->
</solver>

```


5) Review solver score calculation file (Drools Rule):
org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl



Physical file's location:



```

/*
 * Copyright 2010 Red Hat, Inc. and/or its affiliates.
 *
 */

package org.optaplanner.examples.cloudbalancing.solver;
    dialect "java"

import org.optaplanner.core.api.score.buildin.hardsoft.HardSoftScoreHolder;

import org.optaplanner.examples.cloudbalancing.domain.CloudBalance;
import org.optaplanner.examples.cloudbalancing.domain.CloudComputer;
import org.optaplanner.examples.cloudbalancing.domain.CloudProcess;

global HardSoftScoreHolder scoreHolder;

// #####
// Hard constraints
// #####

rule "requiredCpuPowerTotal"
    when
        $computer : CloudComputer($cpuPower : cpuPower)
        accumulate(
            CloudProcess(
                computer == $computer,
                $requiredCpuPower : requiredCpuPower);
            $requiredCpuPowerTotal : sum($requiredCpuPower);
            $requiredCpuPowerTotal > $cpuPower
        )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $cpuPower - $requiredCpuPowerTotal);
    end

rule "requiredMemoryTotal"
    when
        $computer : CloudComputer($memory : memory)
        accumulate(
            CloudProcess(
                computer == $computer,
                $requiredMemory : requiredMemory);
            $requiredMemoryTotal : sum($requiredMemory);
            $requiredMemoryTotal > $memory
        )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $memory - $requiredMemoryTotal);
    end

rule "requiredNetworkBandwidthTotal"
    when
        $computer : CloudComputer($networkBandwidth : networkBandwidth)
        accumulate(
            CloudProcess(
                computer == $computer,
                $requiredNetworkBandwidth : requiredNetworkBandwidth);
            $requiredNetworkBandwidthTotal : sum($requiredNetworkBandwidth);
            $requiredNetworkBandwidthTotal > $networkBandwidth
        )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $networkBandwidth -
$requiredNetworkBandwidthTotal);
    end

// #####
// Soft constraints
// #####

rule "computerCost"
    when
        $computer : CloudComputer($cost : cost)
        exists CloudProcess(computer == $computer)
    then
        scoreHolder.addSoftConstraintMatch(kcontext, - $cost);
    end

```

6) [Optional] { Objective } To compare performance (speed difference) in getting a solution using different score calculation methods, e.g. Drools Rule, Easy Java, Incremental Java.

7) [Optional] Run the solver to obtain performance using Drool Rule. You can run Java program CloudBalanceHelloWorld.java in console mode, to record the **score calculation speed**.

score calculation speed using Drools Rule

```

terminated> CloudBalanceHelloWorld [Java Application] C:\Users\jerry\workspace\drools\java (1 Mar 2018 7:30:25 PM)
19:32:46.618 [main] ] DEBUG LS step (132405), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/1), picked move (CloudProcess-666 {CloudComputer-87} <-> CloudProcess-96 {CloudComputer-355}).
19:32:46.619 [main] ] DEBUG LS step (132406), time spent (119898), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/8), picked move (CloudProcess-651 {CloudComputer-263} <-> CloudProcess-87 {CloudComputer-369}).
19:32:46.724 [main] ] INFO Local Search phase (1) ended: time spent (120003), best score (0hard/-491620soft), score calculation speed (6940/sec), step total (132407).
19:32:46.725 [main] ] INFO Solving ended: time spent (120003), best score (0hard/-491620soft), score calculation speed (8558/sec), phase total (2), environment mode (REPRODUCIBLE).
Solved cloudBalance with 400 computers and 1200 processes:
Process 0 -> Computer 311
Process 1 -> Computer 218
Process 2 -> Computer 379
Process 3 -> Computer 36
Process 4 -> Computer 308
Process 5 -> Computer 299
Process 6 -> Computer 395
Process 7 -> Computer 53
Process 8 -> Computer 232
Process 9 -> Computer 285
Process 10 -> Computer 133
Process 11 -> Computer 213
Process 12 -> Computer 376
Process 13 -> Computer 292
Process 14 -> Computer 272
Process 15 -> Computer 390
Process 16 -> Computer 174
Process 17 -> Computer 103
Process 18 -> Computer 179
Process 19 -> Computer 123
Process 20 -> Computer 42
Process 21 -> Computer 34
Process 22 -> Computer 3
Process 23 -> Computer 8
Process 24 -> Computer 195
Process 25 -> Computer 183
Process 26 -> Computer 99
  
```

19:32:46.618 [main]] DEBUG LS step (132405), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/1), picked move (CloudProcess-666 {CloudComputer-87} <-> CloudProcess-96 {CloudComputer-355}).

19:32:46.619 [main]] DEBUG LS step (132406), time spent (119898), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/8), picked move (CloudProcess-651 {CloudComputer-263} <-> CloudProcess-87 {CloudComputer-369}).

19:32:46.724 [main]] INFO Local Search phase (1) ended: time spent (120003), best score (0hard/-491620soft), **score calculation speed (6940/sec), step total (132407)**.

19:32:46.725 [main]] INFO Solving ended: time spent (120003), best score (0hard/-491620soft), score calculation speed (8558/sec), phase total (2), environment mode (REPRODUCIBLE).

- 8) [Optional] Run the solver to obtain performance using Easy Java. You can run Java program CloudBalanceHelloWorld.java in console mode, to record the **score calculation speed**.

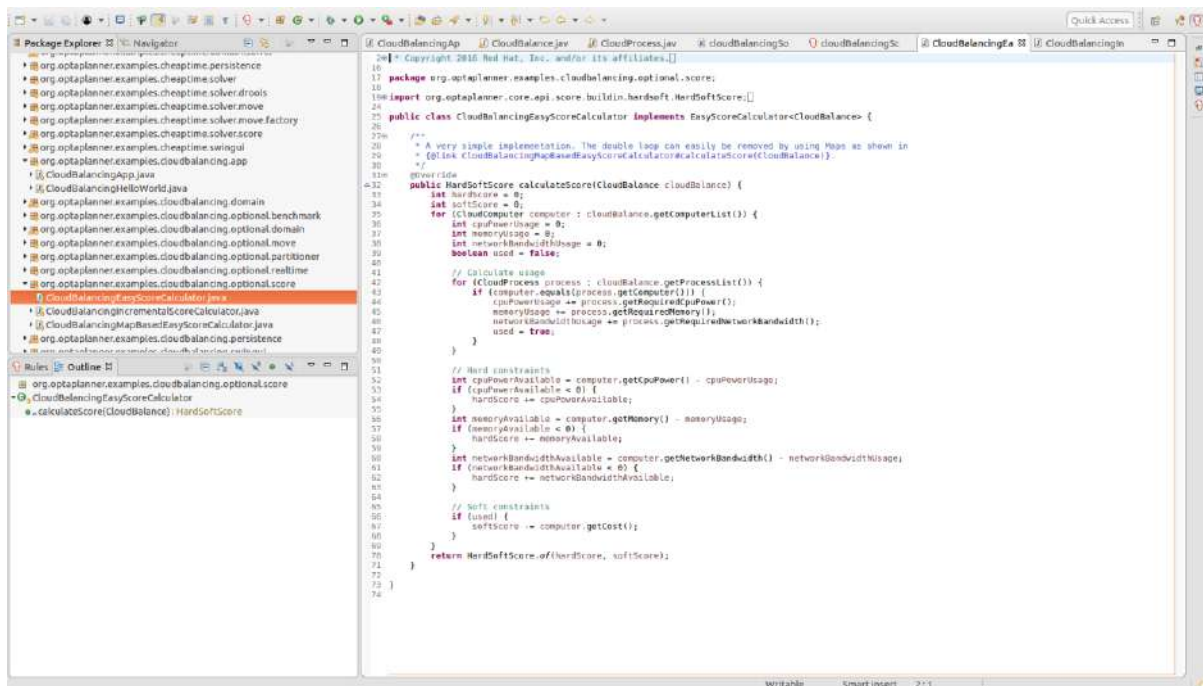
Update **SOLVER_CONFIG** file *cloudBalancingSolverConfig.xml*

From using Drools Rule:

```
<scoreDrl>org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl</scoreDrl>
```

To using Easy Java:

```
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingEasyScoreCalculator</easyScoreCalculatorClass>
```



score calculation speed using Easy Java

```

terminated> CloudBalancingHelloWorld [Java Application] [x86_64-jvm/java-8-openjdk-amd64/bin/java (1 Mar 2019, 7:42:30 PM)]
19:44:38.026 [main] I DEBUG CH step (243), time spent (108620), score (-952init/0hard/-17140soft), selected move count (336), picked move (CloudProcess-1358 {null -> CloudComputer-186}).
19:44:38.072 [main] I DEBUG CH step (243), time spent (108620), score (-952init/0hard/-17140soft), selected move count (336), picked move (CloudProcess-1358 {null -> CloudComputer-186}).
19:44:38.125 [main] I DEBUG CH step (247), time spent (108525), score (-952init/0hard/-17140soft), selected move count (336), picked move (CloudProcess-871 {null -> CloudComputer-300}).
19:44:38.234 [main] I DEBUG CH step (248), time spent (108794), score (-951init/0hard/-17140soft), selected move count (336), picked move (CloudProcess-768 {null -> CloudComputer-302}).
19:44:38.279 [main] I DEBUG CH step (248), time spent (108670), score (-950init/0hard/-17140soft), selected move count (408), picked move (CloudProcess-768 {null -> CloudComputer-377}).
19:44:38.355 [main] I DEBUG CH step (250), time spent (109235), score (-949init/0hard/-17140soft), selected move count (336), picked move (CloudProcess-359 {null -> CloudComputer-371}).
19:44:38.427 [main] I DEBUG CH step (251), time spent (108737), score (-948init/0hard/-17020soft), selected move count (408), picked move (CloudProcess-312 {null -> CloudComputer-274}).
19:44:38.508 [main] I DEBUG CH step (252), time spent (110908), score (-947init/0hard/-17020soft), selected move count (408), picked move (CloudProcess-258 {null -> CloudComputer-251}).
19:44:38.582 [main] I DEBUG CH step (253), time spent (110581), score (-946init/0hard/-17020soft), selected move count (288), picked move (CloudProcess-228 {null -> CloudComputer-373}).
19:44:38.567 [main] I DEBUG CH step (254), time spent (110707), score (-945init/0hard/-17020soft), selected move count (274), picked move (CloudProcess-1183 {null -> CloudComputer-156}).
19:44:38.643 [main] I DEBUG CH step (255), time spent (111843), score (-944init/0hard/-17020soft), selected move count (312), picked move (CloudProcess-1818 {null -> CloudComputer-351}).
19:44:38.670 [main] I DEBUG CH step (256), time spent (111428), score (-943init/0hard/-17020soft), selected move count (288), picked move (CloudProcess-984 {null -> CloudComputer-127}).
19:44:38.753 [main] I DEBUG CH step (257), time spent (111753), score (-942init/0hard/-17020soft), selected move count (427), picked move (CloudProcess-922 {null -> CloudComputer-43}).
19:44:38.820 [main] I DEBUG CH step (258), time spent (111520), score (-941init/0hard/-17020soft), selected move count (156), picked move (CloudProcess-915 {null -> CloudComputer-130}).
19:44:38.921 [main] I DEBUG CH step (259), time spent (112331), score (-940init/0hard/-17020soft), selected move count (200), picked move (CloudProcess-805 {null -> CloudComputer-274}).
19:44:39.046 [main] I DEBUG CH step (260), time spent (112648), score (-939init/0hard/-17020soft), selected move count (305), picked move (CloudProcess-783 {null -> CloudComputer-277}).
19:44:39.124 [main] I DEBUG CH step (261), time spent (112824), score (-938init/0hard/-17020soft), selected move count (298), picked move (CloudProcess-729 {null -> CloudComputer-112}).
19:44:39.191 [main] I DEBUG CH step (262), time spent (113491), score (-937init/0hard/-17020soft), selected move count (307), picked move (CloudProcess-864 {null -> CloudComputer-356}).
19:44:39.263 [main] I DEBUG CH step (263), time spent (112723), score (-936init/0hard/-17020soft), selected move count (118), picked move (CloudProcess-598 {null -> CloudComputer-230}).
19:44:39.316 [main] I DEBUG CH step (264), time spent (114416), score (-935init/0hard/-17020soft), selected move count (312), picked move (CloudProcess-355 {null -> CloudComputer-251}).
19:44:39.341 [main] I DEBUG CH step (265), time spent (114943), score (-934init/0hard/-17020soft), selected move count (312), picked move (CloudProcess-467 {null -> CloudComputer-251}).
19:44:39.353 [main] I DEBUG CH step (266), time spent (115753), score (-933init/0hard/-17020soft), selected move count (288), picked move (CloudProcess-857 {null -> CloudComputer-373}).
19:44:39.409 [main] I DEBUG CH step (267), time spent (116205), score (-932init/0hard/-17020soft), selected move count (306), picked move (CloudProcess-345 {null -> CloudComputer-350}).
19:44:39.475 [main] I DEBUG CH step (268), time spent (117855), score (-931init/0hard/-17020soft), selected move count (381), picked move (CloudProcess-381 {null -> CloudComputer-287}).
19:44:39.508 [main] I DEBUG CH step (269), time spent (117698), score (-930init/0hard/-17020soft), selected move count (304), picked move (CloudProcess-218 {null -> CloudComputer-230}).
19:44:39.516 [main] I DEBUG CH step (270), time spent (117912), score (-929init/0hard/-17020soft), selected move count (204), picked move (CloudProcess-49 {null -> CloudComputer-200}).
19:44:39.579 [main] I DEBUG CH step (271), time spent (118179), score (-928init/0hard/-17020soft), selected move count (251), picked move (CloudProcess-42 {null -> CloudComputer-367}).
19:44:39.561 [main] I DEBUG CH step (272), time spent (118761), score (-927init/0hard/-16100soft), selected move count (408), picked move (CloudProcess-819 {null -> CloudComputer-42}).
19:44:39.645 [main] I DEBUG CH step (273), time spent (119745), score (-926init/0hard/-16100soft), selected move count (311), picked move (CloudProcess-426 {null -> CloudComputer-42}).
19:44:39.687 [main] I DEBUG CH step (274), time spent (119587), score (-925init/0hard/-16100soft), selected move count (313), picked move (CloudProcess-247 {null -> CloudComputer-42}).
19:44:39.698 [main] I DEBUG CH step (275), time spent (120000), score (-924init/-1hard/-181010soft), selected move count (297), picked move (CloudProcess-163 {null -> CloudComputer-226}).
19:44:39.632 [main] I INFO Construction Heuristic phase (0) ended: time spent (120032), best score (-924init/-1hard/-181010soft), score calculation speed (775/sec), step total (276).
19:44:39.666 [main] I INFO Solving ended: time spent (120066), best score (-924init/-1hard/-181010soft), score calculation speed (774/sec), phase total (2), environment mode (REPRODUCIBLE).

