

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

**GRADUATE CERTIFICATE
INTELLIGENT REASONING SYSTEMS**

Workshop Project Guide

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



This is a blank page.

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].....	8
1.1.2. Cloud Balance Solver [KIE Workbench] v1.0.0	20
1.2. Cloud Balance Solver Enhancement.....	45
1.2.1. Cloud Balance Solver Enhancement [Data Objects]	46
1.2.2. Cloud Balance Solver Enhancement [DRL Rule].....	48
1.2.3. Cloud Balance Solver Enhancement [Deploy] v2.0.0.....	52
1.2.4. Cloud Balance Solver Enhancement [API].....	53
1.3. Hybrid Reasoning System [MTech Thru-Train Group Project]	59

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).

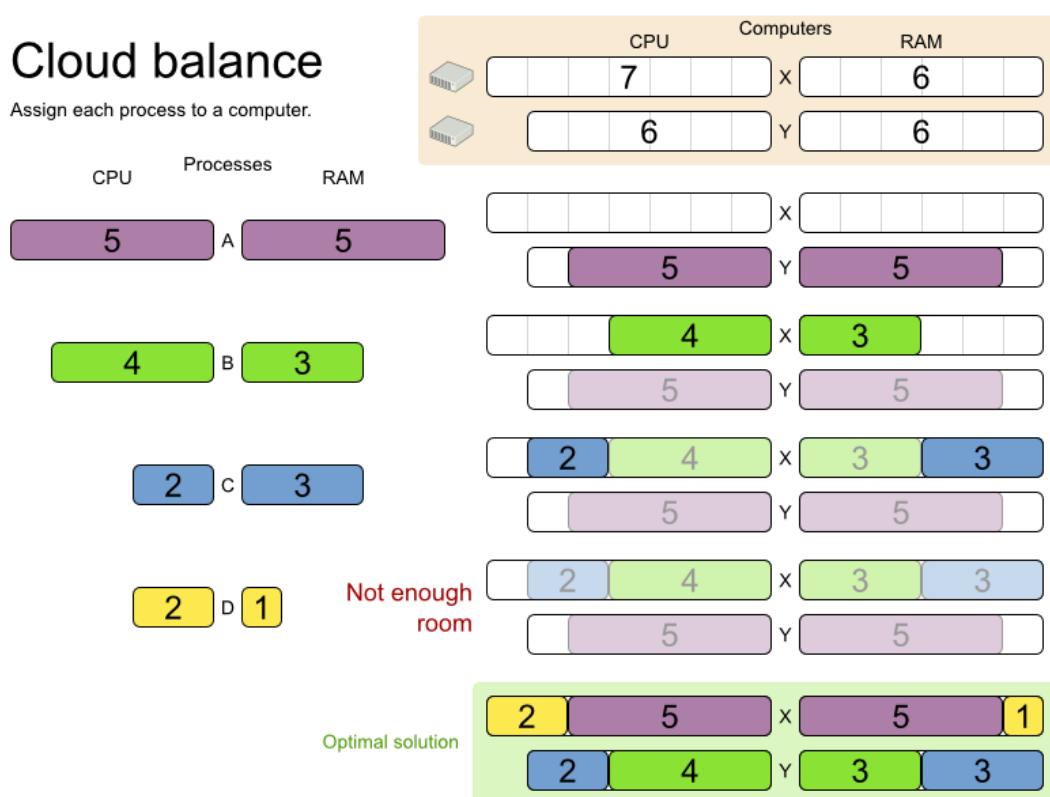
Decompose the end to end system solution, at system modelling level, to address the cloud balancing business resource optimization problem.

KIE OptaPlanner Deep Dive – Cloud Computer Balancing

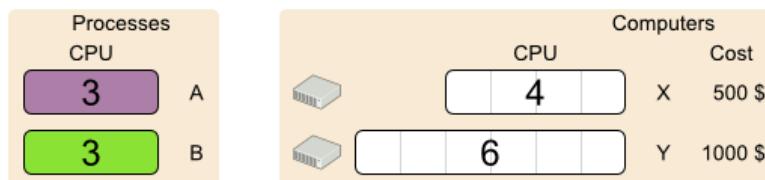
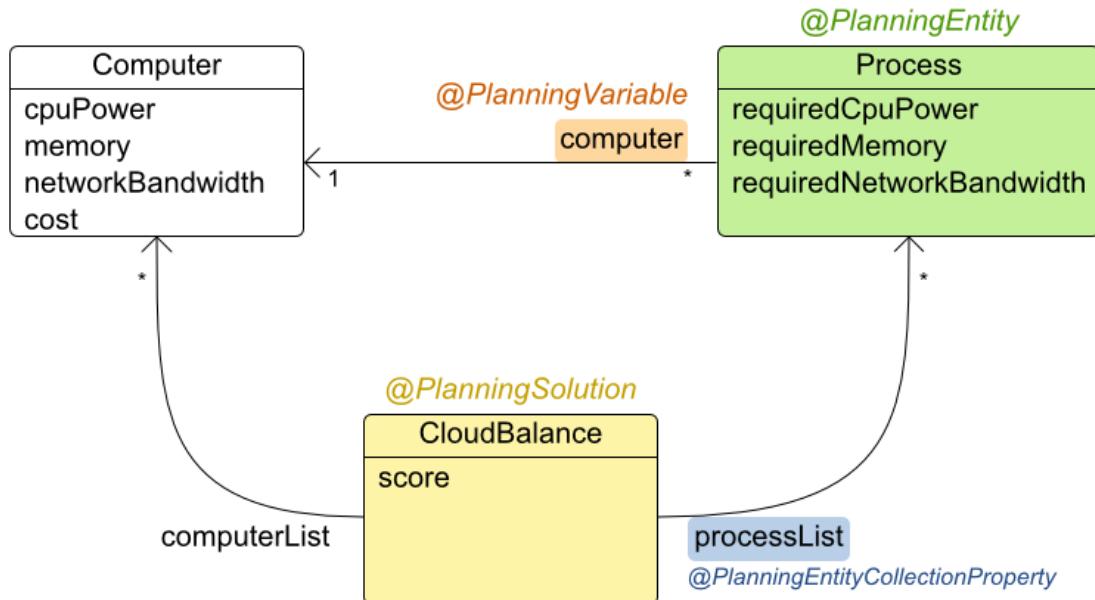
- **Business Scenario / Problem Description**
- **A cloud service provider owns a number of cloud computers and needs to run a number of customers' processes on those computers. Assign each process to a computer.**

Cloud balance

Assign each process to a computer.

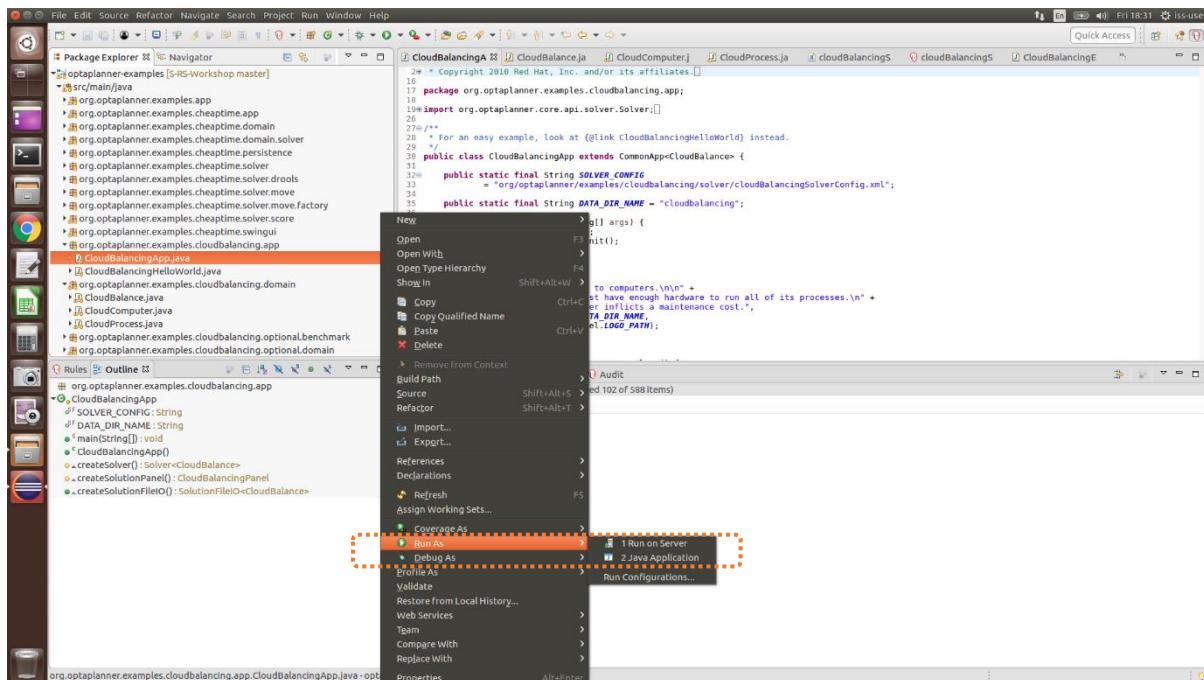


Cloud balance class diagram

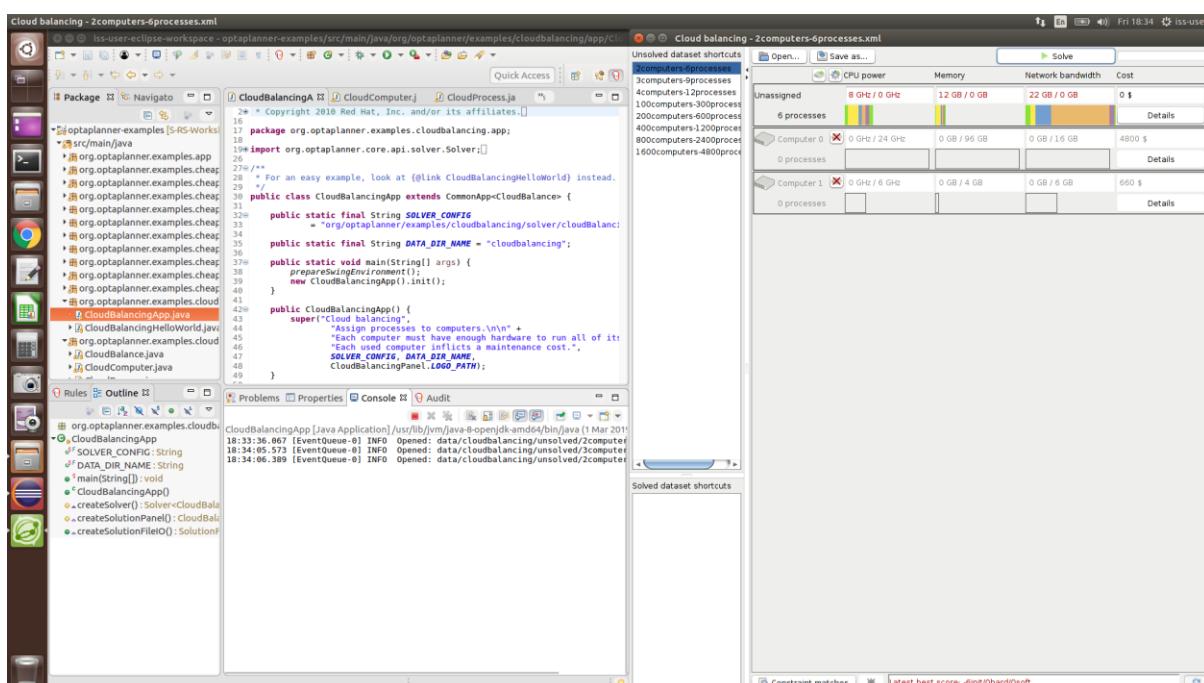


1.1.1. Cloud Balance Solver [Java IDE]

- 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;



The screenshot shows the Eclipse IDE interface with several open windows:

- Cloud balancing - 2computers-6processes.xml**: An XML configuration file for the cloud balancing problem.
- Resource View**: Shows the project structure under "optaplanner-example [S-RW-WorkSpace]".
- Package Explorer**: Shows the Java source code for `CloudBalancingApp.java` and `CloudComputer.java`.
- Quick Access**: A search bar at the top.
- Solver Status**: A window showing the status of the solver: "Solved dataset shortcuts" and "Unsolved dataset shortcuts". It lists various configurations (e.g., 3computers-2processes, 4computers-12processes) and their corresponding scores.
- Cloud balancing - 2computers-6processes**: A detailed view of the solved dataset. It shows two computers: Computer 0 (4 processes) and Computer 1 (2 processes). Each computer has its CPU power, memory, network bandwidth, and cost listed. The table also includes a "Details" column for each row.
- Problems**, **Properties**, **Console**, and **Audit**: Standard Eclipse toolbars.
- CloudBalancingApp [Java Application]**: The Java application's output window showing log messages from the solver.