Solved cloudBalance with 488 computers and 1280 processes:
Process 0 -> null
Process 1 -> null
Process 2 -> null
Process 3 -> null
Process 4 -> Computer 159
Process 5 -> null
Process 6 -> null
Process 7 -> Computer 36
Process 8 -> null
Process 9 -> null
Process 10 -> null
Process 11 -> null
Process 12 -> null
Process 13 -> null
Process 14 -> Computer 324
Process 15 -> null
Process 16 -> null
Process 17 -> null
Process 18 -> null
Process 19 -> Computer 110
Process 20 -> null
Process 21 -> null
Process 22 -> null
Process 23 -> null
Process 24 -> Computer 236
Process 25 -> null
Process 26 -> null

```

19:44:38.187 [main] DEBUG CH step (274), time spent (119587), score (-925init/0hard/-181010soft), selected move count (313), picked move (CloudProcess-247 {null -> CloudComputer-42}).

19:44:38.600 [main] DEBUG CH step (275), time spent (120000), score (-924init/-1hard/-181010soft), selected move count (297), picked move (CloudProcess-163 {null -> CloudComputer-226}).

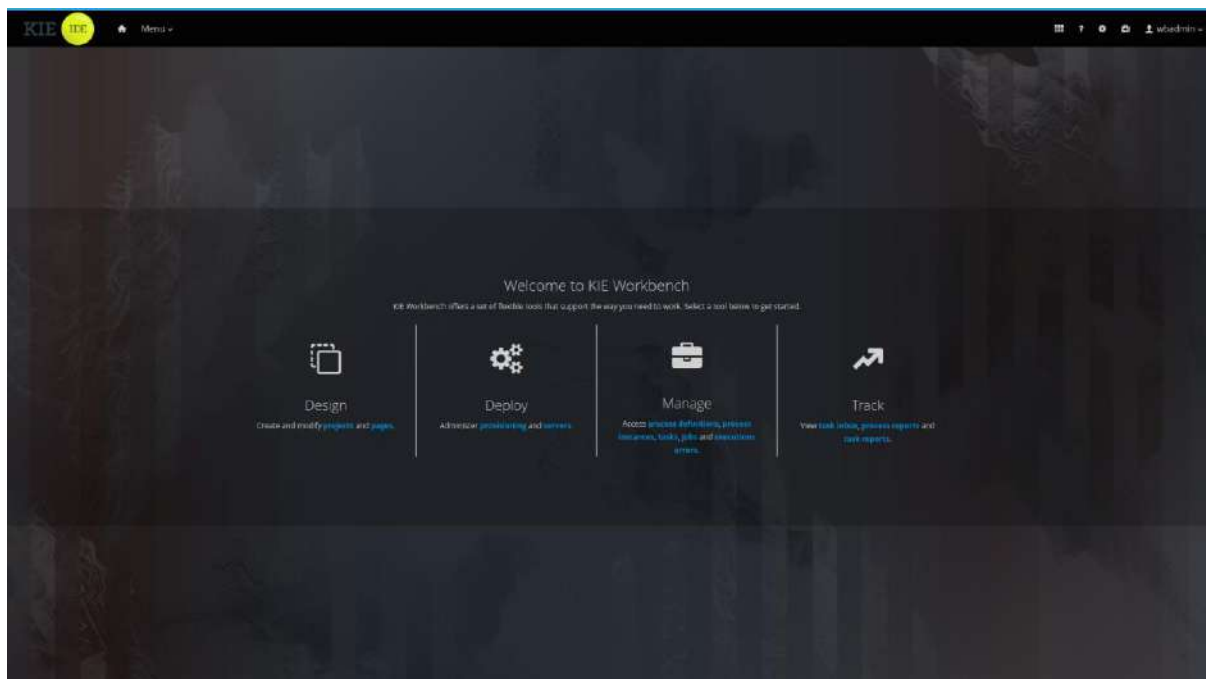
19:44:38.632 [main] INFO Construction Heuristic phase (0) ended: time spent (120032), best score (-924init/-1hard/-181010soft), **score calculation speed (775/sec), step total (276).**

19:44:38.666 [main] INFO Solving ended: time spent (120066), best score (-924init/-1hard/-181010soft), score calculation speed (774/sec), phase total (2), environment mode (REPRODUCIBLE).

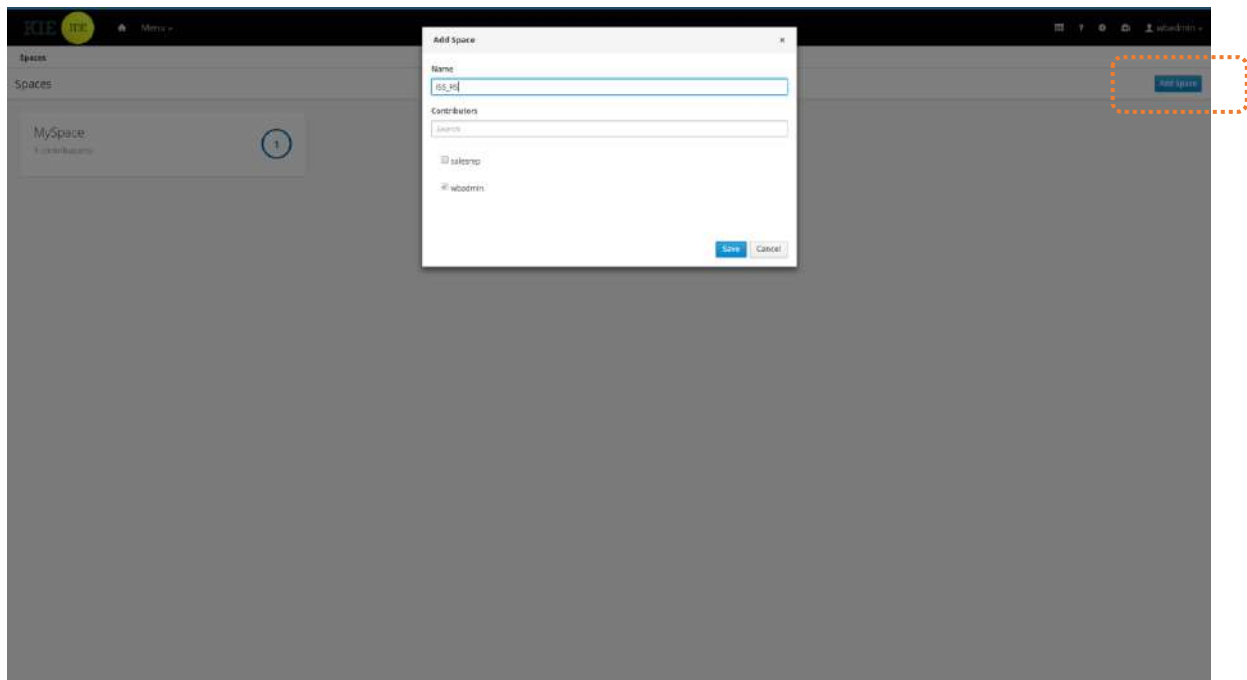
9) [Optional] { Challenge } score calculation speed using Incremental Java;

1.1.2. Cloud Balance Solver [KIE Workbench] v1.0.0

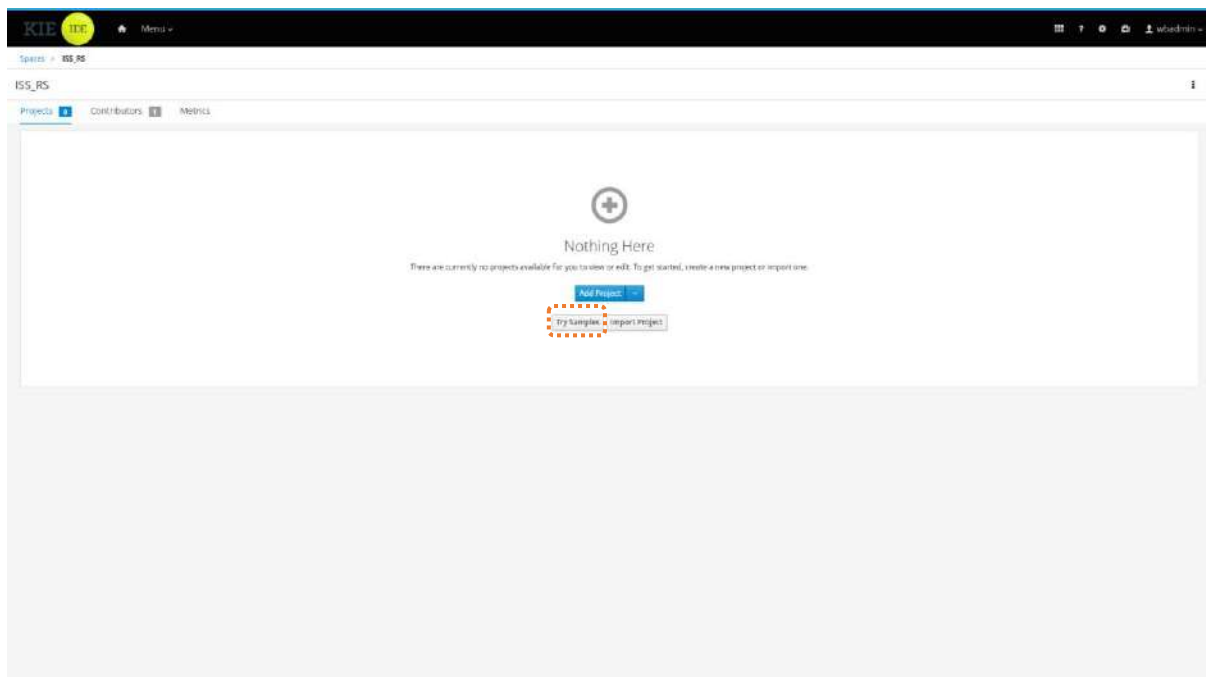
- 1) Install RESTful API tool: Postman
Refer to Annex 6;
- 2) Start KIE Workbench

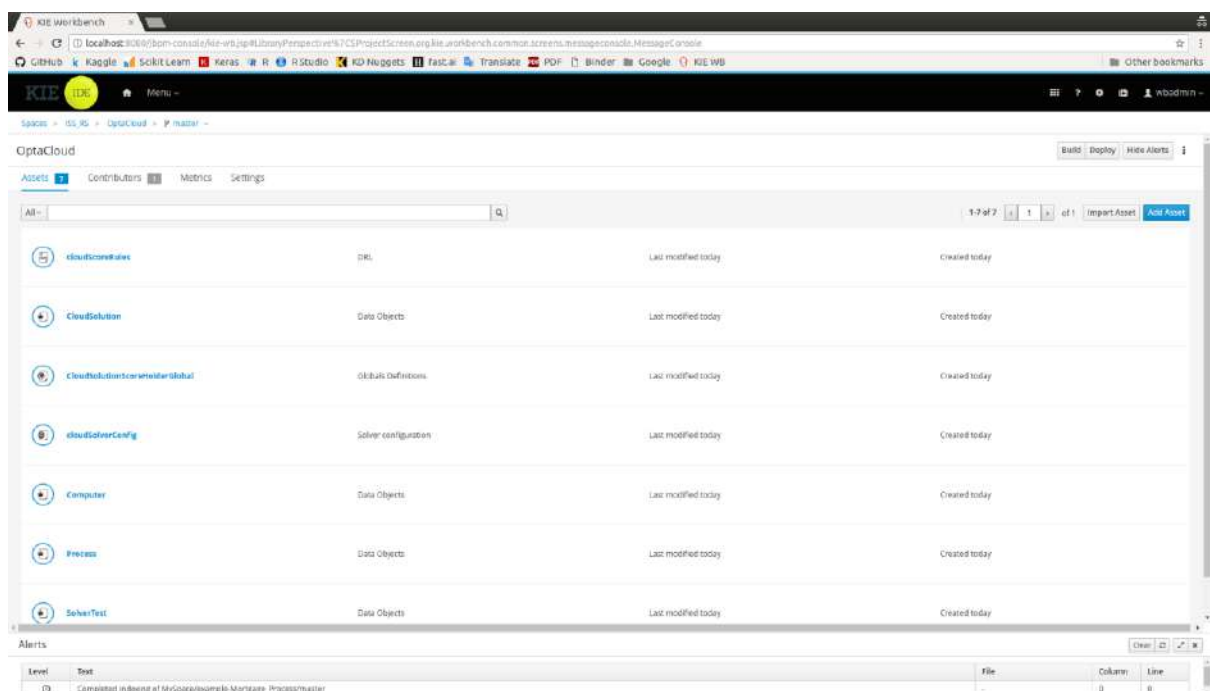
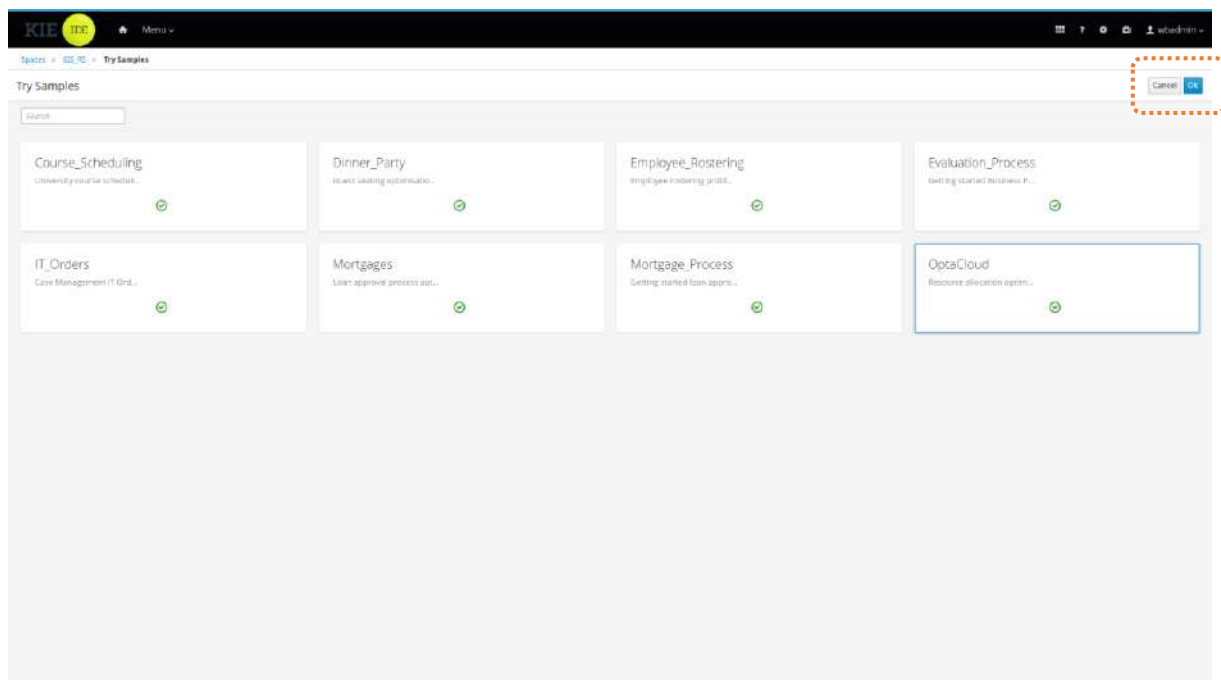


3) Create workspace: *ISS_RS*



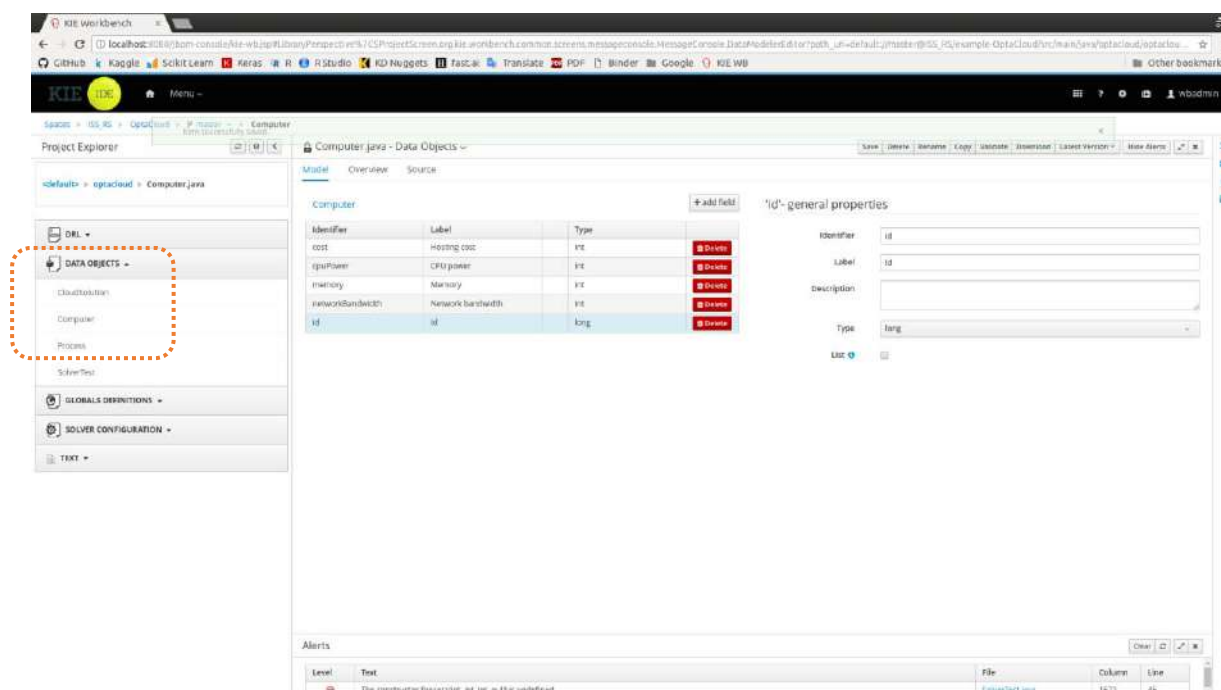
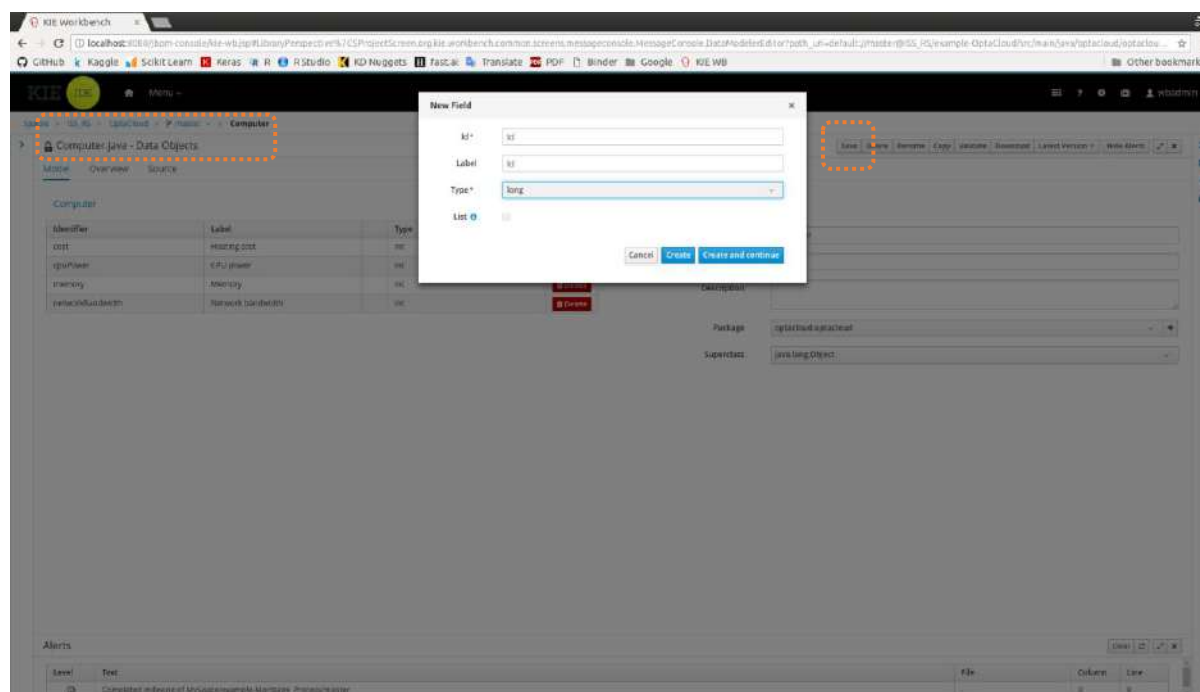
4) Click **Try Samples**