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

CloudBalance.java

CloudComputer.java

The screenshot shows the Eclipse IDE interface with the CloudBalancing project selected in the Package Explorer. The CloudComputer.java file is open in the central editor window. The code implements the Labeled interface with fields for CPU power, memory, and network bandwidth, and methods for setting and getting these values, as well as calculating cost and multiplying values.

```
Copyright 2010 Red Hat, Inc. and/or its affiliates.  
16  
17 package org.optaplanner.examples.cloudbalancing.domain;  
18  
19import com.thoughtworks.xstream.annotations.XStreamAlias;  
20  
21@XStreamAlias("CloudComputer")  
22public class CloudComputer extends AbstractPersistable implements Labeled {  
23  
24    private int cpuPower; // in gigahertz  
25    private int memory; // in gigabyte RAM  
26    private int networkBandwidth; // in gigabyte per hour  
27    private int cost; // in euro per month  
28  
29    public CloudComputer() {  
30        super();  
31    }  
32  
33    public CloudComputer(long id, int cpuPower, int memory, int networkBandwidth, int cost) {  
34        super(id);  
35        this.cpuPower = cpuPower;  
36        this.memory = memory;  
37        this.networkBandwidth = networkBandwidth;  
38        this.cost = cost;  
39    }  
40  
41    public int getcpuPower() {  
42        return cpuPower;  
43    }  
44  
45    public void setcpuPower(int cpuPower) {  
46        this.cpuPower = cpuPower;  
47    }  
48  
49    public int getmemory() {  
50        return memory;  
51    }  
52  
53    public void setMemory(int memory) {  
54        this.memory = memory;  
55    }  
56  
57    public int getnetworkBandwidth() {  
58        return networkBandwidth;  
59    }  
60  
61    public void setNetworkBandwidth(int networkBandwidth) {  
62        this.networkBandwidth = networkBandwidth;  
63    }  
64  
65    public int getcost() {  
66        return cost;  
67    }  
68  
69    public void setCost(int cost) {  
70        this.cost = cost;  
71    }  
72  
73    // Complex methods  
74  
75    public int getMultiplicand() {  
76        return cpuPower * memory * networkBandwidth;  
77    }  
78  
79    public int getLabel() {  
80        return 1;  
81    }  
82}
```

CloudProcess.java

```

1   /* Copyright 2010 Red Hat, Inc. and/or its affiliates. */
2   package org.optaplanner.examples.cloudbalancing.domain;
3   import com.thoughtworks.xstream.annotations.XStreamAlias;
4   import org.optaplanner.examples.cloudbalancing.domain.planningentity.difficultyComparatorClass = CloudProcessDifficultyComparator.class;
5   import org.optaplanner.examples.cloudbalancing.domain.planningentity.difficultyComparatorName = "CloudProcess";
6   public class CloudProcess extends AbstractPersistable {
7       private int requiredCpuPower; // in gigahertz
8       private int requiredMemory; // in gigabyte RAM
9       private int requiredNetworkBandwidth; // in gigabyte per hour
10      // Planning variables: changes during planning, between score calculations.
11      private CloudComputer computer;
12      public CloudProcess() {
13      }
14      public CloudProcess(long id, int requiredCpuPower, int requiredMemory, int requiredNetworkBandwidth) {
15          super(id);
16          this.requiredCpuPower = requiredCpuPower;
17          this.requiredMemory = requiredMemory;
18          this.requiredNetworkBandwidth = requiredNetworkBandwidth;
19      }
20      public int getRequiredCpuPower() {
21          return requiredCpuPower;
22      }
23      public void setRequiredCpuPower(int requiredCpuPower) {
24          this.requiredCpuPower = requiredCpuPower;
25      }
26      public int getRequiredMemory() {
27          return requiredMemory;
28      }
29      public void setRequiredMemory(int requiredMemory) {
30          this.requiredMemory = requiredMemory;
31      }
32      public int getRequiredNetworkBandwidth() {
33          return requiredNetworkBandwidth;
34      }
35      public void setRequiredNetworkBandwidth(int requiredNetworkBandwidth) {
36          this.requiredNetworkBandwidth = requiredNetworkBandwidth;
37      }
38      @PlanningVariable(valueRangeProviderRefs = {"computerRange"}, strengthComparatorClass = CloudComputerStrengthComparator.class)
39      public CloudComputer getComputer() {
40          return computer;
41      }
42      public void setComputer(CloudComputer computer) {
43          this.computer = computer;
44      }
45      // *****
46      // Complex methods
47      // *****
48  }
```

Physical files location:

Name	Size	Type	Modified	Accessed	Owner	Group	Permissions	MIME Type	Location
Home									/media/
Desktop									/media/
Documents									/media/
Downloads									/media/
Music									/media/
Pictures									/media/
Videos									/media/
Trash									/media/
sf_vm_shared_folder									
Network									
Computer									
Connect to Server									
sf-RS-Workshop	5 items	Folder	17:08	17:08	root	vboxsf	drwxrwx-	inode/directory	/media/
sf-RS-Workshop1	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx-	inode/directory	/media/
reference	3 items	Folder	17:09	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	17:09	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/
cheaptim	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/
optional	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx-	inode/directory	/media/
CloudBalance.java	2.8 kB	Text	17:09	17:09	root	vboxsf	-rwxrwx-	text/x-java	/media/
CloudComputer.java	2.4 kB	Text	17:09	17:09	root	vboxsf	-rwxrwx-	text/x-java	/media/
CloudProcess.java	3.2 kB	Text	17:09	17:09	root	vboxsf	-rwxrwx-	text/x-java	/media/
CloudProcess	6 items	Folder	17:09	17:09	root	vboxsf	drwxrwx-	inode/directory	/media/
optional	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx-	inode/directory	/media/
persistence	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx-	inode/directory	/media/
swingul	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx-	inode/directory	/media/
coachshuttlegathering	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/
flightcrewscheduling	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/
machinereassignment	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

```
Copyright 2010 Red Hat, Inc. and/or its affiliates.  
16  
17 package org.optaplanner.examples.cloudbalancing.app;  
18  
19 import org.optaplanner.core.api.solver.Solver;  
20  
21 /*  
22 * For an easy example, look at {@link CloudBalancingHelloWorld} instead.  
23 */  
24  
25 public class CloudBalancingApp extends CommonAppCloudBalance {  
26  
27     public static final String SOLVER_CONFIG =  
28         "org/optaplanner/examples/cloudbalancing/solver/cloudBalancingSolverConfig.xml";  
29  
30     public static final String DATA_DIR_NAME = "cloudbalancing";  
31  
32     public static void main(String[] args) {  
33         prepareswingenvironment();  
34         new CloudBalancingApp().init();  
35     }  
36  
37     public void init() {  
38         new CloudBalancingPanel().  
39             setLayout(new GridLayout(1, 1));  
40         add(CloudBalancingPanel.class.getName());  
41     }  
42     public CloudBalancingApp() {  
43         super("Cloud Balancing");  
44         assignProcessesToComputers();  
45         "Each computer must have enough hardware to run all of its processes.\n" +  
46         "Each user needs to pay a maintenance cost.",  
47         SOLVER_CONFIG, DATA_DIR_NAME,  
48         CloudBalancingPanel.LOGO_PATH);  
49     }  
50  
51     @Override  
52     protected Solver<CloudBalance> createSolver() {  
53         SolverFactory<CloudBalance> solverFactory = SolverFactory.createFromXmlResource(SOLVER_CONFIG);  
54         return solverFactory.buildSolver();  
55     }  
56  
57     @Override  
58     protected CloudBalancingPanel createSolutionPanel() {  
59         return new CloudBalancingPanel();  
60     }  
61  
62     @Override  
63     public SolutionFileIO<CloudBalance> createSolutionFileIO() {  
64         return new XStreamSolutionFileIO<(CloudBalance.class);  
65     }  
66 }  
67  
68
```

```
<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

Score configuration

<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingEasyScoreCalculator</easyScoreCalculatorClass>
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBaseScoreCalculator</easyScoreCalculatorClass>
<incrementalScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingIncrementalScoreCalculator</incrementalScoreCalculatorClass>
<scoreDrl>
<!-- score.drl -->
ONLY_DOWN
<assertionScoreDirectorFactory>
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBaseScoreCalculator</easyScoreCalculatorClass>
</assertionScoreDirectorFactory>
Optimization algorithms configuration

Power tweaked optimization algorithms configuration
<constructionHeuristic>
<constructionHeuristicType>FIRST_FIT_DECREASING</constructionHeuristicType>
</constructionHeuristic>
<localSearch>
<changeMoveSelector>
<changeMoveSelector/>
<swapMoveSelector/>
<pillarChangeMoveSelector/>
<pillarSwapMoveSelector/>
<unionMoveSelector/>
<acceptor>
<entityTabuSize>7</entityTabuSize>
<rejector>
<forager>
<acceptedCountLimit>1000</acceptedCountLimit>
</forager>
</localSearch>
Alternative power tweaked optimization algorithms configuration
<partitionedSearch>
<solutionPartitionerClass>org.optaplanner.examples.cloudbalancing.optional.partitioner.CloudBalancePartitioner</solutionPartitionerClass>
<solutionPartitionerCustomProperties>
```

Physical file's location:

sf_vm_shared_folder RS									
	Name	Size	Type	Modified	Accessed	Owner	Group	Permissions	MIME Type
Home	S-RS-Workshop	5 items	Folder	17:08	17:08	root	vboxsf	drwxrwx--	inode/directory
	S-RS-Workshop1	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	reference	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	optaplanner-examples	4 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	data	22 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	src	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	main	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	java	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	resources	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	org	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	optaplanner	24 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	examples	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	app	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	cheaptiming	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	cloudbalancing	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	optional	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	solver	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	cloudBalancingScoreRules.drl	3.0 kB	Text	17:09	17:09	root	vboxsf	-rwxrwx-	text/x-csrc
	cloudBalancingSolverConfig.xml	2.7 kB	Markup	17:09	17:09	root	vboxsf	-rwxrwx-	application/xml
	swingui	14 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	coachshuttlegathering	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	common	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	conferencescheduling	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	curriculumcourse	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	dinnerparty	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	examination	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	flightcrewscheduling	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	investment	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	machinereassignment	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	meetingscheduling	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	nqueens	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
	"cloudBalancingSolverConfig.xml" selected (2.7 kB)								

```

Open ▾ Save
<?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>
    <packageIncludes>org.optaplanner.examples.cloudbalancing</packageIncludes>
  </scanAnnotatedClasses>

  <!-- Score configuration -->
  <scoreDirectorFactory>
    <calculator>
      <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>
      <initialScoreDrl>org/optaplanner/examples/cloudbalancing/solver/initializingScoreRules.drl</initializingScoreDrl>
      <!--<assertionsScoreDirectorFactory-->
      <!--<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBasedEasyScoreCalculator</easyScoreCalculatorClass-->
    </calculator>
  </scoreDirectorFactory>

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

  <!-- Power tweaked optimization algorithms configuration -->
  <!--<constructionHeuristic-->
  <!--<constructionHeuristicType>FIRST_FIT_DECREASING</constructionHeuristicType-->
  <!--<constructionHeuristic-->
  <!--<unionMoveSelector-->
  <!--<changeMoveSelector-->
  <!--<swapMoveSelector-->
  <!--<partialChangeMoveSelector-->
  <!--<allMoveSelector-->
  <!--<unionMoveSelector-->
  <!--<acceptor-->
  <!--<entityTabuSize>7</entityTabuSize-->
  <!--<acceptor-->
  <!--<forager-->
  <!--<acceptedCount>1000</acceptedCount-->
  <!--<forager-->
  <!--<localSearch-->

  <!-- Alternative power tweaked optimization algorithms configuration -->
  <!--<partitionedSearch-->
  <!--<solutionPartitionerClass>org.optaplanner.examples.cloudbalancing.optional.partitioning.CloudBalancePartitioner</solutionPartitionerClass-->
  <!--<solutionPartitionerCustomProperties-->
  <!--<partitioner>partCount=1000</partitioner-->
  <!--<nInlimumProcessListSize>300</nInlimumProcessListSize-->
  <!--<solutionPartitionerCustomProperties-->
  <!--<partitionedSearch-->
</solver>

```

XML Tab Width: 8 Ln 1, Col 1 INS

```

<?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.CloudBalancin
gEasyScoreCalculator</easyScoreCalculatorClass>-->
        <!--
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancin
gMapBasedEasyScoreCalculator</easyScoreCalculatorClass>-->
        <!--
<incrementalScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudB
alancingIncrementalScoreCalculator</incrementalScoreCalculatorClass>-->

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

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

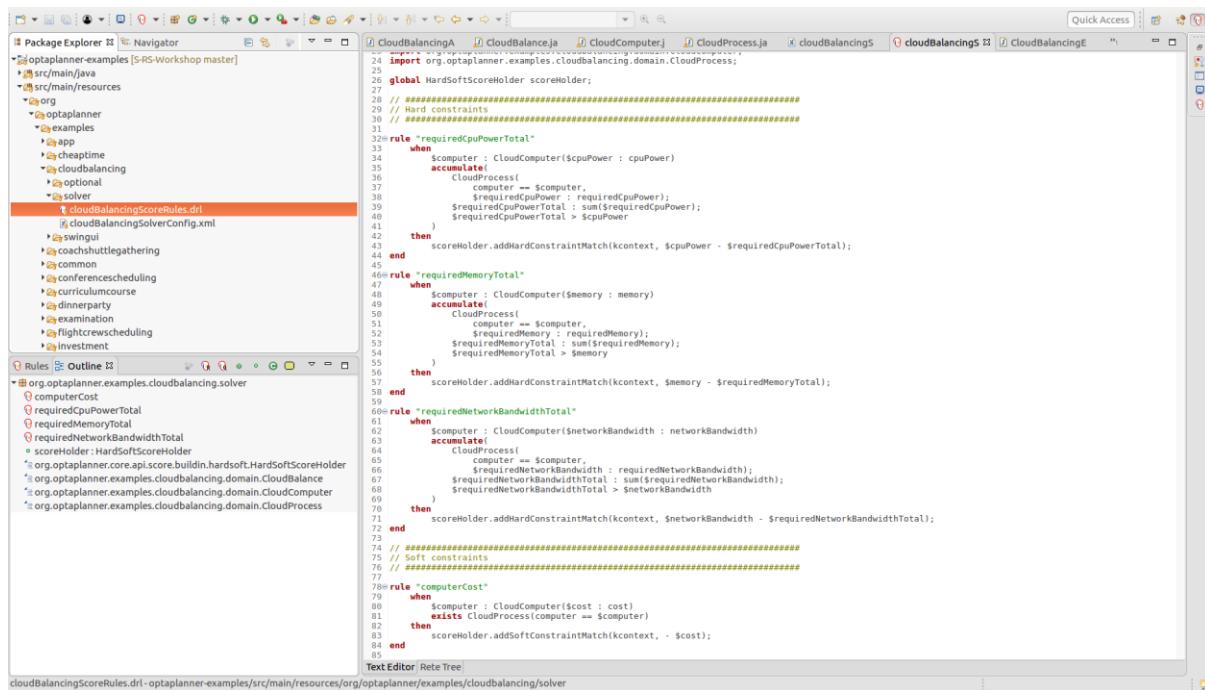
    <!-- Power tweaked optimization algorithms configuration -->
    <!--<constructionHeuristic>-->
        <!--<constructionHeuristicType>FIRST_FIT_DECREASING</constructionHeuristicType>-->
    <!--</constructionHeuristic>-->
    <!--<localSearch>-->
        <!--<unionMoveSelector>-->
            <!--<changeMoveSelector/>-->
            <!--<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.CloudBa
lancePartitioner</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



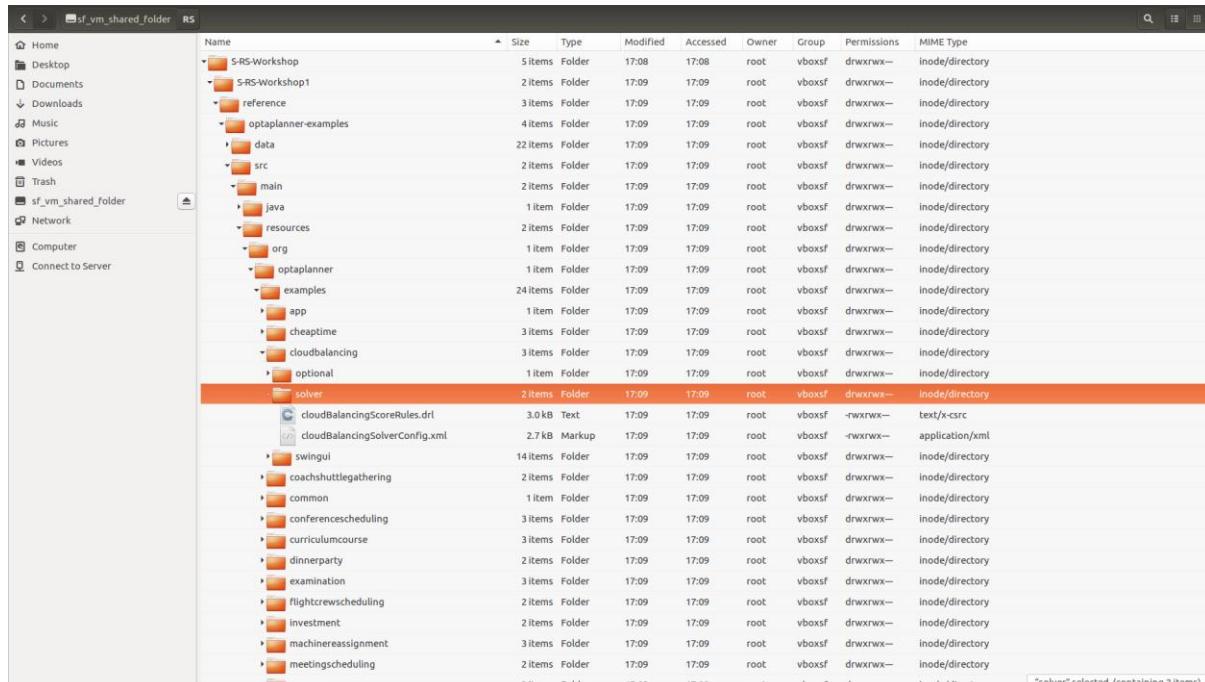
```

package org.optaplanner.examples.cloudbalancing.solver;
import org.optaplanner.examples.cloudbalancing.domain.CloudProcess;
import org.optaplanner.examples.cloudbalancing.domain.CloudComputer;
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 > $cpuPower)
then
    scoreHolder.addHardConstraintMatchInContext($cpuPower - $requiredCpuPowerTotal);
end
rule "requiredMemoryTotal"
when
    $computer : CloudComputer($memory : memory)
    accumulate(
        CloudProcess(
            computer == $computer,
            $requiredMemory : requiredMemory);
        $requiredMemoryTotal : sum($requiredMemory);
        $requiredMemoryTotal > $memory)
then
    scoreHolder.addHardConstraintMatchInContext($memory - $requiredMemoryTotal);
end
rule "requiredNetworkBandwidthTotal"
when
    $computer : CloudComputer($networkBandwidth : networkBandwidth)
    accumulate(
        CloudProcess(
            computer == $computer,
            $requiredNetworkBandwidth : requiredNetworkBandwidth);
        $requiredNetworkBandwidthTotal : sum($requiredNetworkBandwidth);
        $requiredNetworkBandwidthTotal > $networkBandwidth)
then
    scoreHolder.addHardConstraintMatchInContext($networkBandwidth - $requiredNetworkBandwidthTotal);
end
// #####
// Soft constraints
// #####
rule "computerCost"
when
    $computer : CloudComputer($cost : cost)
    exists CloudProcess(computer == $computer)
then
    scoreHolder.addSoftConstraintMatchInContext(-$cost);
end

```

Physical file's location:



Name	Size	Type	Modified	Accessed	Owner	Group	Permissions	MIME Type
sf_vm_shared_folder	5 items	Folder	17:08	17:08	root	vboxsf	drwxrwx--	inode/directory
Desktop								
Documents								
Downloads								
Music								
Pictures								
Videos								
Trash								
solver	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
cloudBalancingScoreRules.drl	3.0kB	Text	17:09	17:09	root	vboxsf	-rwxrwx--	text/x-csrc
cloudBalancingSolverConfig.xml	2.7 kB	Markup	17:09	17:09	root	vboxsf	-rwxrwx--	application/xml
swingui	14 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
coachshuttlegathering	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
common	1 item	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
conferencescheduling	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
curriculumcourse	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
dinnerparty	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
examination	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
flightcrewscheduling	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
investment	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
machinereassignment	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
meetingscheduling	2 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory
nqueens	3 items	Folder	17:09	17:09	root	vboxsf	drwxrwx--	inode/directory

```

/*
 * 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 #####
// 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

```

19:32:46.613 [main] DEBUG LS step (132376), time spent (119892), score (0hard/-492418soft), accepted/selected move count (1/1), picked move (CloudProcess-263 {CloudComputer-124}) <-> CloudProcess-513 {CloudComputer-109}
19:32:46.613 [main] DEBUG LS step (132377), time spent (119892), score (0hard/-491620soft), accepted/selected move count (1/2), picked move (CloudProcess-513 {CloudComputer-109}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.614 [main] DEBUG LS step (132378), time spent (119893), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/3), picked move (CloudProcess-421 {CloudComputer-125}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.614 [main] DEBUG LS step (132379), time spent (119893), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/3), picked move (CloudProcess-186 {CloudComputer-126}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.614 [main] DEBUG LS step (132380), time spent (119893), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/1), picked move (CloudProcess-186 {CloudComputer-126}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.614 [main] DEBUG LS step (132381), time spent (119893), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/21), picked move (CloudProcess-225 {CloudComputer-127}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.614 [main] DEBUG LS step (132382), time spent (119893), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/21), picked move (CloudProcess-225 {CloudComputer-127}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.614 [main] DEBUG LS step (132383), time spent (119893), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/22), picked move (CloudProcess-191 {CloudComputer-128}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.614 [main] DEBUG LS step (132384), time spent (119893), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/22), picked move (CloudProcess-191 {CloudComputer-128}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132385), time spent (119894), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/23), picked move (CloudProcess-867 {CloudComputer-129}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132386), time spent (119894), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/77), picked move (CloudProcess-507 {CloudComputer-130}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132387), time spent (119894), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/44), picked move (CloudProcess-758 {CloudComputer-131}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132388), time spent (119894), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/20), picked move (CloudProcess-1039 {CloudComputer-132}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132389), time spent (119894), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/22), picked move (CloudProcess-698 {CloudComputer-133}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132390), time spent (119895), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/15), picked move (CloudProcess-813 {CloudComputer-134}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132391), time spent (119895), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/23), picked move (CloudProcess-493 {CloudComputer-135}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132392), time spent (119895), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/23), picked move (CloudProcess-493 {CloudComputer-135}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132393), time spent (119895), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/6), picked move (CloudProcess-431 {CloudComputer-136}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132394), time spent (119895), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/6), picked move (CloudProcess-1087 {CloudComputer-137}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.615 [main] DEBUG LS step (132395), time spent (119895), score (0hard/-492418soft), best score (0hard/-491620soft), accepted/selected move count (1/6), picked move (CloudProcess-1087 {CloudComputer-137}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.617 [main] DEBUG LS step (132396), time spent (119896), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/23), picked move (CloudProcess-1139 {CloudComputer-138}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.617 [main] DEBUG LS step (132397), time spent (119896), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/60), picked move (CloudProcess-52 {CloudComputer-139}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.617 [main] DEBUG LS step (132398), time spent (119896), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/60), picked move (CloudProcess-52 {CloudComputer-139}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.617 [main] DEBUG LS step (132399), time spent (119896), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/22), picked move (CloudProcess-116 {CloudComputer-140}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.618 [main] DEBUG LS step (132400), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/10), picked move (CloudProcess-1126 {CloudComputer-141}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.618 [main] DEBUG LS step (132401), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/77), picked move (CloudProcess-293 {CloudComputer-142}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.618 [main] DEBUG LS step (132402), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/77), picked move (CloudProcess-293 {CloudComputer-142}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.618 [main] DEBUG LS step (132403), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/55), picked move (CloudProcess-952 {CloudComputer-143}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.618 [main] DEBUG LS step (132404), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/11), picked move (CloudProcess-666 {CloudComputer-144}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.618 [main] DEBUG LS step (132405), time spent (119897), score (0hard/-492610soft), best score (0hard/-491620soft), accepted/selected move count (1/11), picked move (CloudProcess-666 {CloudComputer-144}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.618 [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-145}) <-> CloudProcess-263 {CloudComputer-124}
19:32:46.725 [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).

```

```

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**.

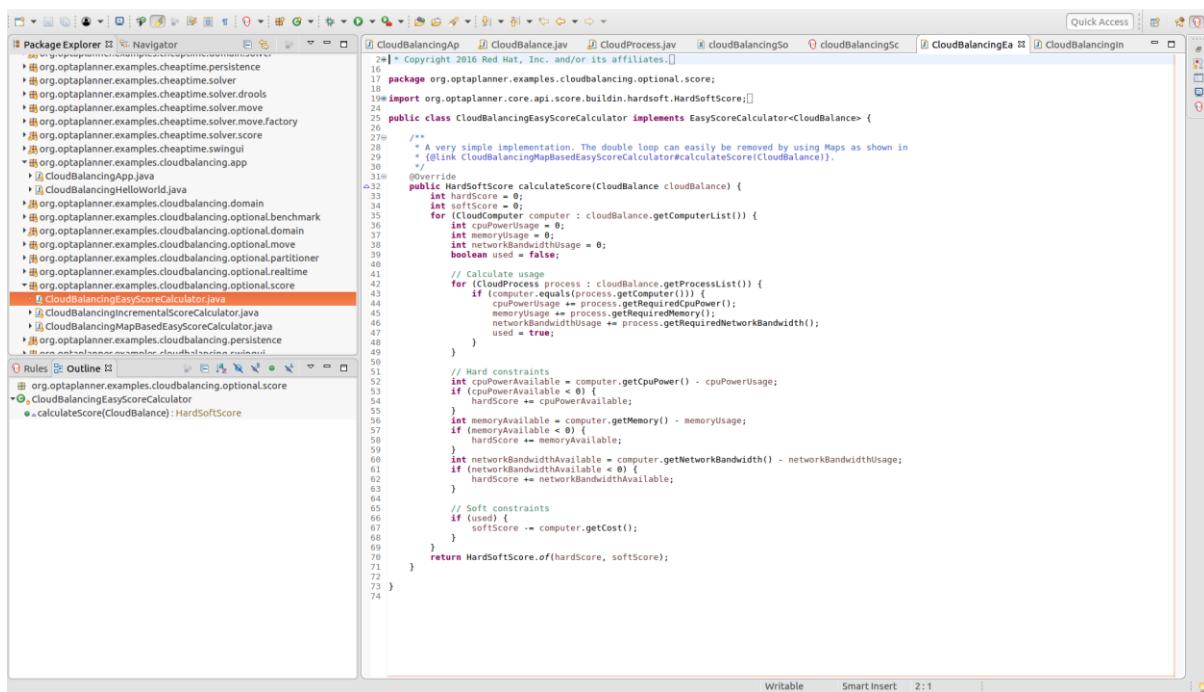
Update **SOLVER_CONFIG** file *cloudBalancingSolverConfig.xml*

From using Drools Rule:

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

To using Easy Java:

```
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbala
ncing.optional.score.CloudBalancingEasyScoreCalculator</easyS
coreCalculatorClass>
```



The screenshot shows the Eclipse IDE interface with the CloudBalancingApp project selected in the Package Explorer. The CloudBalancingEasyScoreCalculator.java file is open in the editor. The code implements the EasyScoreCalculator interface for CloudBalance. It calculates a hard score by summing up available resources minus usage for CPU, memory, and network bandwidth. It also calculates a soft score by summing up the cost of underutilized resources. The code uses annotations like @Override and implements the calculateScore(CloudBalance) method.

```

24/* Copyright 2016 Red Hat, Inc. and/or its affiliates. */
25
26package org.optaplanner.examples.cloudbalancing.optional.score;
27
28import org.optaplanner.examples.core.api.score.buildin.hardsoft.HardSoftScore;
29
30public class CloudBalancingEasyScoreCalculator implements EasyScoreCalculator<CloudBalance> {
31
32    /**
33     * A very simple implementation. The double loop can easily be removed by using Maps as shown in
34     * {@link CloudBalancingMapBasedEasyScoreCalculator#calculateScore(CloudBalance)}.
35     */
36
37    @Override
38    public HardSoftScore calculateScore(CloudBalance cloudBalance) {
39        int hardScore = 0;
40        int softScore = 0;
41        for (CloudComputer computer : cloudBalance.getComputerList()) {
42            int cpuPowerUsage = 0;
43            int cpuPowerAvailable = computer.getCpuPower();
44            int memoryUsage = 0;
45            int memoryAvailable = computer.getMemory();
46            int networkBandwidthUsage = 0;
47            int networkBandwidthAvailable = computer.getNetworkBandwidth();
48            boolean used = false;
49
50            // Calculate usage
51            for (CloudProcess process : computer.getProcessList()) {
52                if (process.getRequiredCpuPower() > 0) {
53                    cpuPowerUsage += process.getRequiredCpuPower();
54                    memoryUsage += process.getRequiredMemory();
55                    networkBandwidthUsage += process.getRequiredNetworkBandwidth();
56                    used = true;
57                }
58            }
59            // Hard constraints
60            if (cpuPowerAvailable - computer.getCpuPower() < 0) {
61                hardScore += cpuPowerAvailable;
62            }
63            if (memoryAvailable - computer.getMemory() < 0) {
64                hardScore += memoryAvailable;
65            }
66            if (networkBandwidthAvailable - computer.getNetworkBandwidth() < 0) {
67                hardScore += networkBandwidthAvailable;
68            }
69            // Soft constraints
70            if (used) {
71                softScore -= computer.getCost();
72            }
73        }
74    }
75
76    return HardSoftScore.of(hardScore, softScore);
77}

```

score calculation speed using Easy Java