5) Select **OptaCloud**; Click **OK**

Above: try-sample project **OptaCloud_ISS_RS** in KIE Workbench

- 6) [Data Object] Add field: **id** (type: **long**) to Data Objects: **Computer**, **Process**, **CloudSolution**; Save all three objects;



Process Explorer

Process.java - Data Objects

Identifier	Label	Type
computer	Computer	org.optaplanner.solver.computer.Computer
requiredCPUpower	Required CPU power	int
requiredMemory	Required memory	int
requiredNetworkBandwidth	Required network bandwidth	int
id	id	long

'id' general properties

Identifier: id

Label: id

Description:

Type: long

Alerts

Level	Text	File	Column	Line
Info	Build of module 'OptaCloud' (requested by workflow) completed. Build: SUCCESSFUL	-	0	0

CloudSolution Explorer

CloudSolution.java - Data Objects

Identifier	Label	Type
computerList	Computer list	org.optaplanner.solver.computer.ComputerList
processList	Process list	org.optaplanner.solver.process.ProcessList
id	id	long

'id' general properties

Identifier: id

Label: id

Description:

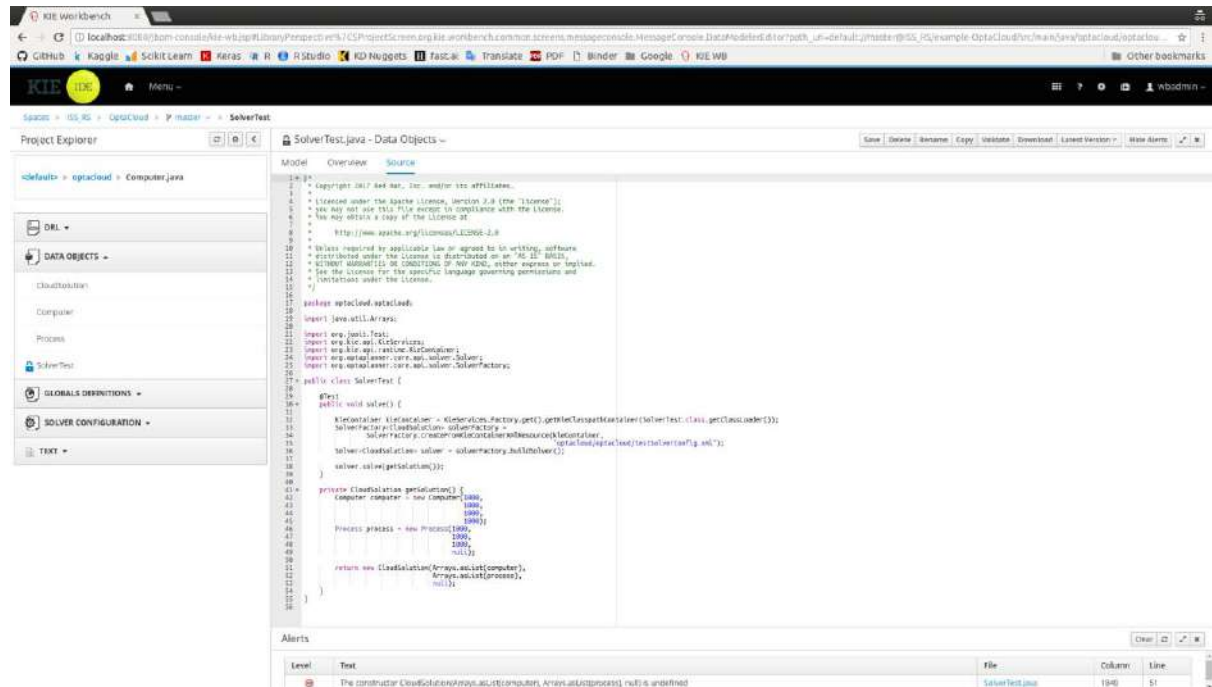
Type: long

Alerts

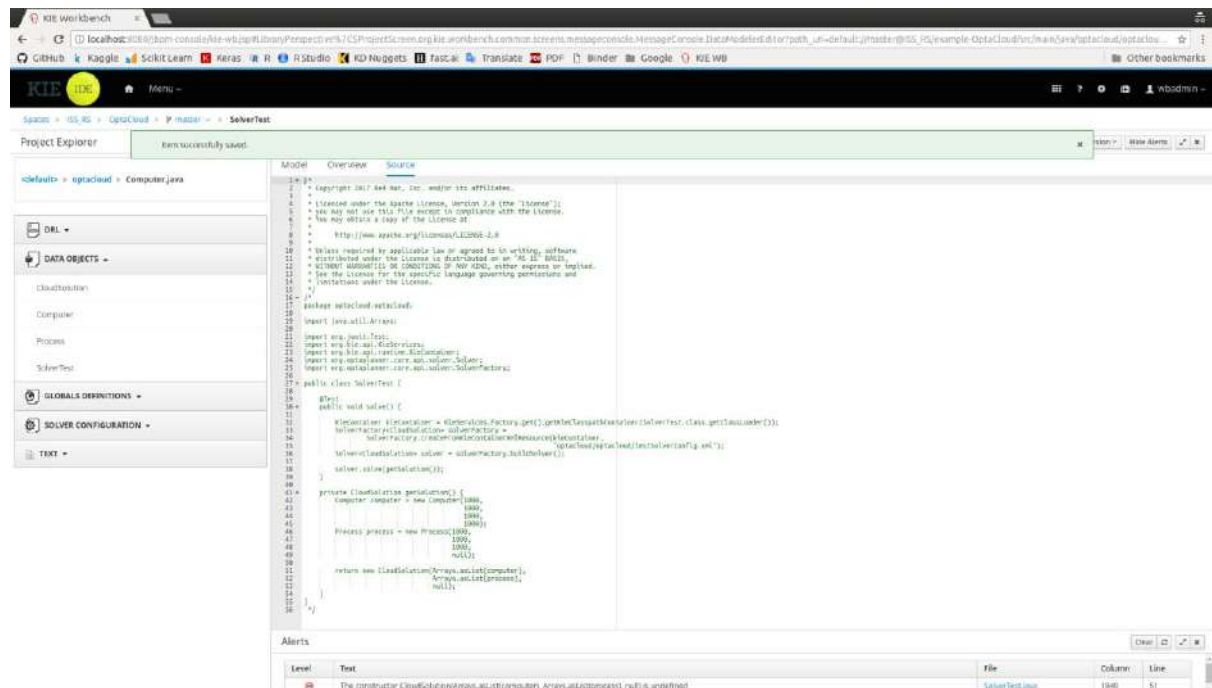
Level	Text	File	Column	Line
Error	The constructor ProcessList(int, int) is undefined	ServerTest.java	1573	45

Above: domain objects based on class diagram

7) [Data Object] To prevent Build/Deploy error, comment all code in Java script:
example-OptaCloud/src/test/java/optacloud/optacloud/SolverTest.java

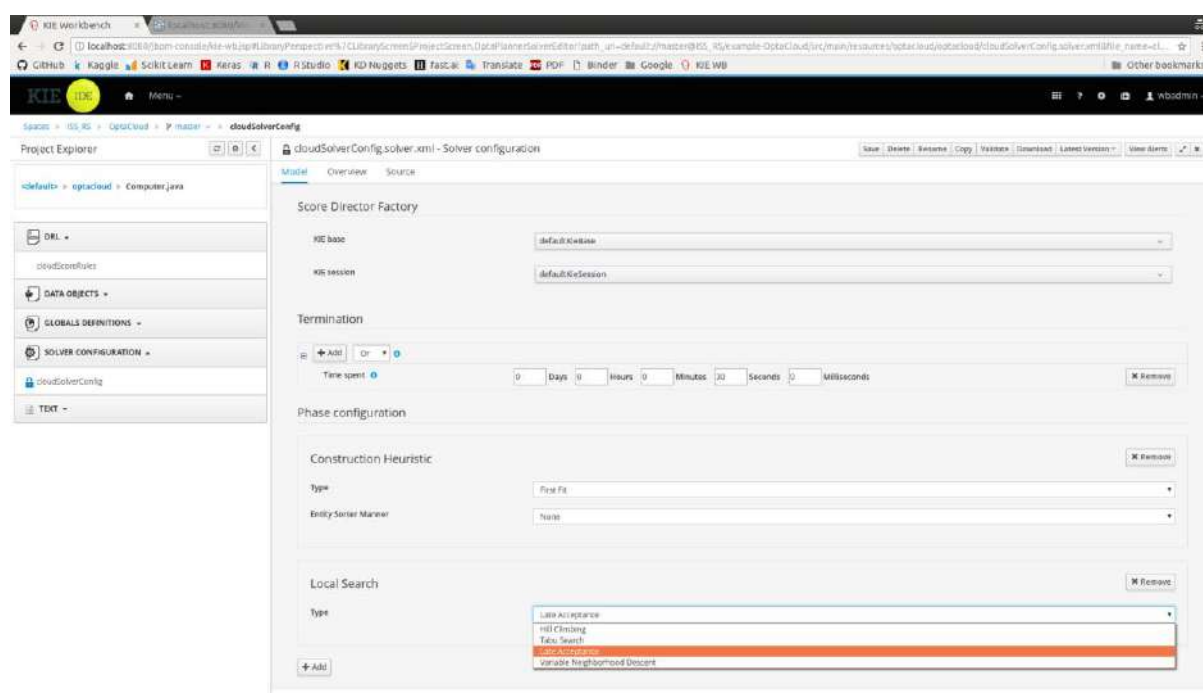
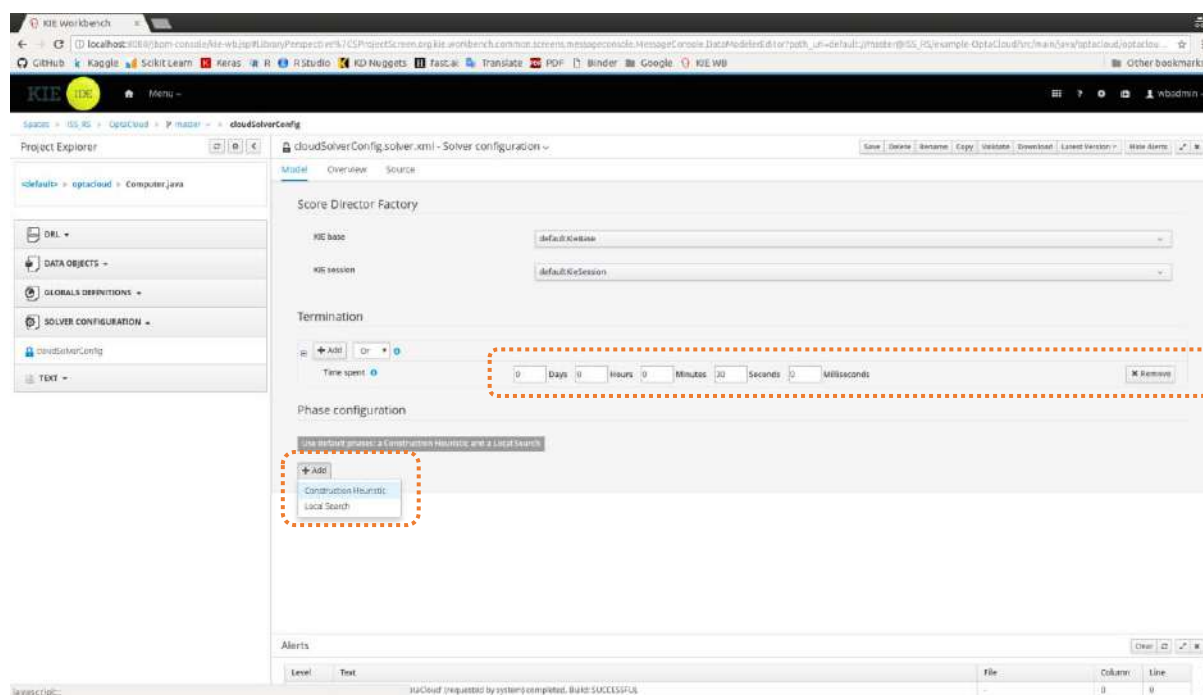


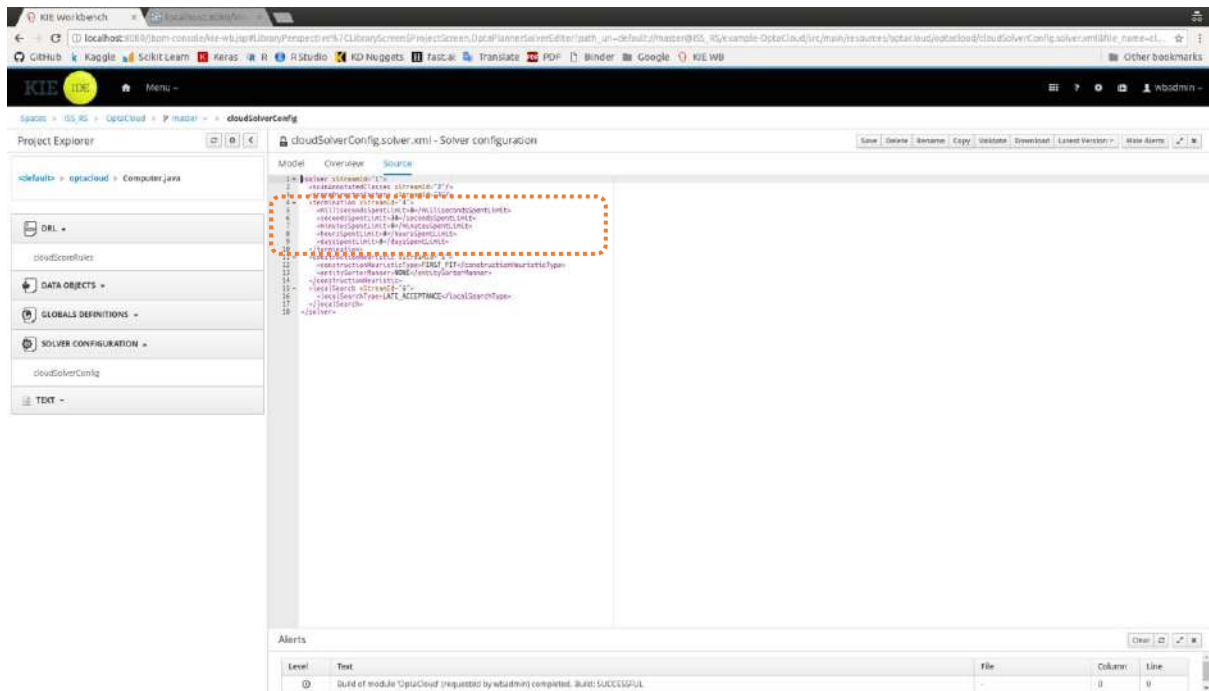
Save



Above: unused **SolverTest** java program

8) [Solver Configuration] Update





Above: solver configuration (xml)

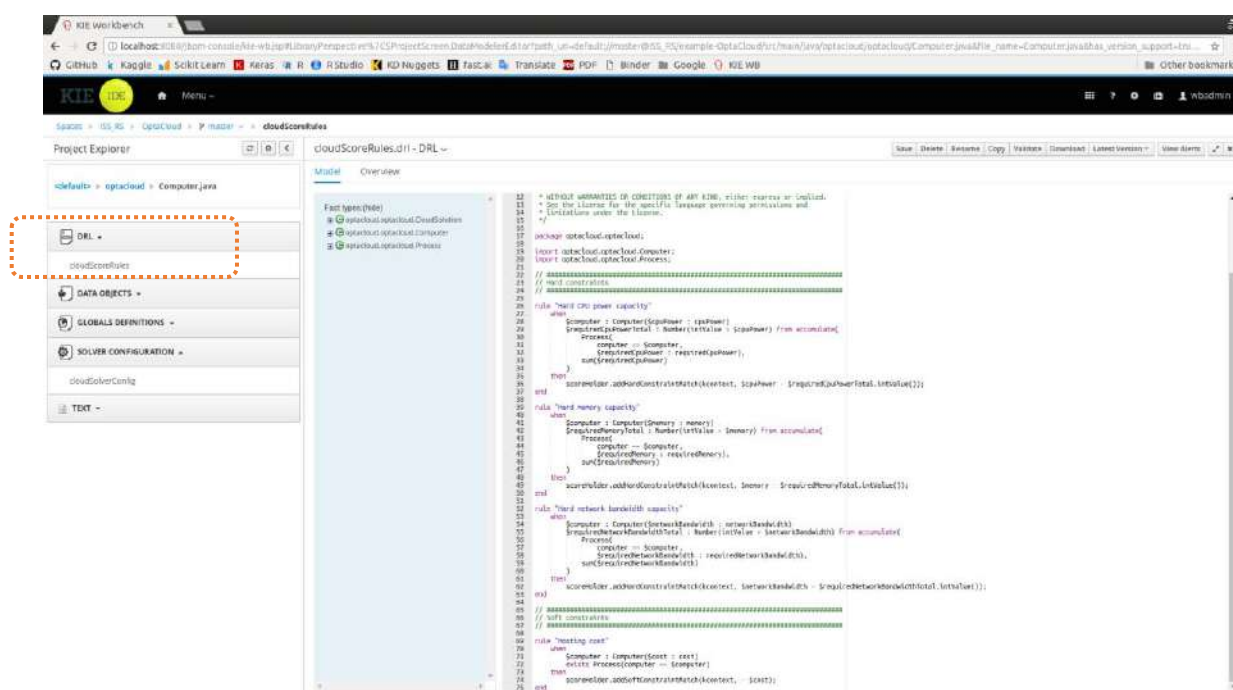
[example-OptaCloud/src/main/resources/optacloud/optacloud/cloudSolverConfig.solver.xml](https://github.com/robertodiaz/example-OptaCloud/blob/master/src/main/resources/optacloud/optacloud/cloudSolverConfig.solver.xml)