```

22:44:26.626 [main] DEBUG CH step (245), time spent (119587), score (-924init/0hard/-181010soft), selected move count (330), picked move (CloudProcess-1158 {null -> CloudComputer-180}).
19:44:26.872 [main] DEBUG CH step (246), time spent (108272), score (-953init/0hard/-171410soft), selected move count (330), picked move (CloudProcess-1144 {null -> CloudComputer-239}).
19:44:27.125 [main] DEBUG CH step (247), time spent (108525), score (-952init/0hard/-171410soft), selected move count (330), picked move (CloudProcess-971 {null -> CloudComputer-189}).
19:44:27.398 [main] DEBUG CH step (248), time spent (108794), score (-950init/0hard/-171410soft), selected move count (330), picked move (CloudProcess-1006 {null -> CloudComputer-187}).
19:44:27.999 [main] DEBUG CH step (249), time spent (109057), score (-949init/0hard/-171410soft), selected move count (330), picked move (CloudProcess-764 {null -> CloudComputer-277}).
19:44:28.377 [main] DEBUG CH step (250), time spent (109395), score (-948init/0hard/-171410soft), selected move count (400), picked move (CloudProcess-312 {null -> CloudComputer-274}).
19:44:28.544 [main] DEBUG CH step (251), time spent (109737), score (-948init/0hard/-176210soft), selected move count (400), picked move (CloudProcess-312 {null -> CloudComputer-274}).
19:44:28.711 [main] DEBUG CH step (252), time spent (110079), score (-947init/0hard/-176210soft), selected move count (400), picked move (CloudProcess-312 {null -> CloudComputer-274}).
19:44:28.982 [main] DEBUG CH step (253), time spent (110380), score (-946init/0hard/-178610soft), selected move count (298), picked move (CloudProcess-224 {null -> CloudComputer-57}).
19:44:29.367 [main] DEBUG CH step (254), time spent (110677), score (-945init/0hard/-178610soft), selected move count (274), picked move (CloudProcess-1183 {null -> CloudComputer-159}).
19:44:29.534 [main] DEBUG CH step (255), time spent (111074), score (-944init/0hard/-178610soft), selected move count (274), picked move (CloudProcess-994 {null -> CloudComputer-127}).
19:44:30.020 [main] DEBUG CH step (256), time spent (111420), score (-943init/0hard/-178610soft), selected move count (288), picked move (CloudProcess-994 {null -> CloudComputer-127}).
19:44:30.353 [main] DEBUG CH step (257), time spent (111793), score (-942init/0hard/-178610soft), selected move count (247), picked move (CloudProcess-922 {null -> CloudComputer-43}).
19:44:30.520 [main] DEBUG CH step (258), time spent (112166), score (-941init/0hard/-178610soft), selected move count (247), picked move (CloudProcess-922 {null -> CloudComputer-43}).
19:44:30.931 [main] DEBUG CH step (259), time spent (112331), score (-940init/0hard/-178610soft), selected move count (299), picked move (CloudProcess-895 {null -> CloudComputer-274}).
19:44:31.246 [main] DEBUG CH step (260), time spent (112646), score (-939init/0hard/-178610soft), selected move count (305), picked move (CloudProcess-789 {null -> CloudComputer-277}).
19:44:31.623 [main] DEBUG CH step (261), time spent (112947), score (-938init/0hard/-178610soft), selected move count (298), picked move (CloudProcess-729 {null -> CloudComputer-112}).
19:44:32.091 [main] DEBUG CH step (262), time spent (113292), score (-937init/0hard/-178610soft), selected move count (292), picked move (CloudProcess-576 {null -> CloudComputer-23}).
19:44:32.323 [main] DEBUG CH step (263), time spent (113723), score (-936init/0hard/-178610soft), selected move count (188), picked move (CloudProcess-598 {null -> CloudComputer-338}).
19:44:33.016 [main] DEBUG CH step (264), time spent (114416), score (-935init/0hard/-178610soft), selected move count (312), picked move (CloudProcess-555 {null -> CloudComputer-257}).
19:44:33.383 [main] DEBUG CH step (265), time spent (115073), score (-934init/0hard/-178610soft), selected move count (298), picked move (CloudProcess-457 {null -> CloudComputer-57}).
19:44:34.353 [main] DEBUG CH step (266), time spent (115753), score (-933init/0hard/-178610soft), selected move count (298), picked move (CloudProcess-457 {null -> CloudComputer-57}).
19:44:34.869 [main] DEBUG CH step (267), time spent (116269), score (-932init/0hard/-178610soft), selected move count (306), picked move (CloudProcess-345 {null -> CloudComputer-359}).
19:44:35.236 [main] DEBUG CH step (268), time spent (116846), score (-931init/0hard/-178610soft), selected move count (306), picked move (CloudProcess-345 {null -> CloudComputer-359}).
19:44:36.208 [main] DEBUG CH step (269), time spent (117688), score (-930init/0hard/-178610soft), selected move count (304), picked move (CloudProcess-210 {null -> CloudComputer-230}).
19:44:36.516 [main] DEBUG CH step (270), time spent (117915), score (-929init/0hard/-178610soft), selected move count (304), picked move (CloudProcess-45 {null -> CloudComputer-239}).
19:44:37.182 [main] DEBUG CH step (271), time spent (118242), score (-928init/0hard/-178610soft), selected move count (304), picked move (CloudProcess-45 {null -> CloudComputer-239}).
19:44:37.361 [main] DEBUG CH step (272), time spent (118761), score (-927init/0hard/-181810soft), selected move count (400), picked move (CloudProcess-819 {null -> CloudComputer-423}).
19:44:37.845 [main] DEBUG CH step (273), time spent (119245), score (-926init/0hard/-181810soft), selected move count (313), picked move (CloudProcess-426 {null -> CloudComputer-422}).
19:44:38.187 [main] DEBUG CH step (274), time spent (119587), score (-925init/0hard/-181010soft), selected move count (313), picked move (CloudProcess-29 {null -> CloudComputer-422}).
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).

```

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).

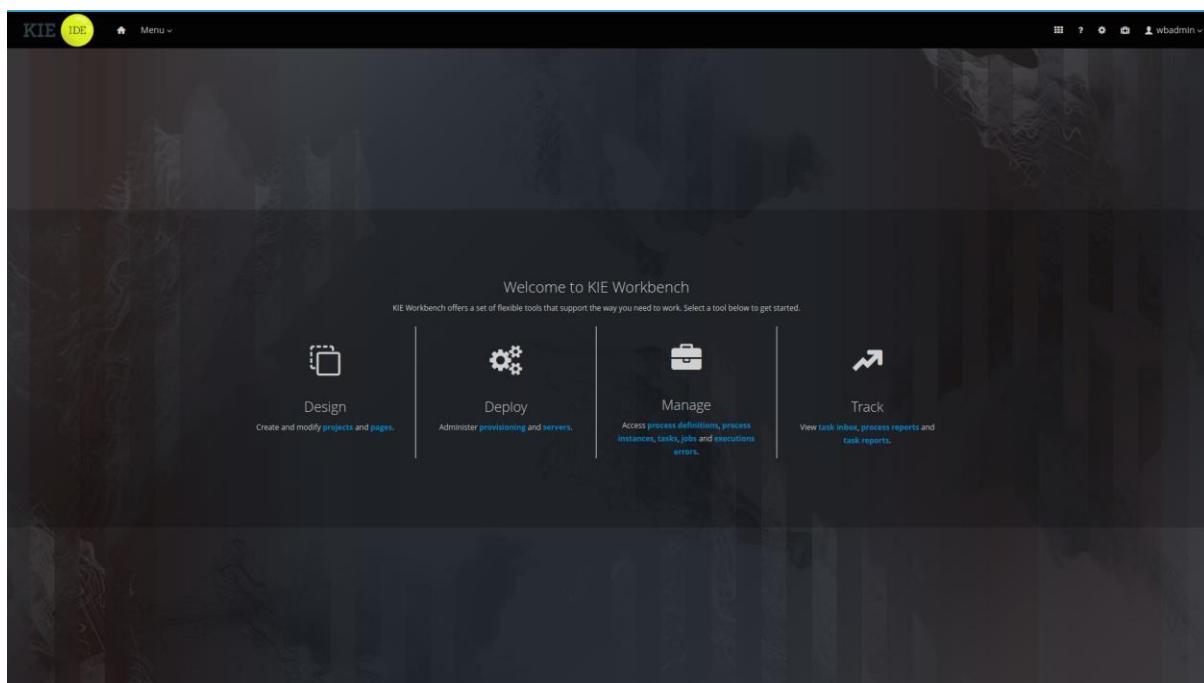
9) [Optional] { Challenge } score calculation speed using Incremental Java;

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

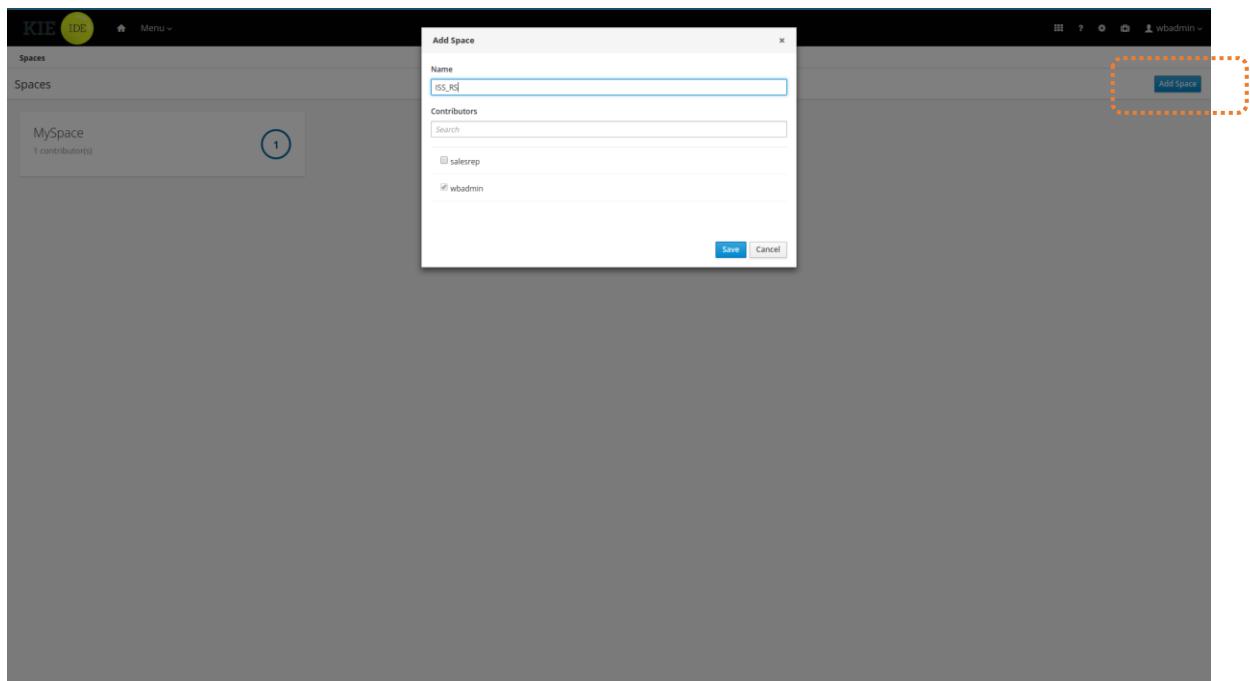
1) Install RESTful API tool: Postman

Refer to Annex 6;

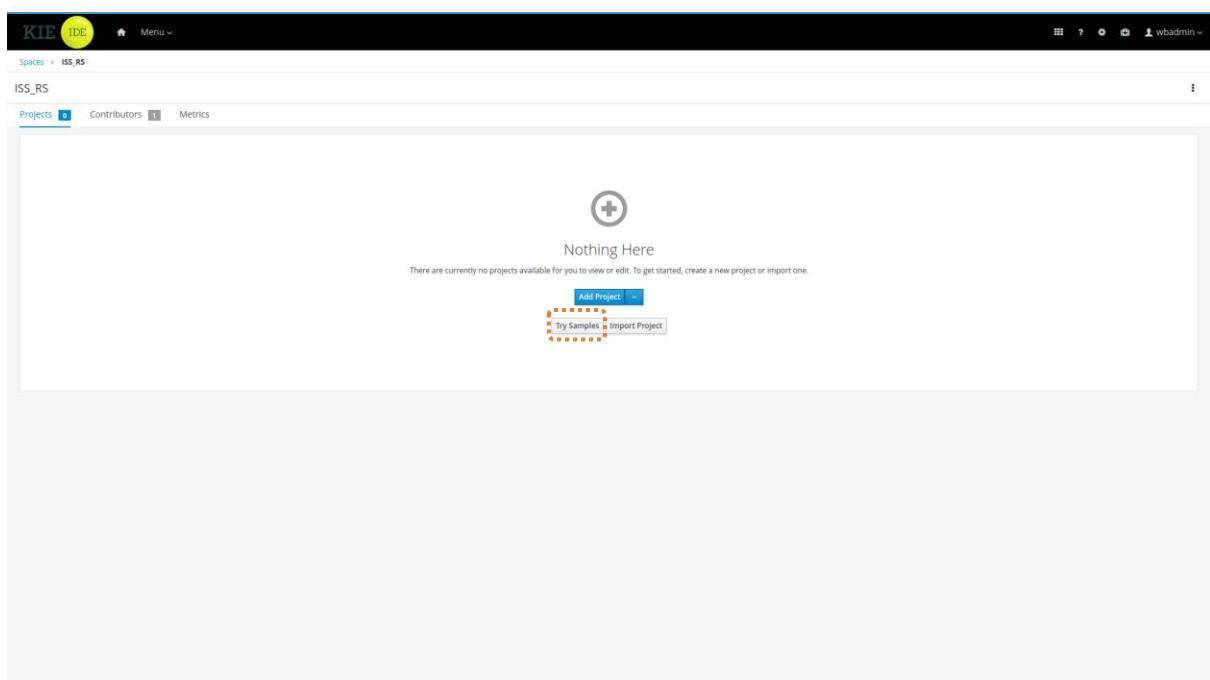
2) Start KIE Workbench



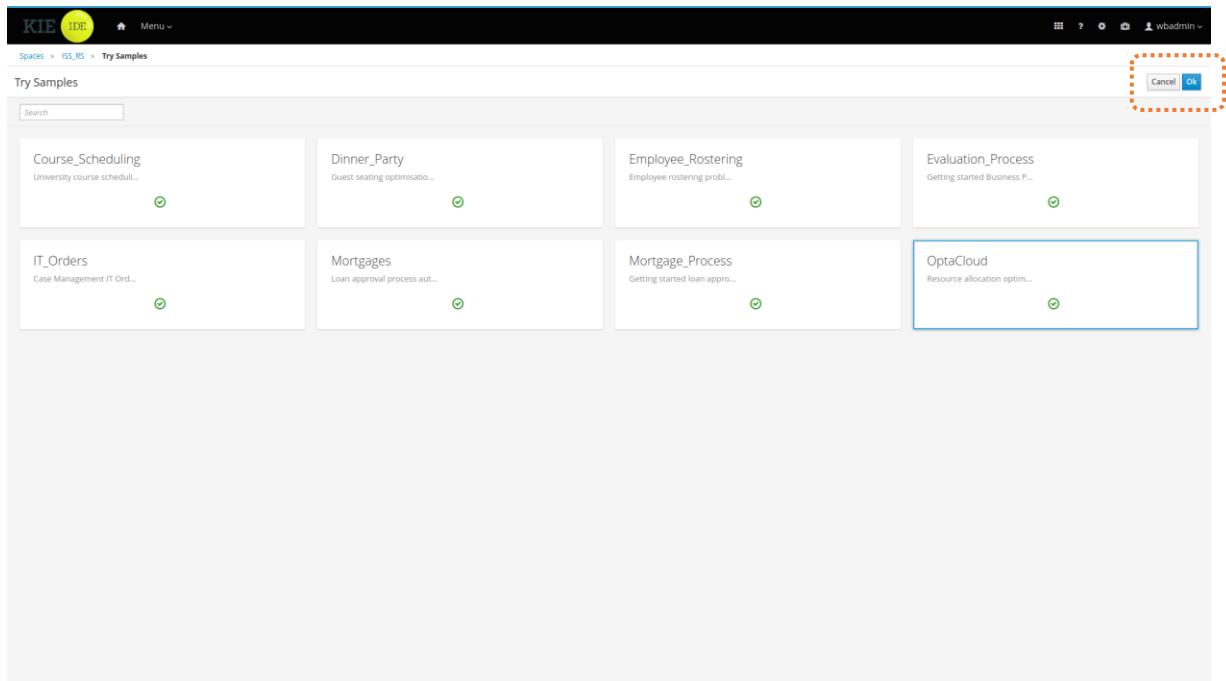
3) Create workspace: **ISS_RS**



4) Click **Try Samples**



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



Asset Type	Description	Last modified	Created
DRL	cloudScoreRules	today	today
Data Objects	CloudSolution	today	today
Globals Definitions	CloudSolutionScoreHolderGlobal	today	today
Solver configuration	cloudServerConfig	today	today
Data Objects	Computer	today	today
Data Objects	Process	today	today
Data Objects	SolverTest	today	today

Level	Text	File	Column	Line
Info	Completed indexing of MySpace/example-Mortgage_Process/master	-	0	0

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

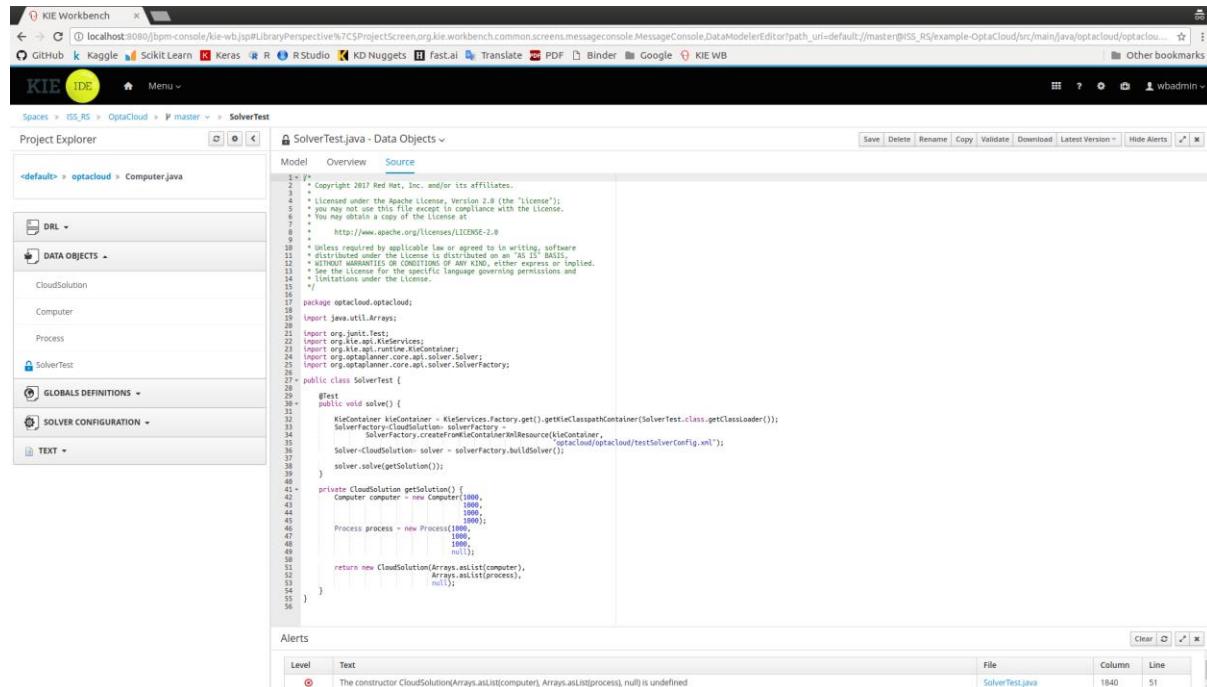
- 6) [Data Object] Add field: **id** (type: **long**) to Data Objects: **Computer**, **Process**, **CloudSolution**; Save all three objects;

KIE Workbench screenshot showing the creation of a new field 'id' for the 'Computer' data object. The 'Type' dropdown is set to 'long'. The 'Create' button is highlighted.

KIE Workbench screenshot showing the 'Computer.java - Data Objects' view. The 'DATA OBJECTS' section is highlighted with a red box. The 'Computer' data object is selected. A new field 'id' has been added and is highlighted with a red box. The 'Save' button is highlighted.

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



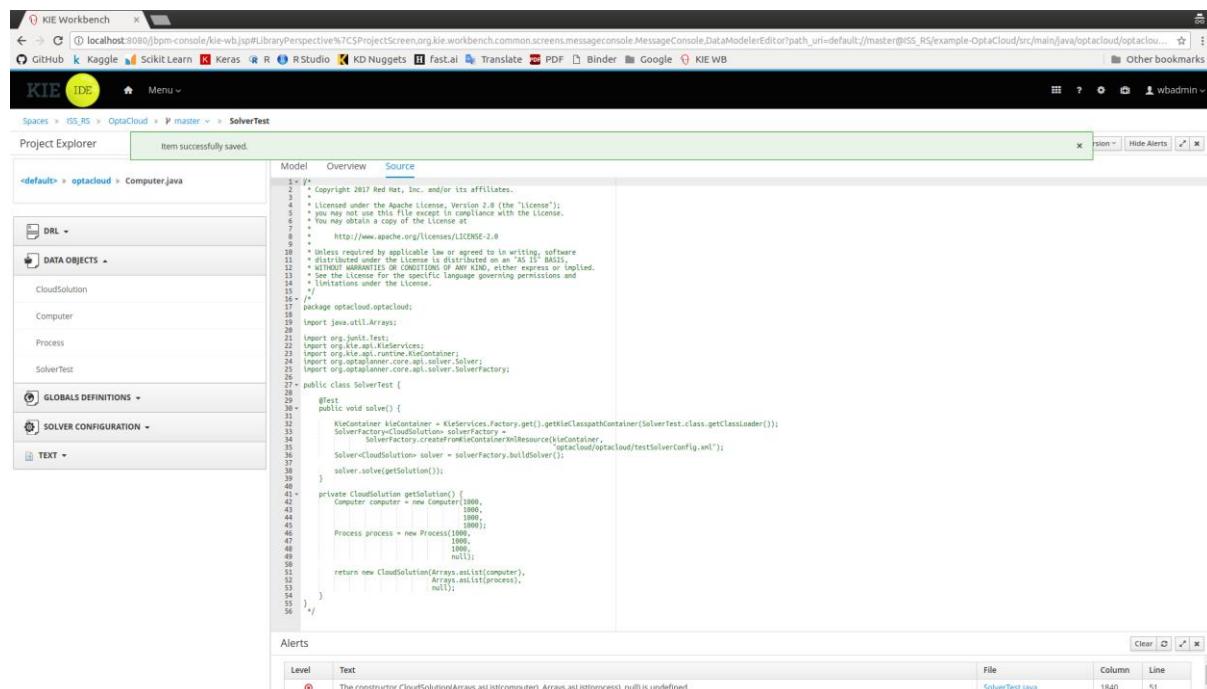
```

/*
 * 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 java.util.Arrays;
import org.junit.Test;
import org.kie.api.KieServices;
import org.kie.container.KieContainer;
import org.optaplanner.core.api.solver.Solver;
import org.optaplanner.core.api.solver.SolverFactory;
public class SolverTest {
    @Test
    public void test() {
        KieContainer kieContainer = KieServices.Factory.get().getKieClasspathContainer(SolverTest.class.getClassLoader());
        SolverFactory<CloudSolution> solverFactory = solverFactory.createFromKieContainerAndResource("kieContainer",
                "optacloud/optacloud/testSolverConfig.xml");
        CloudSolution solver = solverFactory.buildSolver();
        solver.solve();
    }
    private CloudSolution getSolution() {
        Computer computer = new Computer(1000,
                1000,
                1000);
        Process process = new Process(1000,
                1000,
                null);
        return new CloudSolution(Arrays.asList(computer),
                Arrays.asList(process),
                null);
    }
}

```

Level	Text	File	Column	Line
Info	The constructor CloudSolution(Arrays.asList(computer), Arrays.asList(process), null) is undefined	SolverTest.java	1840	51

Save



```

/*
 * 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 java.util.Arrays;
import org.junit.Test;
import org.kie.api.KieServices;
import org.kie.container.KieContainer;
import org.optaplanner.core.api.solver.Solver;
import org.optaplanner.core.api.solver.SolverFactory;
public class SolverTest {
    @Test
    public void test() {
        KieContainer kieContainer = KieServices.Factory.get().getKieClasspathContainer(SolverTest.class.getClassLoader());
        SolverFactory<CloudSolution> solverFactory = solverFactory.createFromKieContainerAndResource("kieContainer",
                "optacloud/optacloud/testSolverConfig.xml");
        CloudSolution solver = solverFactory.buildSolver();
        solver.solve();
    }
    private CloudSolution getSolution() {
        Computer computer = new Computer(1000,
                1000,
                1000);
        Process process = new Process(1000,
                1000,
                null);
        return new CloudSolution(Arrays.asList(computer),
                Arrays.asList(process),
                null);
    }
}

```

Level	Text	File	Column	Line
Info	The constructor CloudSolution(Arrays.asList(computer), Arrays.asList(process), null) is undefined	SolverTest.java	1840	51

Above: unused **SolverTest** java program

8) [Solver Configuration] Update

The screenshot shows the KIE Workbench interface for managing solver configurations. The project is 'cloudSolverConfig' under 'optacloud'. The 'Phase configuration' section is expanded, showing two main categories: 'Construction Heuristic' and 'Local Search'. The 'Local Search' dropdown is open, with 'Late Acceptance' highlighted in red.

The screenshot shows the KIE Workbench interface for managing solver configurations. The project is 'cloudSolverConfig' under 'optacloud'. The 'Phase configuration' section is expanded, showing two main categories: 'Construction Heuristic' and 'Local Search'. The 'Local Search' dropdown is open, with 'Late Acceptance' highlighted in red.

The screenshot shows the KIE Workbench interface with the 'cloudSolverConfig.solver.xml' file open. The 'Source' tab is selected. The XML code defines a solver configuration with various limits and search types. A red dashed box highlights the 'termination' section. The 'Project Explorer' on the left shows 'Computer.java' and 'cloudScoreRules'. The 'Alerts' panel at the bottom right shows a successful build message.

Above: solver configuration (xml)

example-OptaCloud/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

The screenshot shows the KIE Workbench interface with the following details:

- Toolbar:** Includes icons for KIE Workbench, GitHub, Kaggle, Scikit Learn, Keras, R, R Studio, KD Nuggets, fast.ai, Translate, PDF, Binder, Google, KIE WB, and Other bookmarks.
- Header:** Shows the URL localhost:8080/bpm-console/kie-wb/library/Perspective%7CProjectScreen>DataModelerEditor?path_=default://master@ISS_RS/example-OptaCloud/src/main/java/optacloud/optacloud/Computer.java&file_name=Computer.java&has_version_support=true.
- Sidebar:** Contains sections for Project Explorer, Model, Overview, and various configuration tabs like Data Objects, Globals Definitions, Solver Configuration, and Text.
- Central Area:** Displays the code for `cloudScoreRules.drl - DRL`. The code defines rules for computer scoring based on CPU power, memory, network bandwidth, and hosting costs.

```
12 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 * See the License for the specific language governing permissions and
14 * limitations under the License.
15 */
16 package optacloud.optacloud;
17
18 import optacloud.optacloud.Computer;
19 import optacloud.optacloud.Process;
20
21 //#####
22 //#####
23 //#####
24 //#####
25
26 rule "Hard CPU power capacity"
27 when
28   $computer : Computer($cpuPower : cpuPower)
29   $requiredCPUpowerTotal : Number($intValue > $cpuPower) from accumulate(
30     Process -- $computer == $computer,
31     requiredPower == $cpuPower,
32     sum($requiredPower),
33     $sum)
34   then
35     scoreHolder.addHardConstraintMatch($context, $cpuPower - $requiredCPUpowerTotal.intValue());
36   end
37
38 rule "Hard memory capacity"
39 when
40   $computer : Computer($memory : memory)
41   $requiredMemoryTotal : Number($intValue > $memory) from accumulate(
42     Process -- $computer == $computer,
43     requiredMemory == $memory,
44     sum($requiredMemory),
45     $sum)
46   then
47     scoreHolder.addHardConstraintMatch($context, $memory - $requiredMemory.intValue());
48   end
49
50 rule "Hard network bandwidth capacity"
51 when
52   $computer : Computer($networkBandwidth : networkBandwidth)
53   $requiredNetworkBandwidthTotal : Number($intValue > $networkBandwidth) from accumulate(
54     Process -- $computer == $computer,
55     requiredNetworkBandwidth == $networkBandwidth,
56     sum($requiredNetworkBandwidth),
57     $sum)
58   then
59     scoreHolder.addHardConstraintMatch($context, $networkBandwidth - $requiredNetworkBandwidth.intValue());
60   end
61
62 //#####
63 //#####
64 //#####
65 //#####
66 //#####
67
68 rule "Hosting cost"
69 when
70   $computer : Computer($cost : cost)
71   exists ProcessComputer -- $computer
72   then
73     scoreHolder.addSoftConstraintMatch($context, -$cost);
74   end
75
```

Above: constraints using Drools rule

10)[Deploy] Deploy OptaCloud solver to server

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

The screenshot shows the KIE Workbench interface with the URL localhost:8080/bpm-console/kie-wb.jsp#LibraryPerspective%7CProjectScreen,%5BWorg.kie.guvnor.explorer,%5D,_org.kie.workbench.common.screens.messageconsole.MessageConsole. The main content area displays a table of assets under the 'Assets' tab, with a green success message at the top stating 'Deploy to server configuration successful and container successfully started.' The table lists several assets: 'cloudScoreRules' (DRL), 'CloudSolution' (Data Objects), 'CloudSolutionScoreHolderGlobal' (Globals Definitions), 'cloudSolverConfig' (Solver configuration), 'Computer' (Data Objects), 'Process' (Data Objects), and 'SolverTest' (Data Objects). All assets were last modified today and created today. Below the table, there is an 'Alerts' section showing one text alert: 'Build of module 'OptaCloud' (requested by wbadmin) completed. Build: SUCCESSFUL'. The status bar at the bottom indicates '1-7 of 7' items.

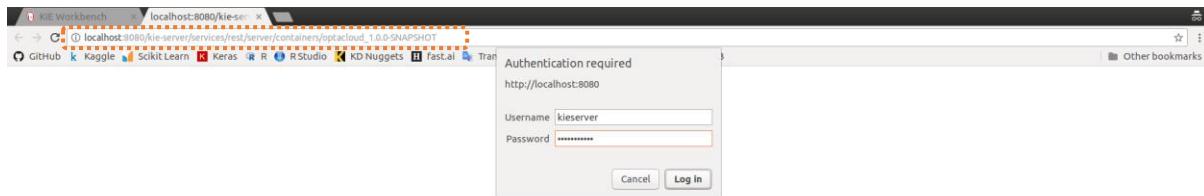
Deploy to KIE Server ***optacloud_1.0.0-SNAPSHOT***

The screenshot shows the KIE Workbench interface with the URL localhost:8080/bpm-console/kie-wb.jsp#ServerManagementPerspective%7CServerManagementBrowser. The left sidebar shows 'SERVER CONFIGURATIONS' with a 'sample-server' entry selected. Under 'sample-server', the 'DEPLOYMENT UNITS' section is expanded, showing 'optacloud_1.0.0-SNAPSHOT' highlighted. The right panel shows the 'optacloud' group with 'Version Configuration' selected. It displays the URL http://localhost:8080/kie-server/services/rest/server/containers/optacloud_1.0.0-SNAPSHOT and version v1.0.0-SNAPSHOT. There are 'Start', 'Stop', and 'Remove' buttons at the top of the right panel.

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