```

<solver xStreamId="1">
  <scanAnnotatedClasses xStreamId="2"/>
  <scoreDirectorFactory xStreamId="3"/>
  <termination xStreamId="4">
    <millisecondsSpentLimit>0</millisecondsSpentLimit>
    <secondsSpentLimit>30</secondsSpentLimit>
    <minutesSpentLimit>0</minutesSpentLimit>
    <hoursSpentLimit>0</hoursSpentLimit>
    <daysSpentLimit>0</daysSpentLimit>
  </termination>
  <constructionHeuristic xStreamId="5">
    <constructionHeuristicType>FIRST_FIT</constructionHeuristicType>
    <entitySorterManner>NONE</entitySorterManner>
  </constructionHeuristic>
  <localSearch xStreamId="6">
    <localSearchType>LATE_ACCEPTANCE</localSearchType>
  </localSearch>
</solver>

```

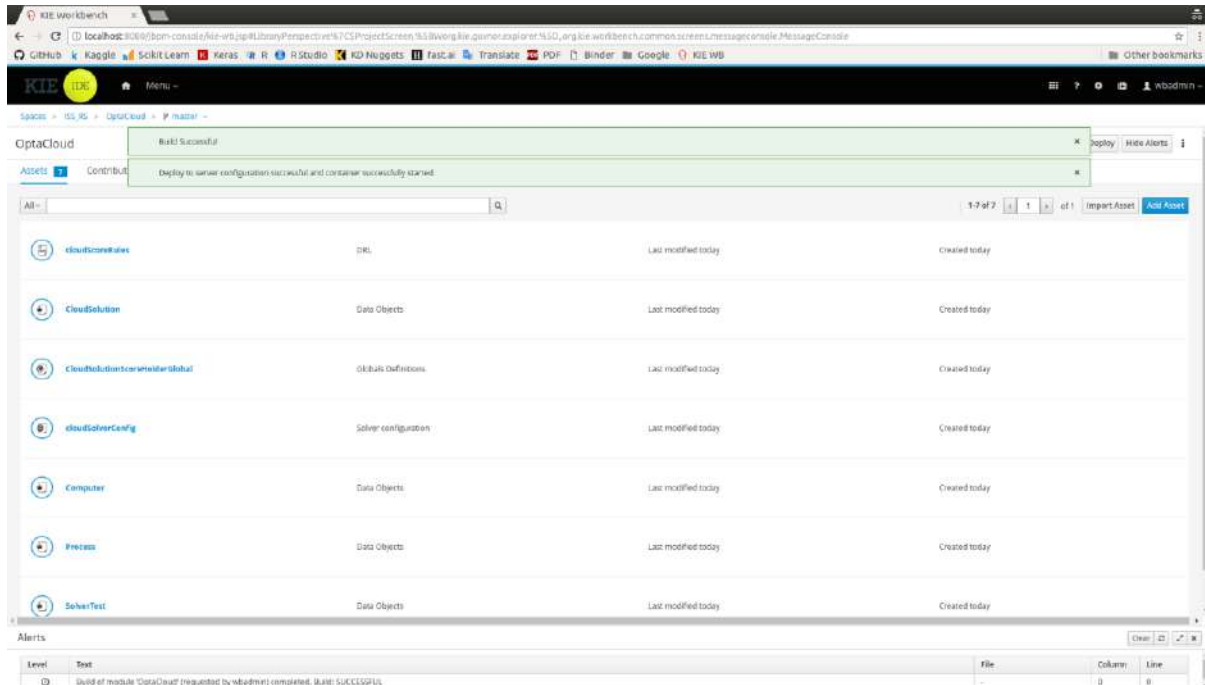
9) [DRL] Constraint Definition / Score Calculation using Drools Rule



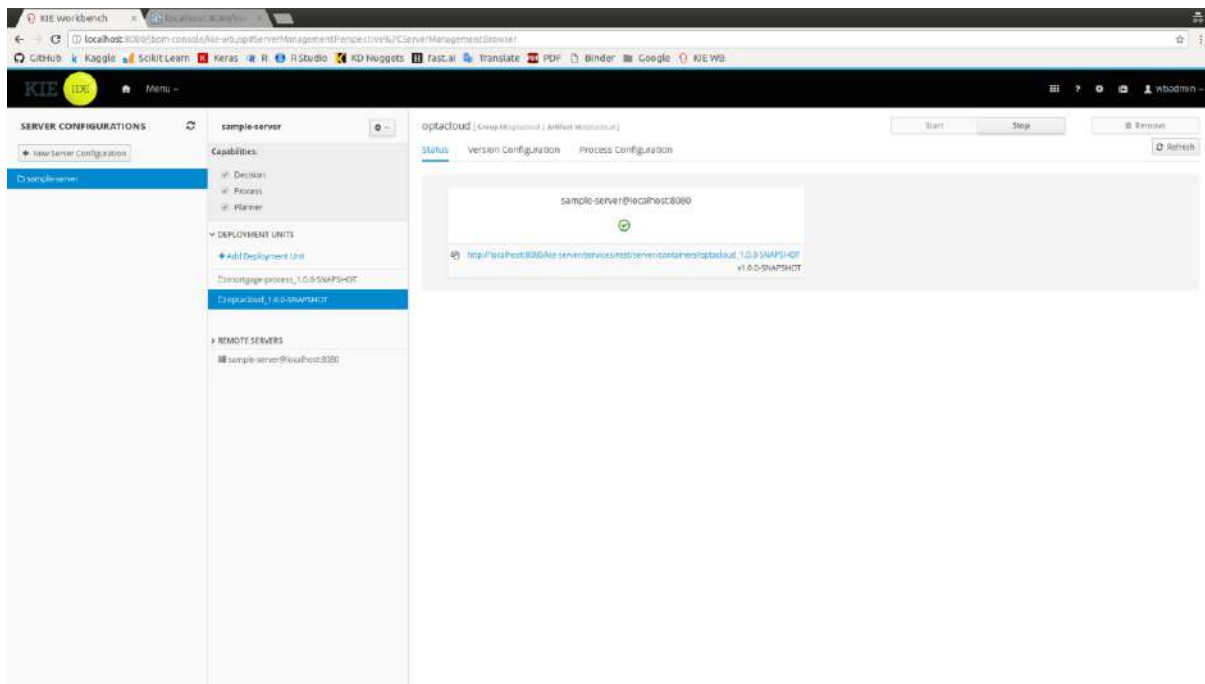
Above: constraints using Drools rule

10)[Deploy] Deploy OptaCloud solver to server

😊 Remember to **Save** all modified assets before build/deploy.



Deploy to KIE Server *optacloud_1.0.0-SNAPSHOT*



Above: solver deployment to KIE Server

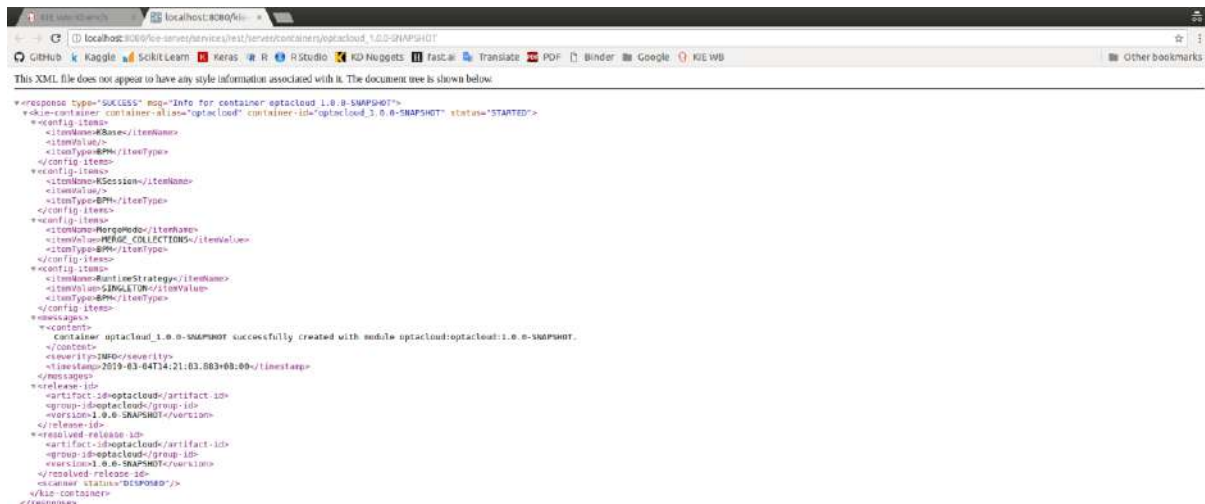
http://localhost:8080/kie-server/services/rest/server/containers/optacloud_1.0.0-SNAPSHOT

Username: kieserver

Password: kieserver1!



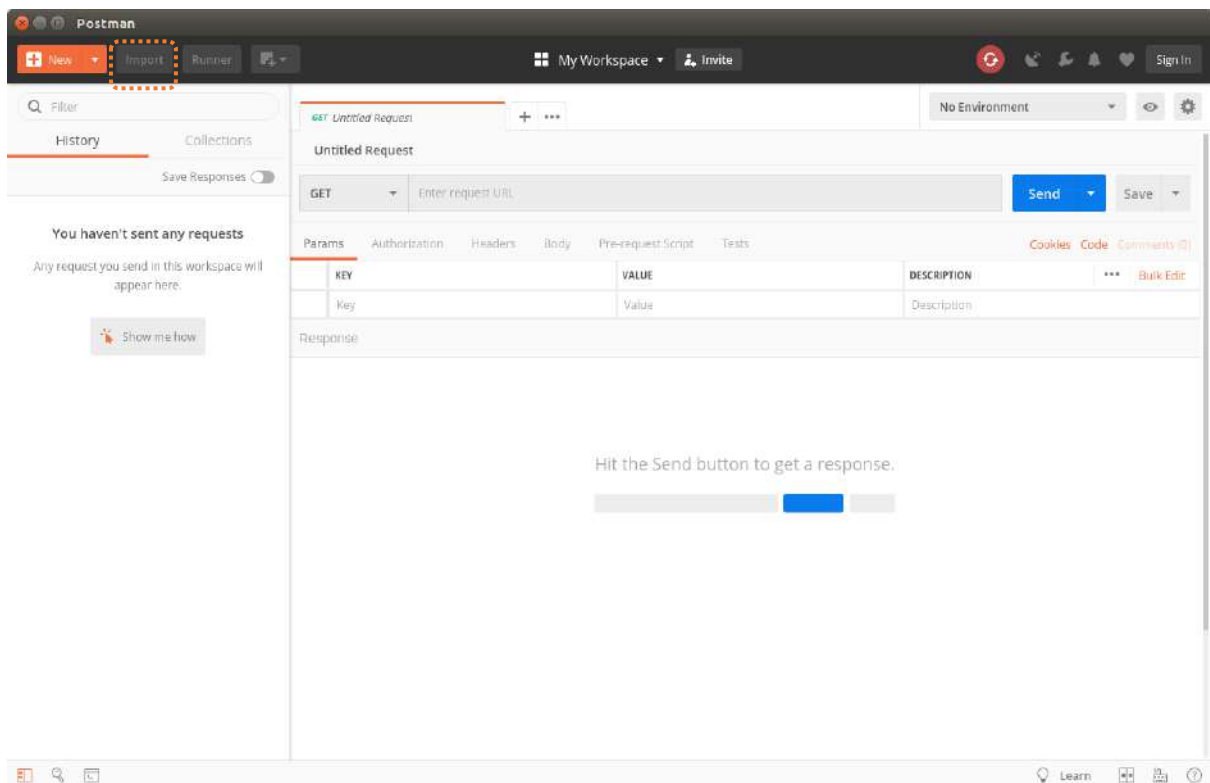
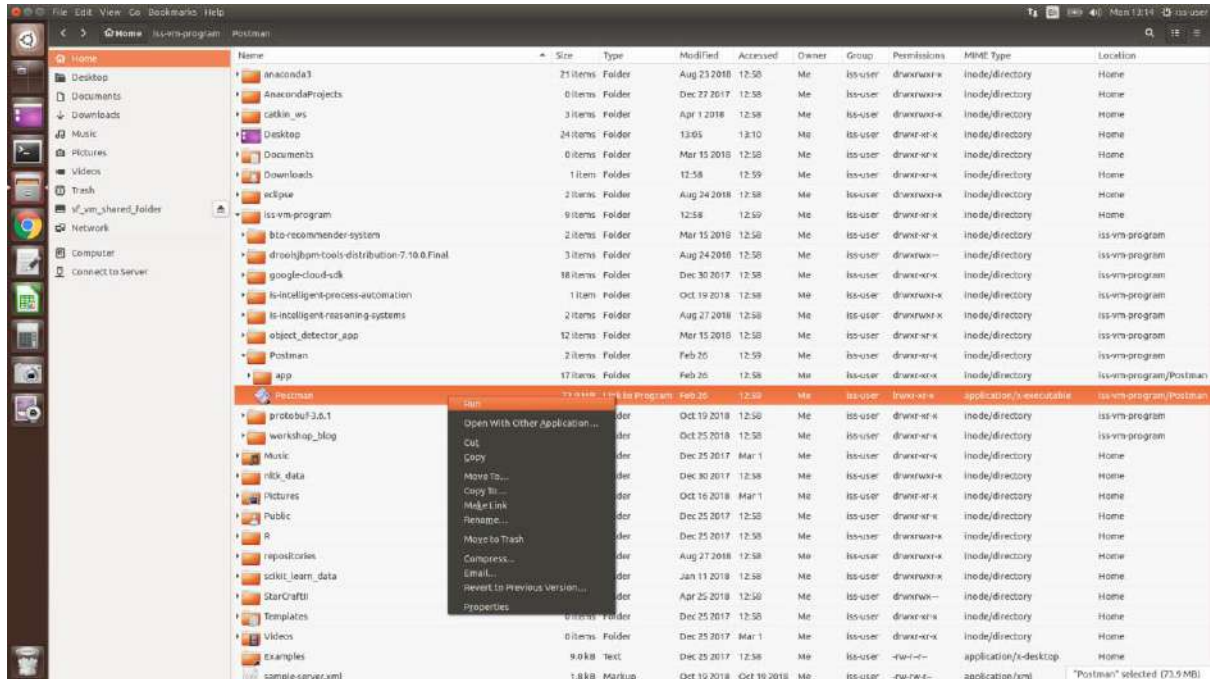
Above: solver web service end point



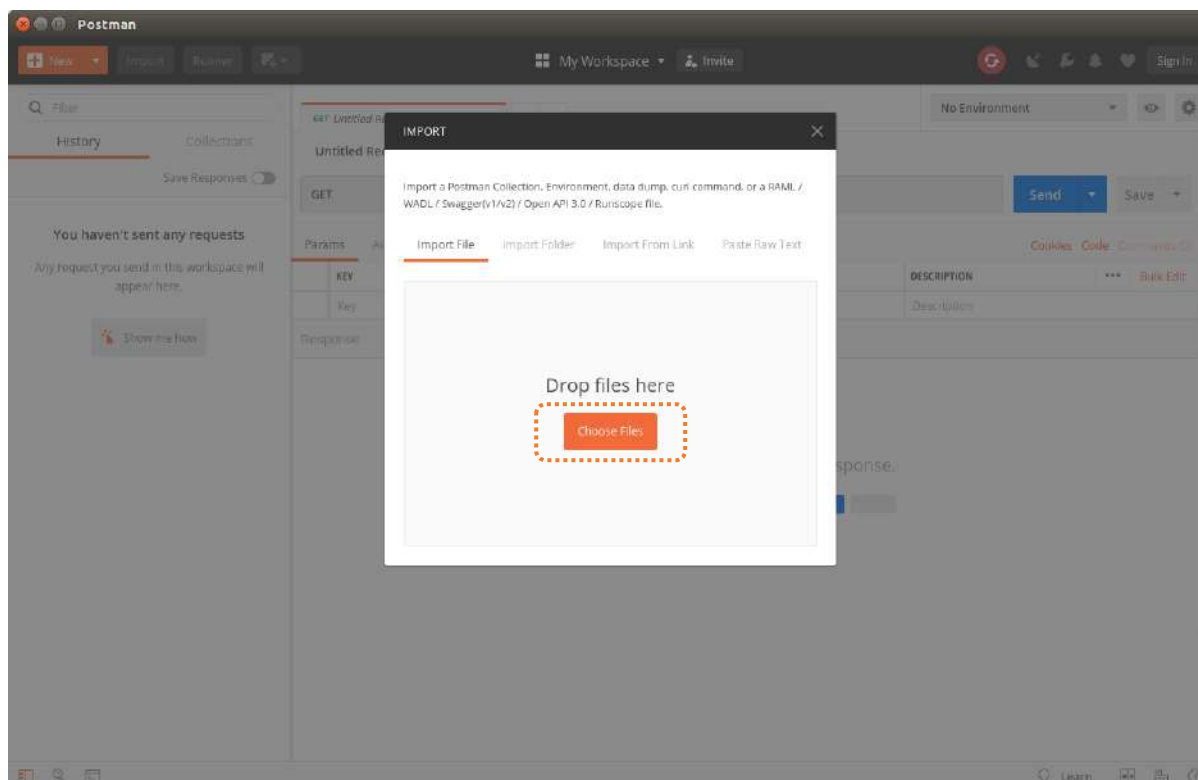
```
<response type="SUCCESS" msg="Info for container optacloud_1.0.0-SNAPSHOT">
<kie-container container-alias="optacloud" container-id="optacloud_1.0.0-
SNAPSHOT" status="STARTED">
<config-items>
<itemName>KBase</itemName>
<itemValue/>
<itemType>BPM</itemType>
</config-items>
<config-items>
<itemName>KSession</itemName>
<itemValue/>
<itemType>BPM</itemType>
</config-items>
<config-items>
<itemName>MergeMode</itemName>
<itemValue>MERGE_COLLECTIONS</itemValue>
<itemType>BPM</itemType>
</config-items>
<config-items>
<itemName>RuntimeStrategy</itemName>
<itemValue>SINGLETON</itemValue>
<itemType>BPM</itemType>
</config-items>
<messages>
<content>
Container optacloud_1.0.0-SNAPSHOT successfully created with module
optacloud:optacloud:1.0.0-SNAPSHOT.
</content>
<severity>INFO</severity>
<timestamp>2019-03-04T14:21:03.883+08:00</timestamp>
</messages>
<release-id>
<artifact-id>optacloud</artifact-id>
<group-id>optacloud</group-id>
<version>1.0.0-SNAPSHOT</version>
</release-id>
<resolved-release-id>
<artifact-id>optacloud</artifact-id>
<group-id>optacloud</group-id>
<version>1.0.0-SNAPSHOT</version>
</resolved-release-id>
<scanner status="DISPOSED"/>
</kie-container>
</response>
```

11)[API] Use RESTful API to interact with deployed OptaCloud solver

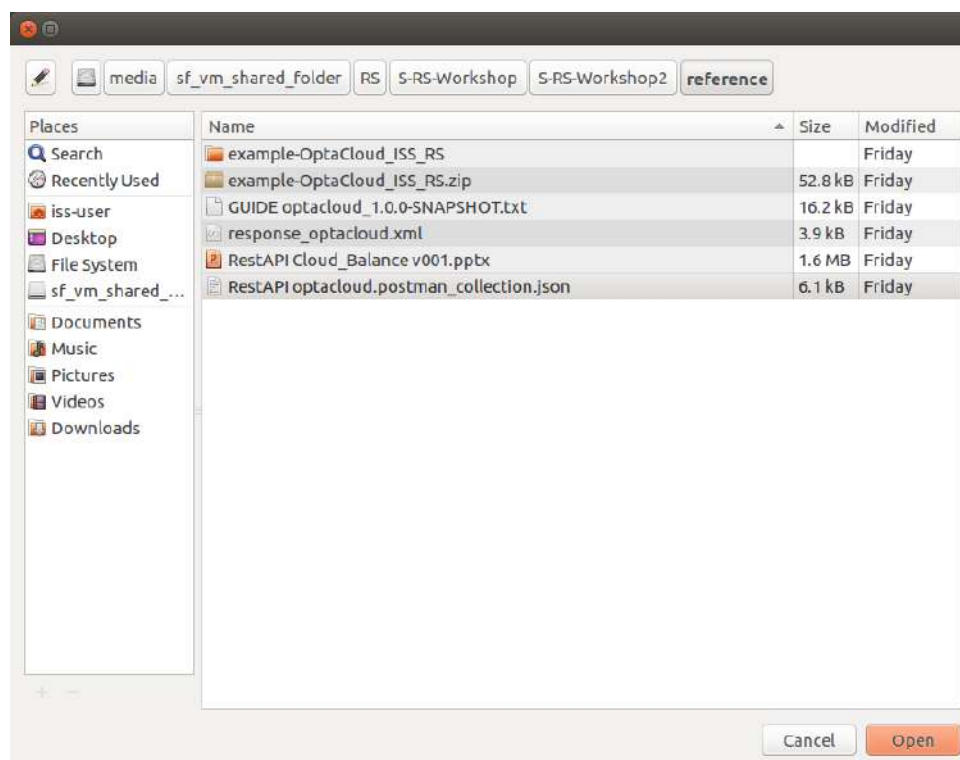
Start tool Postman

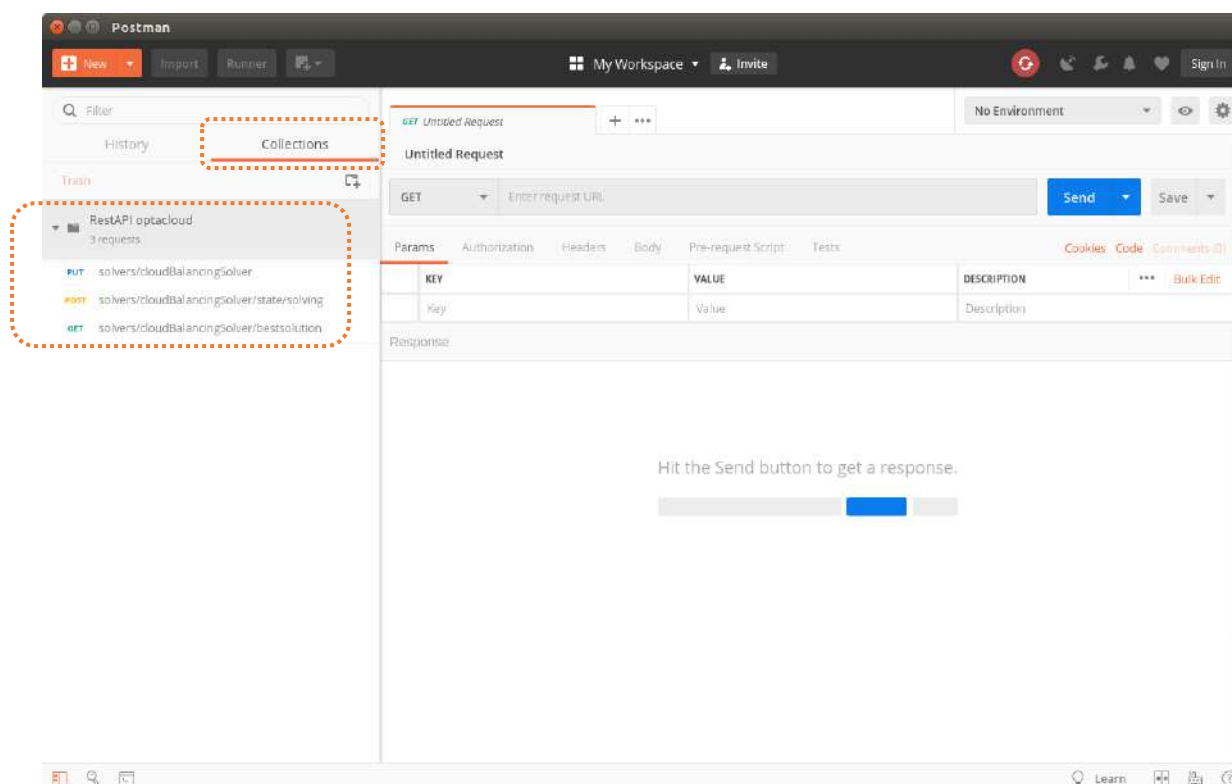


Import Pre-built API calls for OptaCloud project



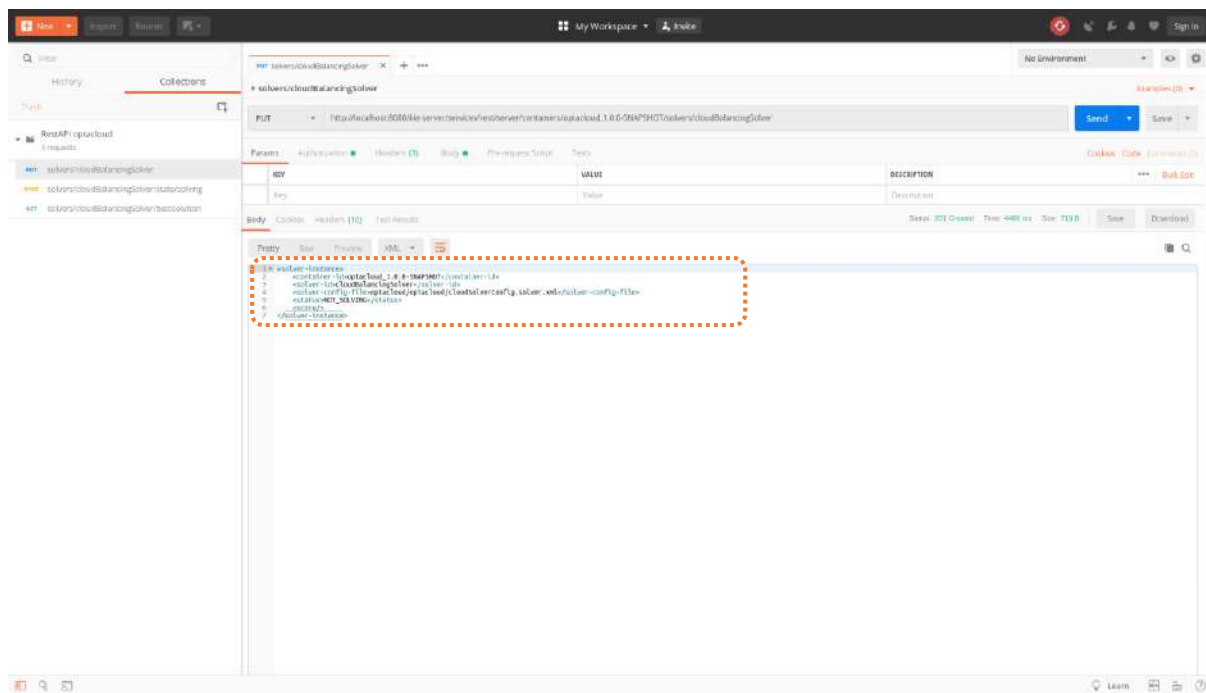
Select file: ***RestAPI optacloud v?.0.0.postman_collection.json***





Above: Imported **PUT/POST/GET** APIs





Above: REST-API PUT **Solver_Config** via Postman tool