```

KIE Workbench | localhost:8080/kie ...
localhost:8080/kie-server/services/rest/server/containers/optacloud_1.0.0-SNAPSHOT
GitHub Kaggle Scikit Learn Keras R R Studio KD Nuggets fast.ai PDF Binder Google KIE WB Other bookmarks

This XML file does not appear to have any style information associated with it. The document tree is shown below.

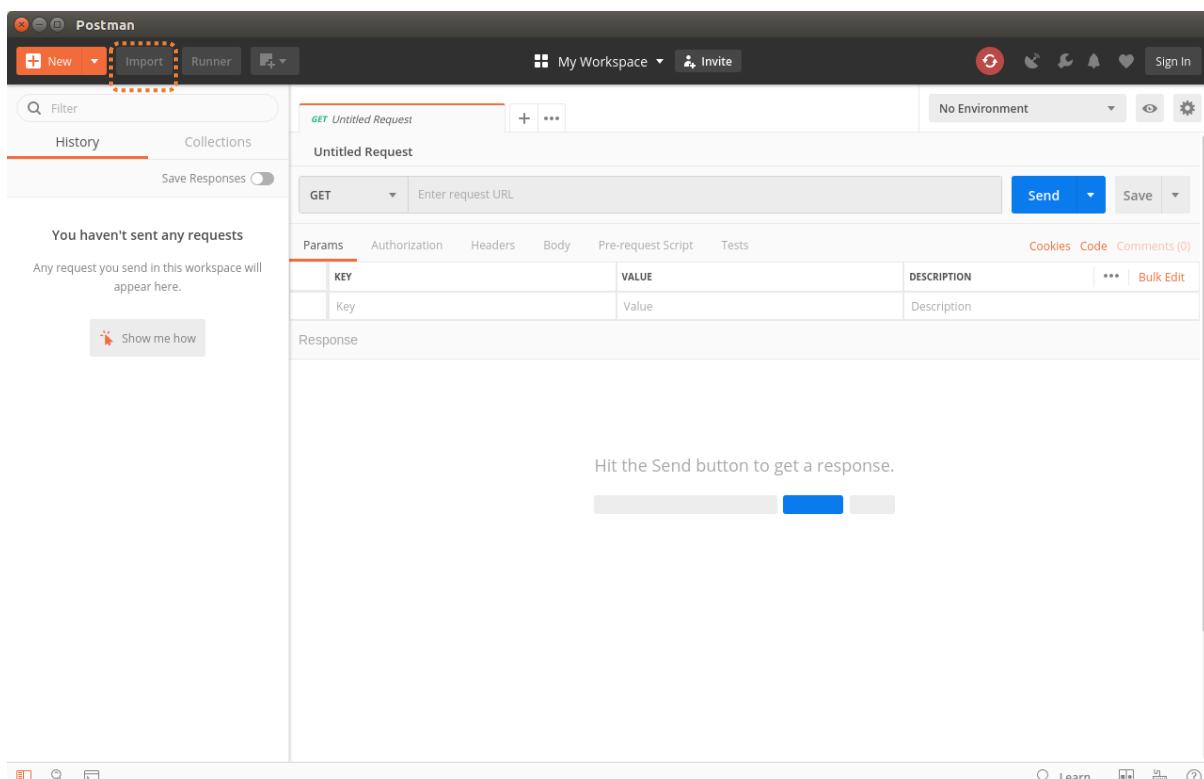
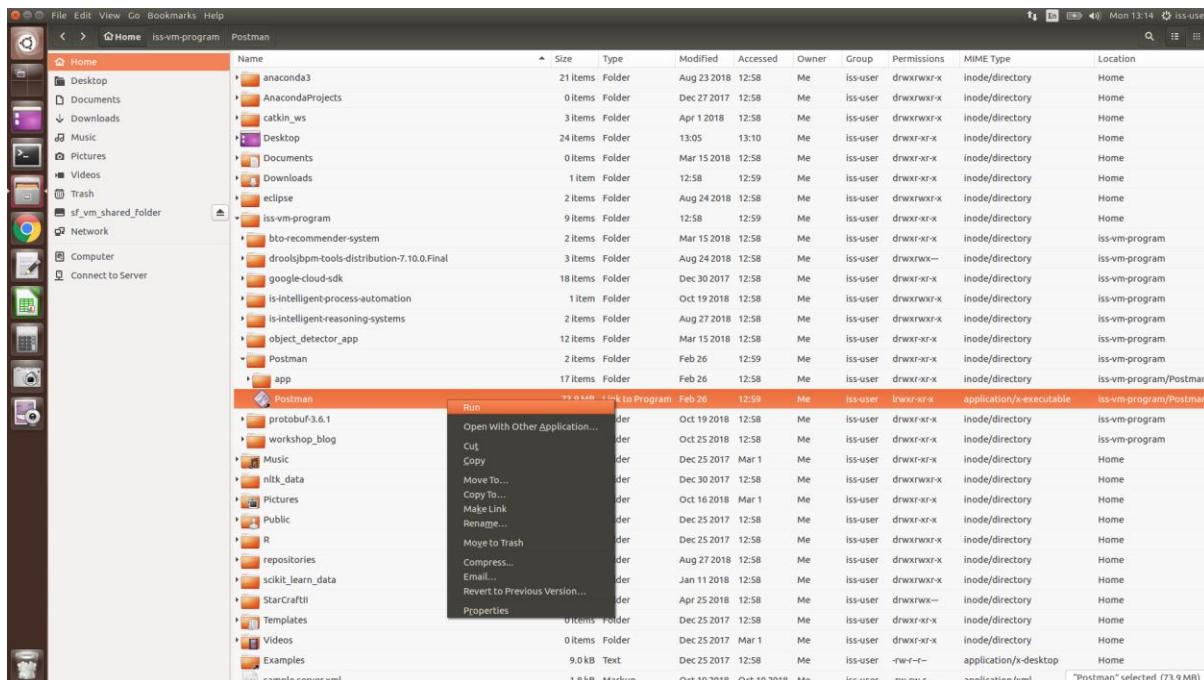
<?xml version="1.0" encoding="UTF-8"?>
<response type="SUCCESS" msg="Info for container optacloud_1.0.0-SNAPSHOT">
  <kie-container container-alias="optacloud" container-id="optacloud_1.0.0-SNAPSHOT" status="STARTED">
    <item>
      <itemName>KBase</itemName>
      <itemValue/>
      <itemType>BPMN</itemType>
    </item>
    <config-items>
      <config-item>
        <itemName>MergeMode</itemName>
        <itemValue>MERGE_COLLECTIONS</itemValue>
        <itemType>BPMN</itemType>
      </config-item>
      <config-item>
        <itemName>ContainerItemStrategy</itemName>
        <itemValue>SINGLETON</itemValue>
        <itemType>BPMN</itemType>
      </config-item>
    </config-items>
    <content>
      Container optacloud_1.0.0-SNAPSHOT successfully created with module optacloud:optacloud:1.0.0-SNAPSHOT.
    </content>
  </kie-container>
  <severity>INFO</severity>
  <timestamp>2019-03-04T14:21:03.883+08:00</timestamp>
</response>
<release-ids>
  <artifact-id>optacloud</artifact-id>
  <group-id>optacloud</group-id>
  <version>1.0.0-SNAPSHOT</version>
</release-ids>
<scanner-ids>
  <artifact-id>optacloud</artifact-id>
  <group-id>optacloud</group-id>
  <version>1.0.0-SNAPSHOT</version>
  <status>DISPOSED</status>
</scanner-ids>
</kie-container>

```

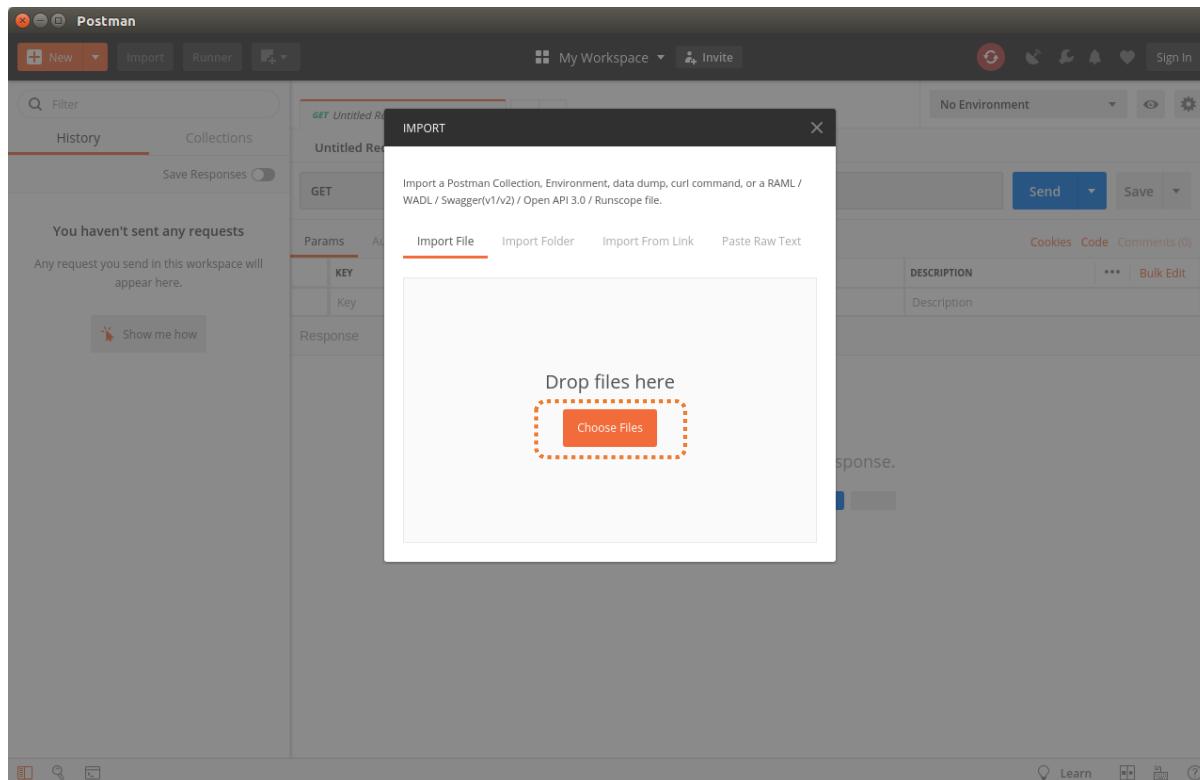
```
<response type="SUCCESS" msg="Info for container optacloud_1.0.0-SNAPSHOT">
<kie-container container-alias="optacloud" container-id="optacloud_1.0.0-SNAPSHOT" status="STARTED">
<config-items>
<itemName>KBase</itemName>
<itemValue/>
<itemType>BPM</itemType>
</config-items>
<config-items>
<itemName>KSession</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

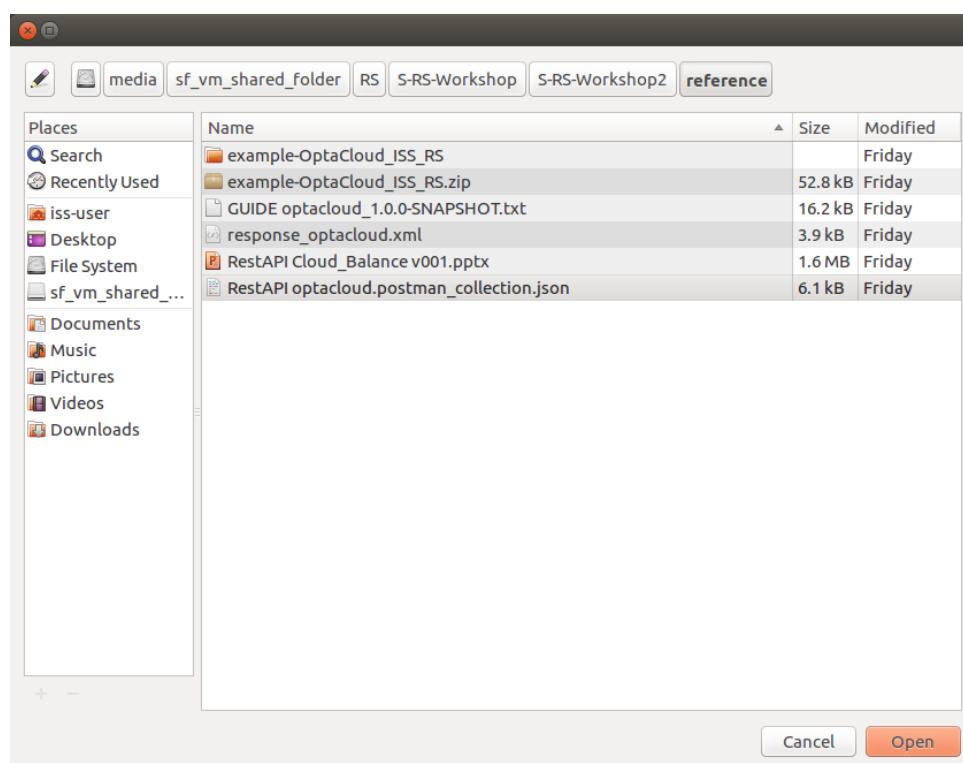
Start tool Postman



Import Pre-built API calls for OptaCloud project



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



The screenshot shows the Postman application interface. On the left, the sidebar displays 'RestAPI optacloud' with three requests: 'solvers/cloudBalancingSolver', 'solvers/cloudBalancingSolver/state/solving', and 'solvers/cloudBalancingSolver/bestsolution'. The main workspace shows an 'Untitled Request' with a 'GET' method selected. The 'Params' tab is active, showing a single entry: 'Key' under 'KEY' and 'Value' under 'VALUE'. A message at the bottom says 'Hit the Send button to get a response.'