```
< solver-instance>
  < solver-config-file>optacloud/optacloud/cloudSolverConfig.solver.xml</solver-config-file>
</ solver-instance>
```





```
<planning-problem class="optacloud.optacloud.CloudSolution" id="1">
  <id>0</id>
  <computerList id="2">
    <optacloud.optacloud.Computer id="3">
      <id>0</id>
      <cpuPower>24</cpuPower>
      <memory>96</memory>
      <networkBandwidth>16</networkBandwidth>
      <cost>4800</cost>
    </optacloud.optacloud.Computer>
    <optacloud.optacloud.Computer id="4">
      <id>1</id>
      <cpuPower>6</cpuPower>
      <memory>4</memory>
      <networkBandwidth>6</networkBandwidth>
      <cost>660</cost>
    </optacloud.optacloud.Computer>
  </computerList>
  <processList id="5">
    <optacloud.optacloud.Process id="6">
      <id>0</id>
      <requiredCpuPower>1</requiredCpuPower>
      <requiredMemory>1</requiredMemory>
      <requiredNetworkBandwidth>1</requiredNetworkBandwidth>
    </optacloud.optacloud.Process>
    <optacloud.optacloud.Process id="7">
      <id>1</id>
      <requiredCpuPower>3</requiredCpuPower>
      <requiredMemory>6</requiredMemory>
      <requiredNetworkBandwidth>1</requiredNetworkBandwidth>
    </optacloud.optacloud.Process>
    <optacloud.optacloud.Process id="8">
      <id>2</id>
      <requiredCpuPower>1</requiredCpuPower>
      <requiredMemory>1</requiredMemory>
      <requiredNetworkBandwidth>3</requiredNetworkBandwidth>
    </optacloud.optacloud.Process>
    <optacloud.optacloud.Process id="9">
```

```
<id>3</id>

<requiredCpuPower>1</requiredCpuPower>

<requiredMemory>2</requiredMemory>

<requiredNetworkBandwidth>11</requiredNetworkBandwidth>
</optacloud.optacloud.Process>
<optacloud.optacloud.Process id="10">
  <id>4</id>

  <requiredCpuPower>1</requiredCpuPower>

  <requiredMemory>1</requiredMemory>

  <requiredNetworkBandwidth>1</requiredNetworkBandwidth>
</optacloud.optacloud.Process>
<optacloud.optacloud.Process id="11">
  <id>5</id>

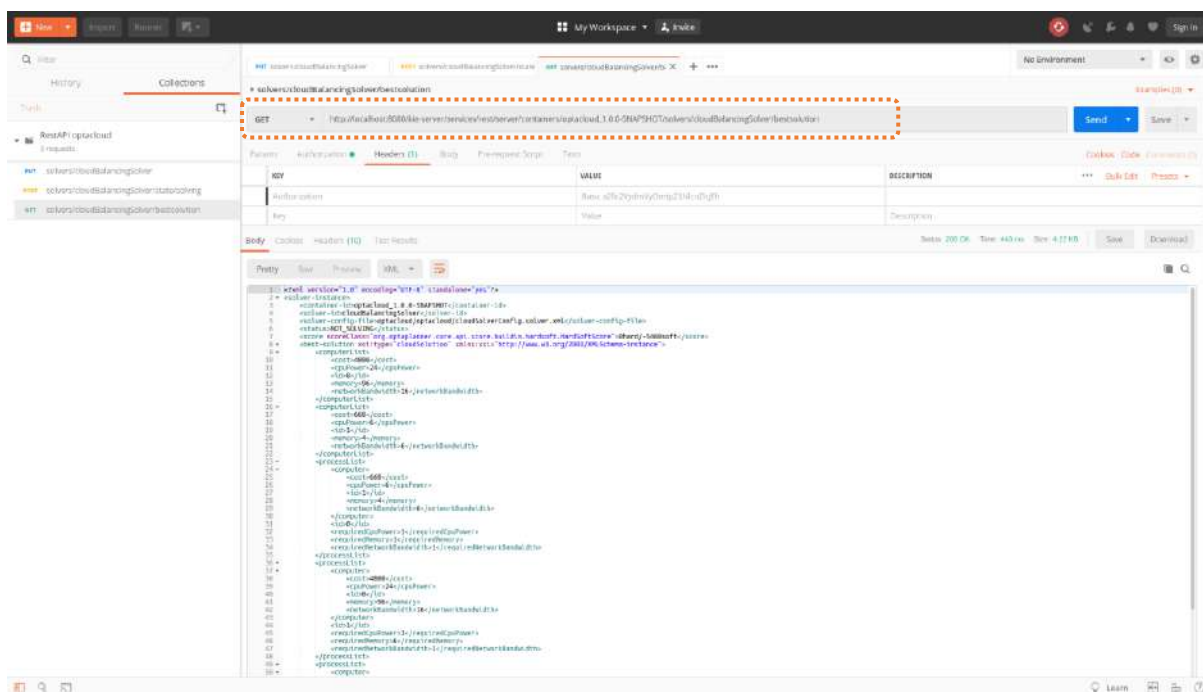
  <requiredCpuPower>1</requiredCpuPower>

  <requiredMemory>1</requiredMemory>

  <requiredNetworkBandwidth>5</requiredNetworkBandwidth>
</optacloud.optacloud.Process>
</processList>
</planning-problem>
```




GET – Obtain best solution (xml)



Above: REST-API GET **best solution** via Postman tool

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<solver-instance>
  <container-id>optacloud_1.0.0-SNAPSHOT</container-id>
  <solver-id>cloudBalancingSolver</solver-id>
  <solver-config-file>optacloud/optacloud/cloudSolverConfig.solver.xml</solver-config-file>
  <status>NOT_SOLVING</status>
  <score scoreClass="org.optaplanner.core.api.score.buildin.hardsoft.HardSoftScore">0hard/-
5460soft</score>
  <best-solution xsi:type="cloudSolution" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <computerList>
      <cost>4800</cost>
      <cpuPower>24</cpuPower>
      <id>0</id>
      <memory>96</memory>
      <networkBandwidth>16</networkBandwidth>
    </computerList>
    <computerList>
      <cost>660</cost>
      <cpuPower>6</cpuPower>
      <id>1</id>
      <memory>4</memory>
      <networkBandwidth>6</networkBandwidth>
    </computerList>
    <processList>
      <computer>
        <cost>660</cost>
        <cpuPower>6</cpuPower>
        <id>1</id>
        <memory>4</memory>
        <networkBandwidth>6</networkBandwidth>
      </computer>
      <id>0</id>
      <requiredCpuPower>1</requiredCpuPower>
      <requiredMemory>1</requiredMemory>
      <requiredNetworkBandwidth>1</requiredNetworkBandwidth>
    </processList>
    <processList>
      <computer>
        <cost>4800</cost>

```

```

        <cpuPower>24</cpuPower>

        <id>0</id>

        <memory>96</memory>

        <networkBandwidth>16</networkBandwidth>

    </computer>

    <id>1</id>

    <requiredCpuPower>3</requiredCpuPower>

    <requiredMemory>6</requiredMemory>

    <requiredNetworkBandwidth>1</requiredNetworkBandwidth>

</processList>
<processList>
    <computer>

        <cost>4800</cost>

        <cpuPower>24</cpuPower>

        <id>0</id>

        <memory>96</memory>

        <networkBandwidth>16</networkBandwidth>

    </computer>

    <id>2</id>

    <requiredCpuPower>1</requiredCpuPower>

    <requiredMemory>1</requiredMemory>

    <requiredNetworkBandwidth>3</requiredNetworkBandwidth>

</processList>
<processList>
    <computer>

        <cost>4800</cost>

        <cpuPower>24</cpuPower>

        <id>0</id>

        <memory>96</memory>

        <networkBandwidth>16</networkBandwidth>

    </computer>

    <id>3</id>

    <requiredCpuPower>1</requiredCpuPower>

    <requiredMemory>2</requiredMemory>

    <requiredNetworkBandwidth>11</requiredNetworkBandwidth>

</processList>
<processList>
    <computer>

        <cost>4800</cost>

```

```

        <cpuPower>24</cpuPower>

        <id>0</id>

        <memory>96</memory>

        <networkBandwidth>16</networkBandwidth>

    </computer>

    <id>4</id>

    <requiredCpuPower>1</requiredCpuPower>

    <requiredMemory>1</requiredMemory>

    <requiredNetworkBandwidth>1</requiredNetworkBandwidth>

</processList>
<processList>
    <computer>

        <cost>660</cost>

        <cpuPower>6</cpuPower>

        <id>1</id>

        <memory>4</memory>

        <networkBandwidth>6</networkBandwidth>

    </computer>

    <id>5</id>

    <requiredCpuPower>1</requiredCpuPower>

    <requiredMemory>1</requiredMemory>

    <requiredNetworkBandwidth>5</requiredNetworkBandwidth>

</processList>

<score>0hard/-5460soft</score>

<id>0</id>

</best-solution>

</solver-instance>

```

Reference

- https://docs.optaplanner.org/7.12.0.Final/optaplanner-wb-es-docs/html_single/
- <http://www.optaplanner.org/learn/useCases/cloudOptimization.html>
- <http://www.optaplanner.org/learn/slides/optaplanner-presentation/training.html#/4/26>
- https://docs.jboss.org/optaplanner/release/latestFinal/optaplanner-wb-es-docs/html_single/

1.2. Cloud Balance Solver Enhancement

{ Objective } Enhance the cloud balance solution to address additional business considerations/constraints, e.g. deep learning process with GPU requirements; Data centre physical locations; Network latency, etc.

Reuse KIE Workbench OptaCloud REST-API project. Enrich the Cloud Balancing domain model and add extra constraints such as:

- Some Process running deep learning neural network models can require graphical processing units GPU chips, so these processes should (or must) be assigned to computers with sufficient number of GPU chips.
- Each Process belongs to a Service. A computer might crash, so processes running the same service must be assigned to different computers.
- Each Computer is located in a Building. A building might burn down, so processes of the same services should (or must) be assigned to computers in different buildings.

Reference

https://docs.optaplanner.org/latest/optaplanner-docs/html_single/index.html#cloudBalancingBeyondThisTutorial

{ Reference Guide for GPU Enhancement }

1.2.1. Cloud Balance Solver Enhancement [Data Objects]

1) Add new **gpuPower** field for **Computer** data object; **Save**

The screenshot shows the KIE Workbench interface. On the left, the 'Project Explorer' shows the 'Computer' data object under 'Data Objects'. The main panel displays the 'Computer' data object configuration. A table lists the fields for the 'Computer' object:

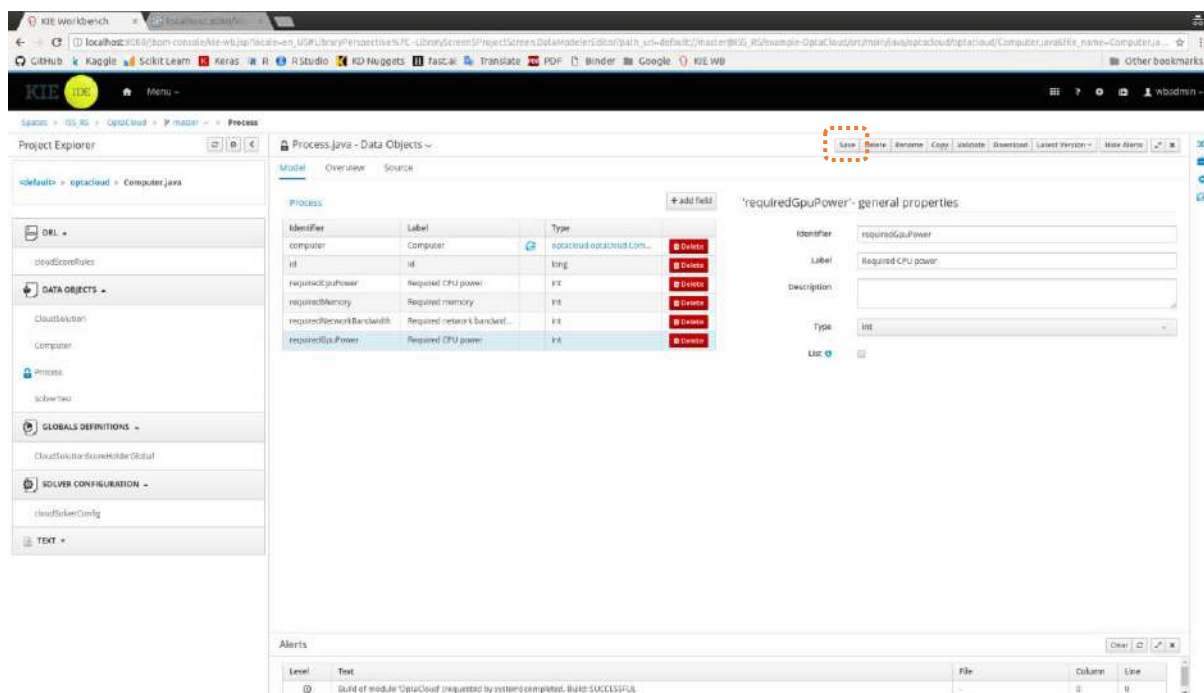
Identifier	Label	Type	Action
cost	Hosting cost	int	Delete
cpuPower	CPU power	int	Delete
id	id	long	Delete
memory	Memory	int	Delete
networkBandwidth	Network bandwidth	int	Delete
gpuPower	GPU Power	int	Delete

The 'gpuPower' field is highlighted in blue. To the right of the table, the 'gpuPower' general properties are shown:

- Identifier: gpuPower
- Label: GPU Power
- Description:
- Type: int
- Use:

The 'Save' button is highlighted with a red dashed box. At the bottom, an 'Alerts' panel shows a message: 'Build of module 'OptiCloud' (requested by subadmin) completed. Build: SUCCESSFUL.'

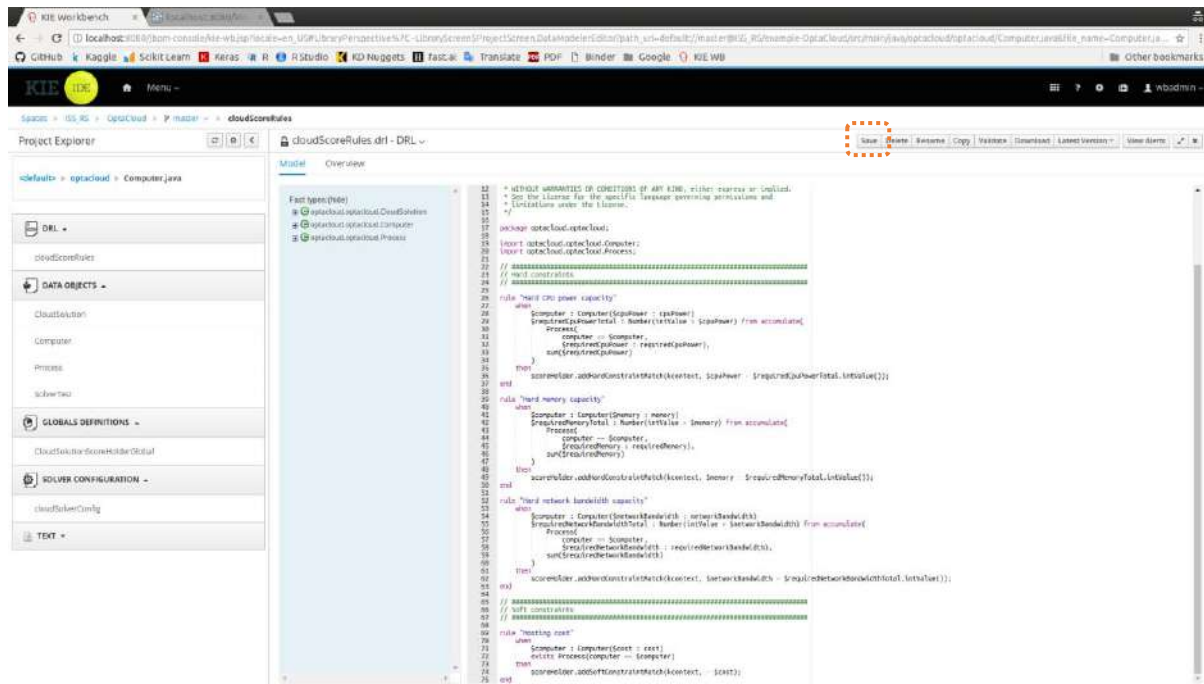
2) Add new **requiredGpuPower** field for **Process** data object; **Save**



1.2.2. Cloud Balance Solver Enhancement [DRL Rule]

1) Update DRL rule to include new constraint **GPU** into score calculation; **Save**

{ Quiz } Should the **GPU** be treated as **hard constraint** or **soft constraint**?



```

/*
 * Copyright 2017 Red Hat, Inc. and/or its affiliates.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package optacloud.optacloud;

import optacloud.optacloud.Computer;
import optacloud.optacloud.Process;

// #####
// Hard constraints
// #####

rule "Hard CPU power capacity"
    when
        $computer : Computer($cpuPower : cpuPower)
        $requiredCpuPowerTotal : Number(intValue > $cpuPower) from accumulate(
            Process(
                computer == $computer,
                $requiredCpuPower : requiredCpuPower),
            sum($requiredCpuPower)
        )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $cpuPower -
$requiredCpuPowerTotal.intValue());
    end