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



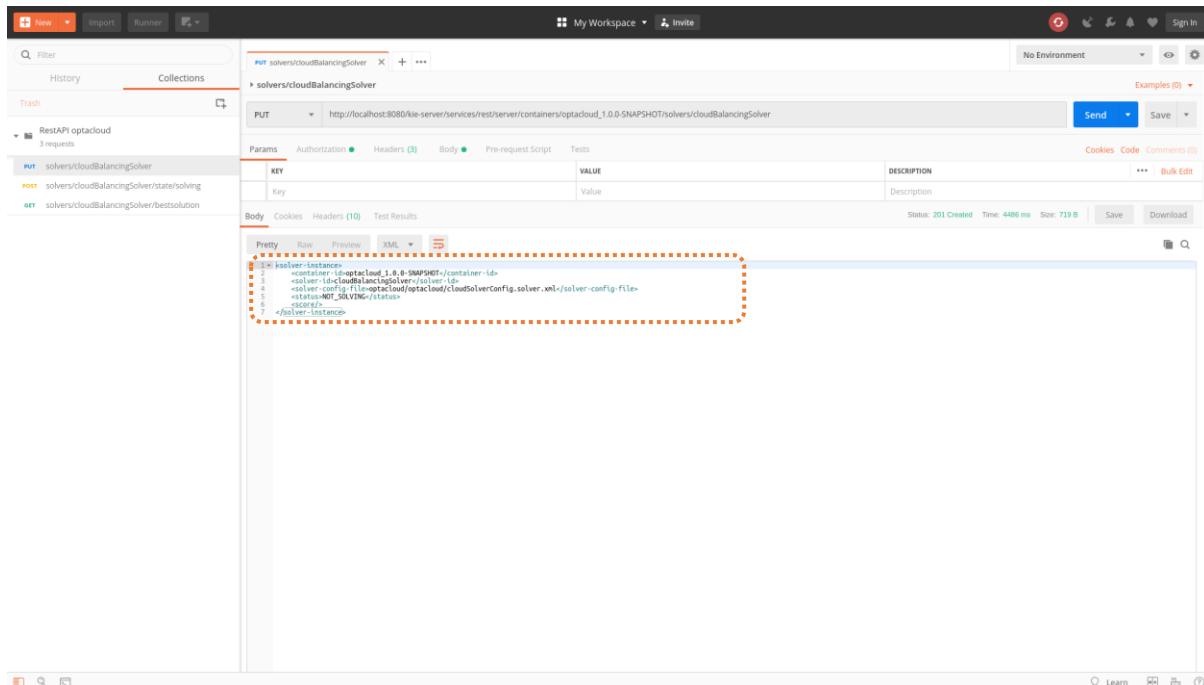
PUT – Initiate a solver object-instance

The screenshot shows the Postman interface with a PUT request to the URL `http://localhost:8080/kie-server/services/rest/server/containers/optacloud_1.0.0-SNAPSHOT/solvers/cloudBalancingSolver`. The Headers tab displays the following configuration:

KEY	VALUE
Key	Value

Below the Headers, the Response tab shows the status: 201 Created, Time: 4496 ms, Size: 719 B.

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



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

```

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

```



POST – Start the solver with biz context

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'History' and 'Collections'. Under 'Collections', there's a section for 'RestAPI optacloud' containing three requests: 'PUT solvers/cloudBalancingSolver', 'POST solvers/cloudBalancingSolver/state/solving', and 'GET solvers/cloudBalancingSolver/bestsolution'. The main area shows a 'POST' request to 'http://localhost:9080/kie-server/services/rest/server/containers/optacloud_1.0.0-SNAPSHOT/solvers/cloudBalancingSolver/state/solving'. The 'Headers' tab is selected, showing two headers: 'X-KIE-Content-Type' set to 'xstream' and 'Content-Type' set to 'application/xml'. Below the request, the 'Test Results' tab is open, displaying the response status as '200 OK' with a time of '84 ms' and a size of '440 B'. The response body contains various HTTP headers and a long session ID.

```

HTTP/1.1 200 OK
Date: Mon, 04 Mar 2019 06:23:30 GMT
Server: WildFly/11
Content-Type: application/xml; charset=UTF-8
Content-Length: 0
Connection: keep-alive
X-KIE-ConversationId: %27sample-server%27%3A%27optacloud_1.0.0-SNAPSHOT%27%3A%27optacloud%3A%27%3A%274b47999-737e-4f05-b054-b74b59584041%27
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: 0
X-Powered-By: Undertow/1

```

Above: REST-API POST **biz context** via Postman tool

```

POST /solvers/cloudBalancingSolver/state/solving
Content-Type: application/xml

<!DOCTYPE problem SYSTEM "http://optacloud.org/optacloud/CloudBalancing.dtd">
<problem>
    <computerList id="2">
        <id>1</id>
        <cpuPower>24</cpuPower>
        <memory>56</memory>
        <networkBandwidth>10</networkBandwidth>
        <cost>480</cost>
    </computer>
    <computerList id="4">
        <id>2</id>
        <cpuPower>4</cpuPower>
        <memory>4</memory>
        <networkBandwidth>4</networkBandwidth>
        <cost>640</cost>
    </computer>
    <processList id="5">
        <id>3</id>
        <cpuPower>1</cpuPower>
        <memory>1</memory>
        <networkBandwidth>1</networkBandwidth>
        <cost>100</cost>
    </process>
    <processList id="6">
        <id>4</id>
        <cpuPower>1</cpuPower>
        <memory>1</memory>
        <networkBandwidth>1</networkBandwidth>
        <cost>100</cost>
    </process>
    <processList id="7">
        <id>5</id>
        <cpuPower>1</cpuPower>
        <memory>1</memory>
        <networkBandwidth>1</networkBandwidth>
        <cost>100</cost>
    </process>
    <processList id="8">
        <id>6</id>
        <cpuPower>1</cpuPower>
        <memory>1</memory>
        <networkBandwidth>1</networkBandwidth>
        <cost>100</cost>
    </process>
    <processList id="9">
        <id>7</id>
        <cpuPower>1</cpuPower>
        <memory>2</memory>
        <networkBandwidth>2</networkBandwidth>
        <cost>200</cost>
    </process>
    <processList id="10">
        <id>8</id>
        <cpuPower>1</cpuPower>
        <memory>2</memory>
        <networkBandwidth>2</networkBandwidth>
        <cost>200</cost>
    </process>
    <processList id="11">
        <id>9</id>
        <cpuPower>1</cpuPower>
        <memory>2</memory>
        <networkBandwidth>2</networkBandwidth>
        <cost>200</cost>
    </process>
</problem>
  
```

Above: REST-API POST biz context via Postman tool

```
<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)

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/kie-server/services/rest/server/containers/optacloud_1.0.0-SNAPSHOT/solvers/cloudBalancingSolver/bestsolution
- Headers:**
 - Authorization: Basic a2lkZ2ydmVyoMtpZHNlcjEh
 - Key: Value
- Body:** XML (Pretty, Raw, Preview)
- Response:** Status: 200 OK, Time: 443 ms, Size: 4.22 KB

```

<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<solvers>
  <solver-instance>
    <solver-id>optacloud_1.0.0-SNAPSHOT</solver-id>
    <solver-id>cloudBalancingSolver</solver-id>
    <status>NOT_RUNNING</status>
    <solver-config-file>optacloud/cloudServerConfig.solver.xml</solver-config-file>
    <score>1</score>
    <score-class>org.optaplanner.core.api.score.buildin.hardsoft.HardsoftScore</score-class>
    <score-factory>org.optaplanner.core.api.score.buildin.hardsoft.HardsoftScoreFactory</score-factory>
    <solver-score-factory>org.optaplanner.core.api.score.buildin.hardsoft.HardsoftScoreFactory</solver-score-factory>
    <solver-score-factory-fqn>org.optaplanner.core.impl.solver.config.factories.HardsoftScoreFactory</solver-score-factory-fqn>
    <solver-score-factory-xml><!--留空--></solver-score-factory-xml>
    <computerList>
      <computer>
        <cost>1</cost>
        <cpuPower>24</cpuPower>
        <id>1</id>
        <memory>96</memory>
        <networkBandwidth>16</networkBandwidth>
      </computer>
      <computer>
        <cost>1</cost>
        <cpuPower>4</cpuPower>
        <id>2</id>
        <memory>4</memory>
        <networkBandwidth>4</networkBandwidth>
      </computer>
      <computer>
        <cost>1</cost>
        <cpuPower>24</cpuPower>
        <id>3</id>
        <memory>96</memory>
        <networkBandwidth>16</networkBandwidth>
      </computer>
      <computer>
        <cost>1</cost>
        <cpuPower>4</cpuPower>
        <id>4</id>
        <memory>4</memory>
        <networkBandwidth>4</networkBandwidth>
      </computer>
    </computerList>
    <processList>
      <process>
        <cost>1</cost>
        <cpuPower>24</cpuPower>
        <id>1</id>
        <memory>96</memory>
        <networkBandwidth>16</networkBandwidth>
      </process>
      <process>
        <cost>1</cost>
        <cpuPower>24</cpuPower>
        <id>2</id>
        <memory>96</memory>
        <networkBandwidth>16</networkBandwidth>
      </process>
      <process>
        <cost>1</cost>
        <cpuPower>24</cpuPower>
        <id>3</id>
        <memory>96</memory>
        <networkBandwidth>16</networkBandwidth>
      </process>
      <process>
        <cost>1</cost>
        <cpuPower>24</cpuPower>
        <id>4</id>
        <memory>96</memory>
        <networkBandwidth>16</networkBandwidth>
      </process>
    </processList>
  </solver-instance>
</solvers>

```

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 for editing the 'Computer.java' file. The left sidebar shows the project structure under 'optacloud'. The main area displays the 'Computer' data object with its fields: cost, cpuPower, id, memory, networkBandwidth, and gpuPower. A new 'gpuPower' row has been added to the table, highlighted with a red dashed box. To the right, the 'gpuPower' general properties panel is open, showing its identifier as 'gpuPower', label as 'GPU Power', type as 'int', and description as ''gpuPower'- general properties''. At the top right of the main area, there is a toolbar with buttons for Save, Delete, Rename, Copy, Validate, Download, Latest Version, Hide Alerts, and Close. The 'Save' button is circled in red.

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

The screenshot shows the KIE Workbench interface for editing Java code. The left sidebar shows a project structure with 'Process.java' selected. The main area displays the code for 'Process.java'.

```

    package optacloud.optacloud.Com...
    public class Process {
        @Id
        private long id;
        private String requiredCpuPower;
        private String requiredMemory;
        private String requiredNetworkBandwidth;
        private String requiredGpuPower;
    }
  
```

A new field, 'requiredGpuPower', has been added to the class definition. The 'Model' tab is selected in the top navigation bar. A context menu is open over the newly added field, with the 'Save' option highlighted.

The right side of the screen shows the 'Process.java - Data Objects' panel, which lists the fields of the 'Process' class. The 'requiredGpuPower' field is visible there.

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**?

```

12 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 * See the License for the specific language governing permissions and
14 * limitations under the License.
15 */
16 package optacloud.optacloud;
17
18 import optacloud.optacloud.Computer;
19 import optacloud.optacloud.Process;
20
21 // Hard constraints
22 //#####
23
24 rule "Hard CPU power capacity"
25 when
26   $computer : Computer($cpuPower : $cpuPower)
27   $requiredCpuPowerTotal : Number($intValue = $cpuPower) from accumulate(
28     Process
29     computer == $computer
30     $requiredCpuPower >= $cpuPower
31     sum($requiredCpuPower)
32   )
33   then
34     scoreholder.addHardConstraintMatch($context, $cpuPower - $requiredCpuPowerTotal.intValue());
35 end
36
37 rule "Hard memory capacity"
38 when
39   $computer : Computer($memory : $memory)
40   $requiredMemoryTotal : Number($intValue = $memory) from accumulate(
41     Process
42     computer == $computer
43     $requiredMemory >= $memory
44     sum($requiredMemory)
45   )
46   then
47     scoreholder.addHardConstraintMatch($context, $memory - $requiredMemoryTotal.intValue());
48 end
49
50 rule "Hard network bandwidth capacity"
51 when
52   $computer : Computer($networkBandwidth : $networkBandwidth)
53   $requiredNetworkBandwidthTotal : Number($intValue = $networkBandwidth) from accumulate(
54     Process
55     computer == $computer
56     $requiredNetworkBandwidth >= $networkBandwidth
57     sum($requiredNetworkBandwidth)
58   )
59   then
60     scoreholder.addHardConstraintMatch($context, $networkBandwidth - $requiredNetworkBandwidthTotal.intValue());
61 end
62
63 //#####
64
65 // Soft constraints
66 //#####
67
68 rule "Soft cost"
69 when
70   $computer : Computer($cost : $cost)
71   exists Process($computer == $computer)
72   then
73     scoreholder.addSoftConstraintMatch($context, -$cost);
74 end

```

```
/*
 * 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 #####
// 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

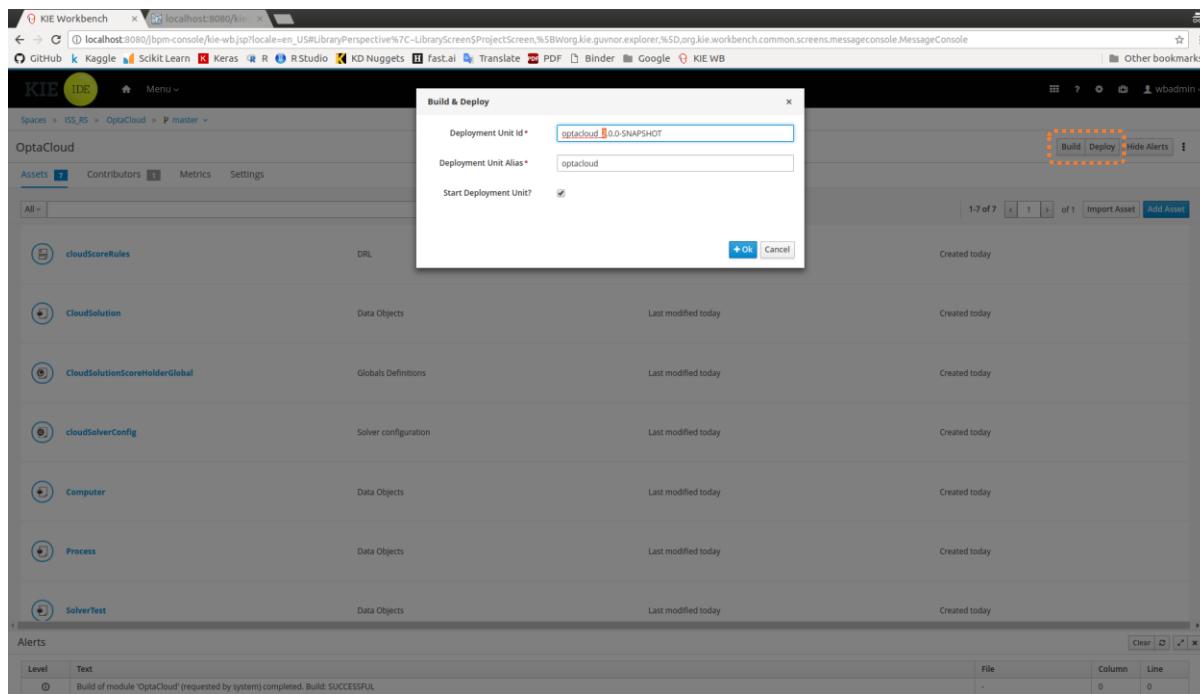
rule "Hard network bandwidth capacity"
when
    $computer : Computer($networkBandwidth : networkBandwidth)
    $requiredNetworkBandwidthTotal : Number(intValue > $networkBandwidth) from accumulate(
        Process(
            computer == $computer,

```

```
$requiredNetworkBandwidth : requiredNetworkBandwidth),  
sum($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
```

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**

1.2.4. Cloud Balance Solver Enhancement [API]

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

The screenshot shows the Postman API client interface. On the left, the 'Trash' section contains two collections: 'RestAPI optacloud' and 'RestAPI optacloud v2.0.0', both highlighted with a red dashed box. The main workspace shows a 'PUT solvers/cloudBalancingSolver' request. The 'Body' tab displays the XML payload for initializing a solver instance:

```
<!><solver><container-id>optacloud_2.0.0-SNAPSHOT</container-id>
<solver-id>cloudBalancingSolver</solver-id>
<solver-type>optacloud</solver-type>
<cloudServerConfig.solver.xml></solver-config-file>
<state>NOT_SOLVING</state>
<solver>
</solver>
</solver>
```

- 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.

The screenshot shows the Red Hat OpenShift developer console interface. The top navigation bar includes 'New', 'Import', 'Runner', 'Filter', 'My Workspace', 'Invite', and environment selection. A sidebar on the left lists 'History' and 'Collections', with 'RestAPI optacloud' and 'RestAPI optacloud v2.0.0' expanded to show their contents. The main area displays a POST request to 'solvers/cloudBalancingSolver/state/solving'. The 'Body' tab is selected, showing the XML payload for the request:

```
POST http://localhost:8080/kie-server/services/rest/server/containers/optacloud_2.0-SNAPSHOT/solvers/cloudBalancingSolver/state/solving
```

XML (application/xml) content:

```
<?xml version="1.0"?>
<planning-problem class="optacloud.optacloud.CloudSolution" id="1">
  <cloud/>
  <computerList id="2">
    <optacloud.optacloud.Computer id="3">
      <cpuPower>24</cpuPower>
      <memory>4</memory>
      <networkBandwidth>16</networkBandwidth>
      <gpuPower>4</gpuPower>
      <cpuPower>7800</cpuPower>
    </optacloud.optacloud.Computer>
    <optacloud.optacloud.Computer id="4">
      <cpuPower>14</cpuPower>
      <memory>4</memory>
      <networkBandwidth>4</networkBandwidth>
      <gpuPower>4</gpuPower>
      <cpuPower>7800</cpuPower>
    </optacloud.optacloud.Computer>
    <optacloud.optacloud.Computer id="5">
      <optacloud.optacloud.Process id="6">
        <requiredCpuPower>1</requiredCpuPower>
        <requiredMemory>1</requiredMemory>
        <requiredNetworkBandwidth>1</requiredNetworkBandwidth>
        <optacloud.optacloud.Process id="7">
          <cpuPower>2</cpuPower>
          <memory>1</memory>
          <networkBandwidth>1</networkBandwidth>
          <gpuPower>1</gpuPower>
        </optacloud.optacloud.Process>
        <optacloud.optacloud.Process id="8">
          <cpuPower>2</cpuPower>
          <memory>2</memory>
          <networkBandwidth>2</networkBandwidth>
          <gpuPower>2</gpuPower>
        </optacloud.optacloud.Process>
        <optacloud.optacloud.Process id="9">
          <cpuPower>2</cpuPower>
          <memory>2</memory>
          <networkBandwidth>2</networkBandwidth>
          <gpuPower>2</gpuPower>
        </optacloud.optacloud.Process>
        <optacloud.optacloud.Process id="10">
          <cpuPower>2</cpuPower>
          <memory>2</memory>
          <networkBandwidth>2</networkBandwidth>
          <gpuPower>2</gpuPower>
        </optacloud.optacloud.Process>
      </optacloud.optacloud.Process>
      <optacloud.optacloud.Process id="11">
        <cpuPower>2</cpuPower>
        <memory>2</memory>
        <networkBandwidth>2</networkBandwidth>
        <gpuPower>2</gpuPower>
      </optacloud.optacloud.Process>
    </optacloud.optacloud.Computer>
  </computerList>
</planning-problem>
```

```
<planning-problem class="optacloud.optacloud.CloudSolution" id="1">

<id>0</id>

<computerList id="2">

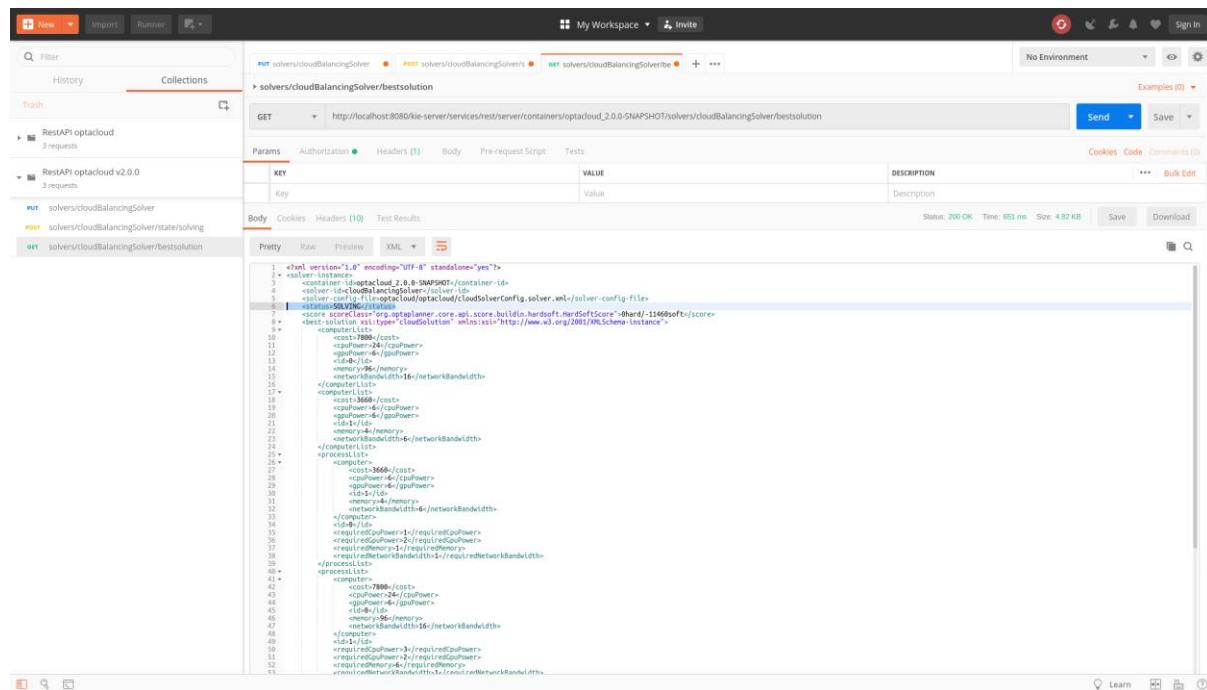
    <optacloud.optacloud.Computer id="3">
        <id>0</id>
        <cpuPower>24</cpuPower>
        <memory>96</memory>
        <networkBandwidth>16</networkBandwidth>
        <gpuPower>6</gpuPower>
        <cost>7800</cost>
    </optacloud.optacloud.Computer>
    <optacloud.optacloud.Computer id="4">
        <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>
<b><requiredGpuPower>4</requiredGpuPower></b>
</optacloud.optacloud.Process>
<optacloud.optacloud.Process id="9">
<id>3</id>
<requiredCpuPower>1</requiredCpuPower>
<requiredMemory>2</requiredMemory>
<requiredNetworkBandwidth>11</requiredNetworkBandwidth>
<b><requiredGpuPower>0</requiredGpuPower></b>
</optacloud.optacloud.Process>
<optacloud.optacloud.Process id="10">
<id>4</id>
<requiredCpuPower>1</requiredCpuPower>
<requiredMemory>1</requiredMemory>
<requiredNetworkBandwidth>1</requiredNetworkBandwidth>
<b><requiredGpuPower>0</requiredGpuPower></b>
</optacloud.optacloud.Process>
<optacloud.optacloud.Process id="11">
<id>5</id>
<requiredCpuPower>1</requiredCpuPower>
<requiredMemory>1</requiredMemory>
<requiredNetworkBandwidth>5</requiredNetworkBandwidth>
<b><requiredGpuPower>0</requiredGpuPower></b>
</optacloud.optacloud.Process>
</processList>
</planning-problem>
```

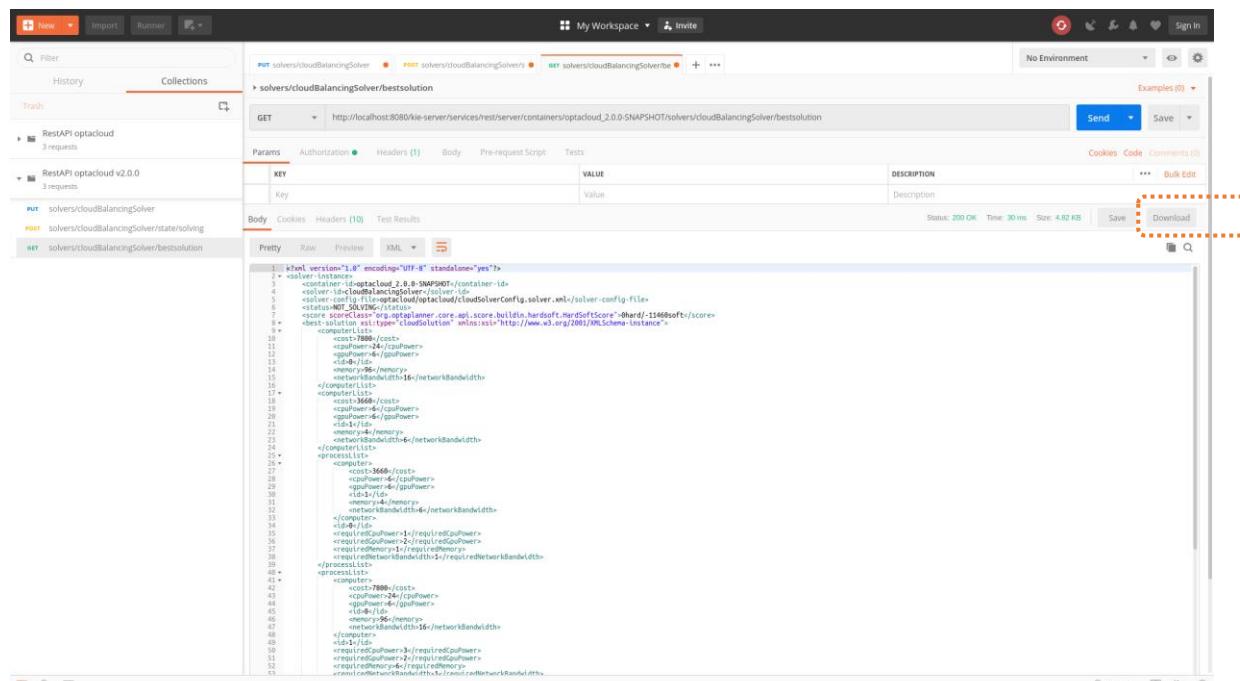
3) [API: GET] Retrieve best solution (with GPU requirements);



```

<status>SOLVING</status>

```



```

<status>NOT_SOLVING</status>

```

{ 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.

1.3. Hybrid Reasoning System [MTech Thru-Train Group Project]

{ Objective } Identify a relevant business scenario/problem. Propose and create a hybrid reasoning system, e.g. build on top of Machine Reasoning group project.

Form a project team of **4 - 6** members, appoint a team leader, enrol in LumiNUS project group.

The proposed project must develop, integrate, and demonstrate at least **two** out of following **three** technique groups:

1. Business rule **OR** Business process based reasoning techniques
2. Business resource optimization techniques: Heuristic search **OR** Constraint satisfaction **OR** Evolutional computing [Lecture 3]
3. Knowledge Discovery **OR** Data Mining techniques [Lecture 4]

The submitted runnable system should have a graphical user interface for end user to input or update data to execute different business scenarios, e.g. to enable initial planning and frequent re-planning/re-optimization. And to display system output results in a user friendly manner. (Input/output using xml file or console log is **not** considered as user friendly.)

Candidate/Example Project:

ANNEX 1 WORKSHOP PROJECT CANDIDATE

Project Submission Guideline:

ANNEX 3 WORKSHOP PROJECT SUBMISSION

OptaPlanner Domain Modelling Reference

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

😊 Congratulations!

You have completed today's challenging workshop!

The End of Workshop Project Guide