```

```
// -----
// 2019 March 04 - GU Zhan (Sam) - GPU constraints - START

rule "Hard GPU power capacity"
    when
        $computer : Computer($gpuPower : gpuPower)
        $requiredGpuPowerTotal : Number(intValue > $gpuPower) from accumulate(
            Process(
                computer == $computer,
                $requiredGpuPower : requiredGpuPower),
            sum($requiredGpuPower)
        )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $gpuPower -
$requiredGpuPowerTotal.intValue());
    end

// 2019 March 04 - GU Zhan (Sam) - GPU constraints - E N D
// -----

rule "Hard memory capacity"
    when
        $computer : Computer($memory : memory)
        $requiredMemoryTotal : Number(intValue > $memory) from accumulate(
            Process(
                computer == $computer,
                $requiredMemory : requiredMemory),
            sum($requiredMemory)
        )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $memory -
$requiredMemoryTotal.intValue());
    end

rule "Hard network bandwidth capacity"
    when
        $computer : Computer($networkBandwidth : networkBandwidth)
        $requiredNetworkBandwidthTotal : Number(intValue > $networkBandwidth) from accumulate(
            Process(
                computer == $computer,
```



```

        $requiredNetworkBandwidth : requiredNetworkBandwidth),
        sum($requiredNetworkBandwidth)
    )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $networkBandwidth -
$requiredNetworkBandwidthTotal.intValue());
    end

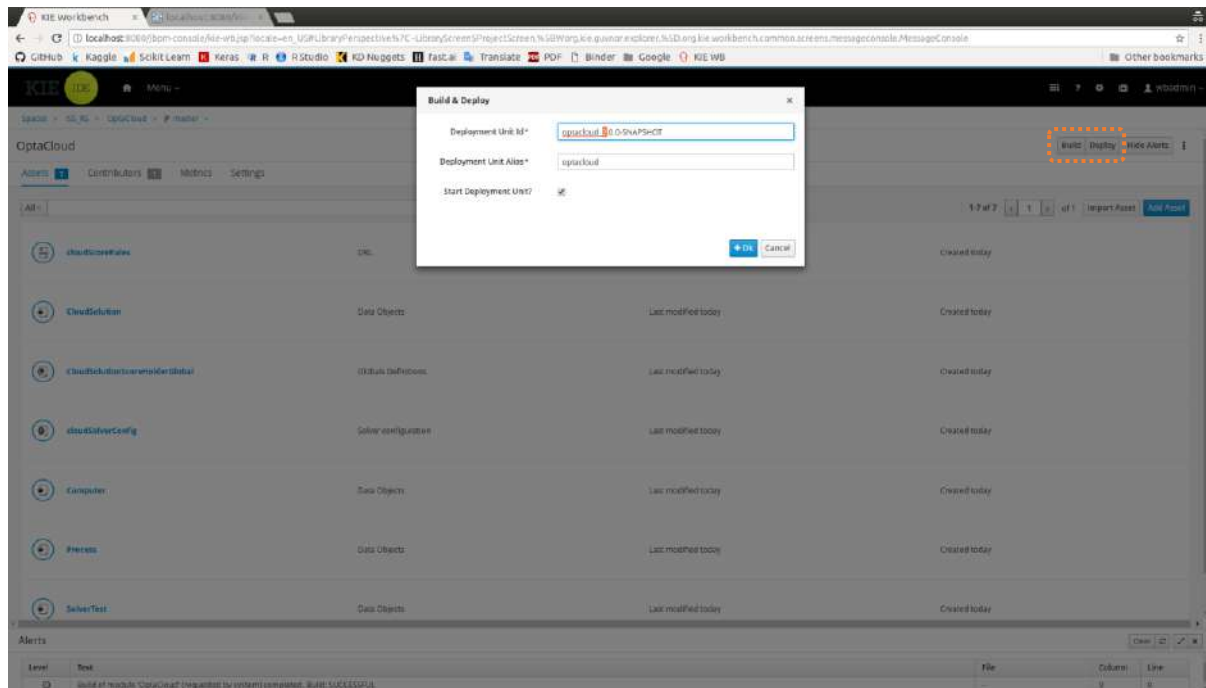
// #####
// Soft constraints
// #####

rule "Hosting cost"
    when
        $computer : Computer($cost : cost)
        exists Process(computer == $computer)
    then
        scoreHolder.addSoftConstraintMatch(kcontext, - $cost);
    end
end

```

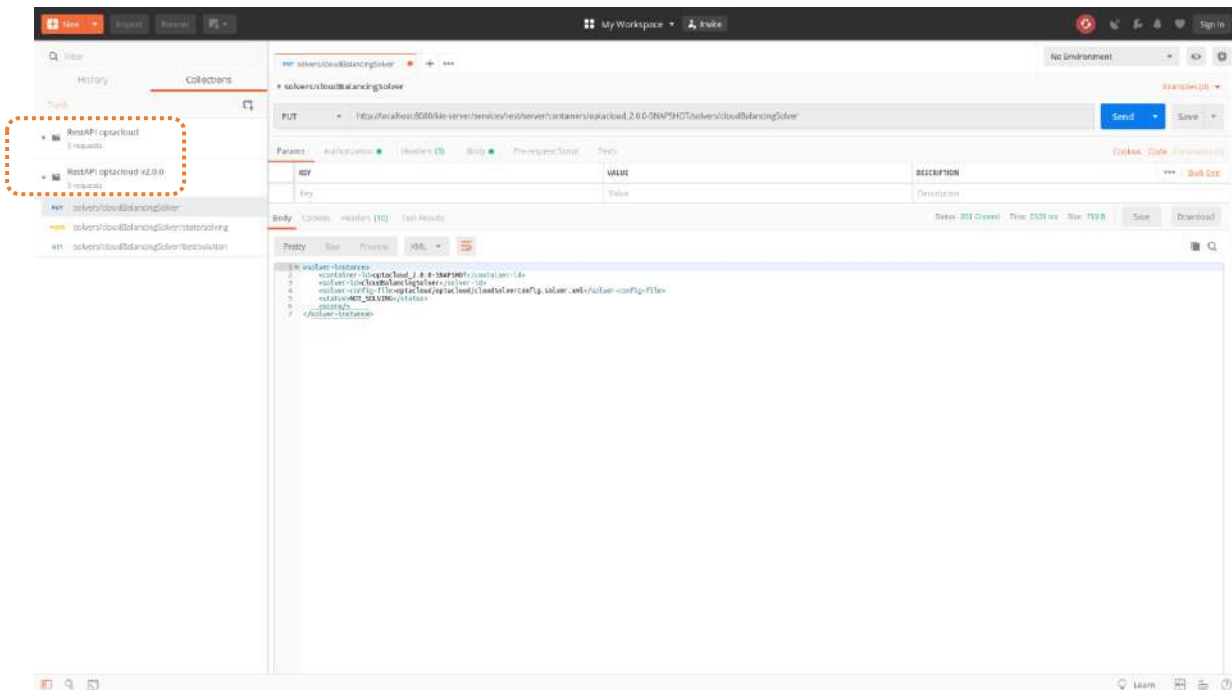
1.2.3. Cloud Balance Solver Enhancement [Deploy] v2.0.0

1) [Deploy] Deploy OptaCloud solver to server



Deploy to KIE Server **optacloud_2.0.0-SNAPSHOT**

1) [API: PUT] Initialize a new solver instance;



2) [API: POST] Provide a new business problem/context (with GPU requirements);

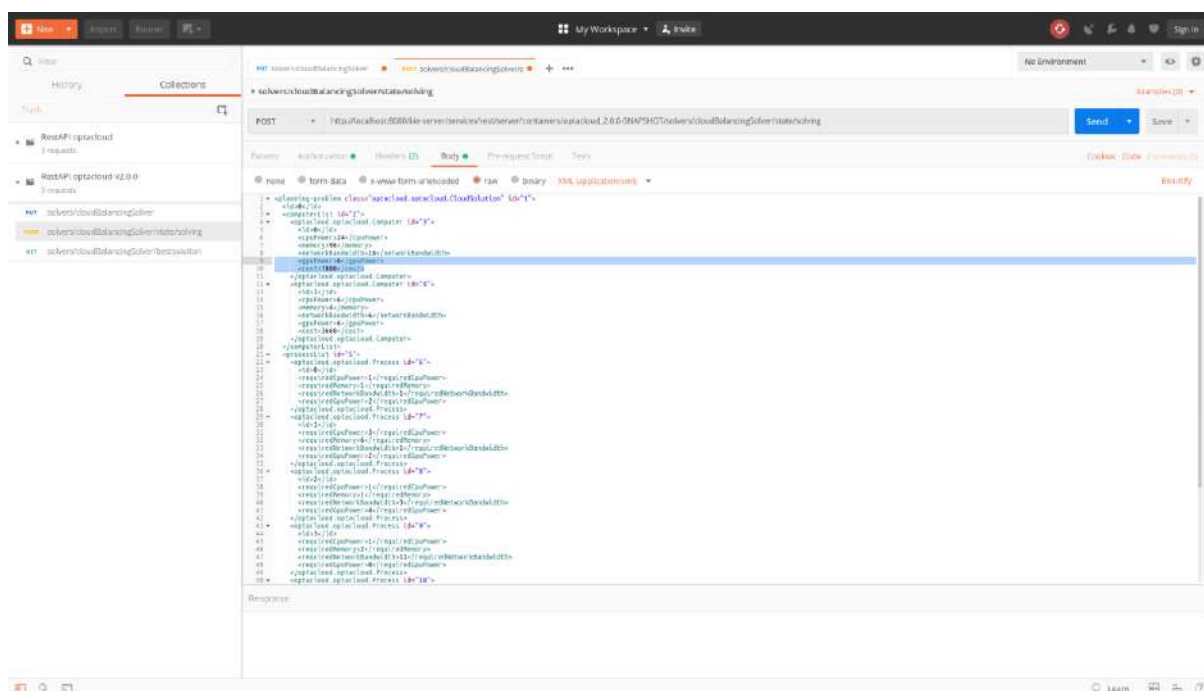
Update XML in API Body to introduce a new cloud balance problem:

Two computers with GPUs:

- Computer 0 has 6 GPU cards.
- Computer 1 has 6 GPU cards.

Six processes to assign to computer:

- Process 0 requires 2 GPU cards.
- Process 1 requires 2 GPU cards.
- Process 2 requires 4 GPU cards.
- Process 3 requires 0 GPU cards.
- Process 4 requires 0 GPU cards.
- Process 5 requires 0 GPU cards.



```
<planning-problem class="optacloud.optacloud.CloudSolution" id="1">
  <id>0</id>
  <computerList id="2">
    <optacloud.optacloud.Computer id="3">
      <id>0</id>
      <cpuPower>24</cpuPower>
      <memory>96</memory>
      <networkBandwidth>16</networkBandwidth>
      <gpuPower>6</gpuPower>
      <cost>7800</cost>
    </optacloud.optacloud.Computer>
    <optacloud.optacloud.Computer id="4">
      <id>1</id>
      <cpuPower>6</cpuPower>
      <memory>4</memory>
      <networkBandwidth>6</networkBandwidth>
      <gpuPower>6</gpuPower>
      <cost>3660</cost>
    </optacloud.optacloud.Computer>
  </computerList>
  <processList id="5">
    <optacloud.optacloud.Process id="6">
      <id>0</id>
      <requiredCpuPower>1</requiredCpuPower>
      <requiredMemory>1</requiredMemory>
      <requiredNetworkBandwidth>1</requiredNetworkBandwidth>
      <requiredGpuPower>2</requiredGpuPower>
    </optacloud.optacloud.Process>
    <optacloud.optacloud.Process id="7">
      <id>1</id>
      <requiredCpuPower>3</requiredCpuPower>
      <requiredMemory>6</requiredMemory>
      <requiredNetworkBandwidth>1</requiredNetworkBandwidth>
      <requiredGpuPower>2</requiredGpuPower>
    </optacloud.optacloud.Process>
    <optacloud.optacloud.Process id="8">
      <id>2</id>
      <requiredCpuPower>1</requiredCpuPower>
```



```
<requiredMemory>1</requiredMemory>

<requiredNetworkBandwidth>3</requiredNetworkBandwidth>

<requiredGpuPower>4</requiredGpuPower>

</optaccloud.optaccloud.Process>

<optaccloud.optaccloud.Process id="9">

  <id>3</id>

  <requiredCpuPower>1</requiredCpuPower>

  <requiredMemory>2</requiredMemory>

  <requiredNetworkBandwidth>11</requiredNetworkBandwidth>

  <requiredGpuPower>0</requiredGpuPower>

</optaccloud.optaccloud.Process>

<optaccloud.optaccloud.Process id="10">

  <id>4</id>

  <requiredCpuPower>1</requiredCpuPower>

  <requiredMemory>1</requiredMemory>

  <requiredNetworkBandwidth>1</requiredNetworkBandwidth>

  <requiredGpuPower>0</requiredGpuPower>

</optaccloud.optaccloud.Process>

<optaccloud.optaccloud.Process id="11">

  <id>5</id>

  <requiredCpuPower>1</requiredCpuPower>

  <requiredMemory>1</requiredMemory>

  <requiredNetworkBandwidth>5</requiredNetworkBandwidth>

  <requiredGpuPower>0</requiredGpuPower>

</optaccloud.optaccloud.Process>

</processList>

</planning-problem>
```

[illegible][illegible]

{ Quiz } What controls solver to stop?

[Optional] { Challenge } Update the last three process to require 2 GPUs respectively; Run the solver to get solution; Observe the final score: Are all constraints (business resource requirements) satisfied?

Update XML in API Body to introduce a new cloud balance problem:

Two computers with GPUs:

- Computer 0 has 6 GPU cards.
- Computer 1 has 6 GPU cards.

Six processes to assign to computer:

- Process 0 requires 2 GPU cards.
- Process 1 requires 2 GPU cards.
- Process 2 requires 4 GPU cards.
- Process 3 requires 2 GPU cards.
- Process 4 requires 2 GPU cards.
- Process 5 requires 2 GPU cards.

😊 Congratulations!

You have completed today's challenging workshop!

The End of Workshop Project Guide