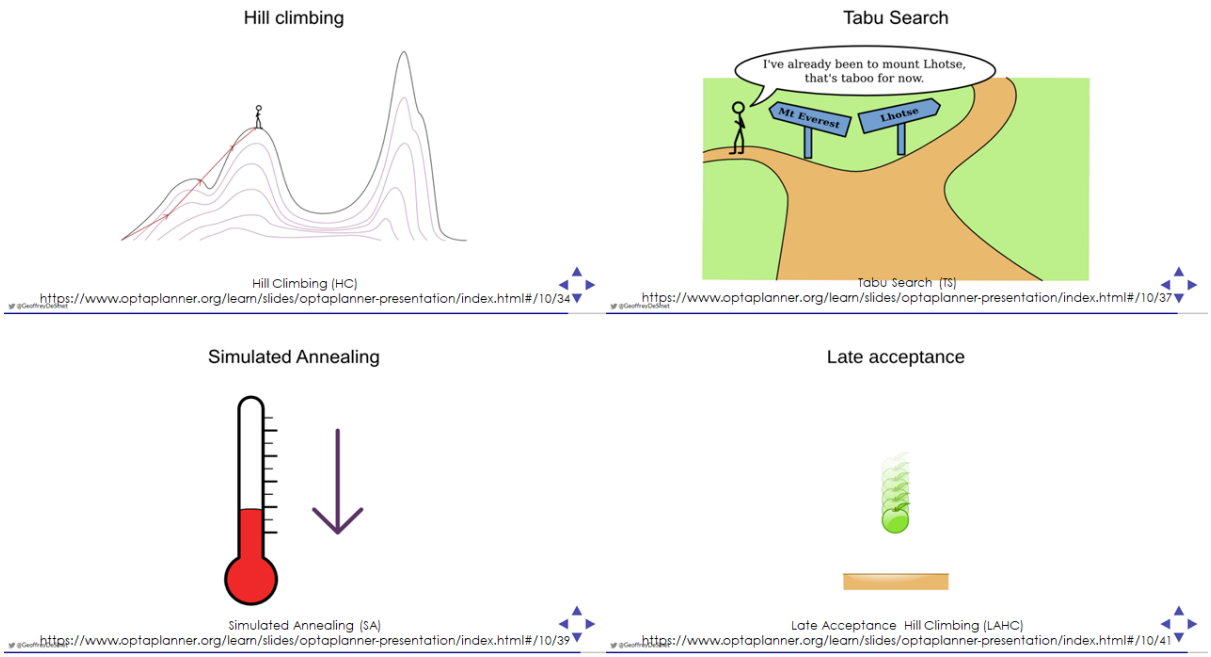
Workshop 2 Guide

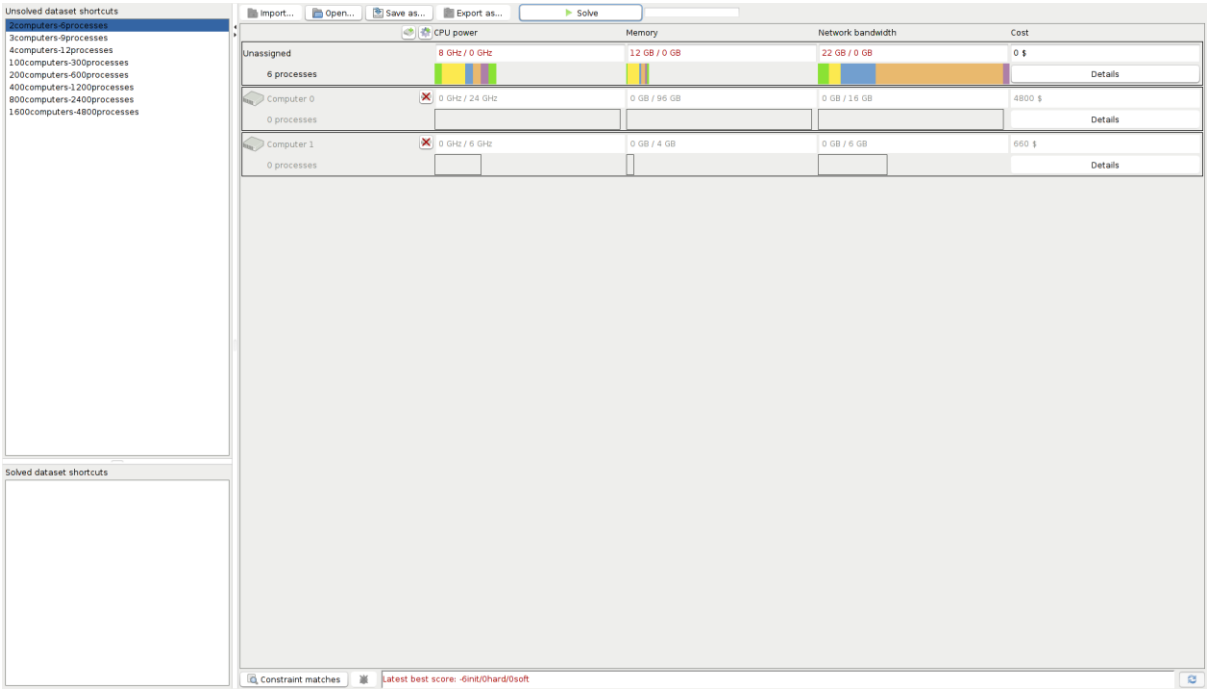
WORKSHOP SEARCH REASONING

- **KIE OptaPlanner Deep Dive**
  - Search Algorithms in Action
  - Cloud Computer Balancing
- **KIE OptaPlanner Development – Individual Work**
  - Choose one OptaPlanner use case, e.g. course curriculum scheduling, airport gate assignment, etc.
  - Analyse, adapt, import, and solve using KIE Workbench or programming.

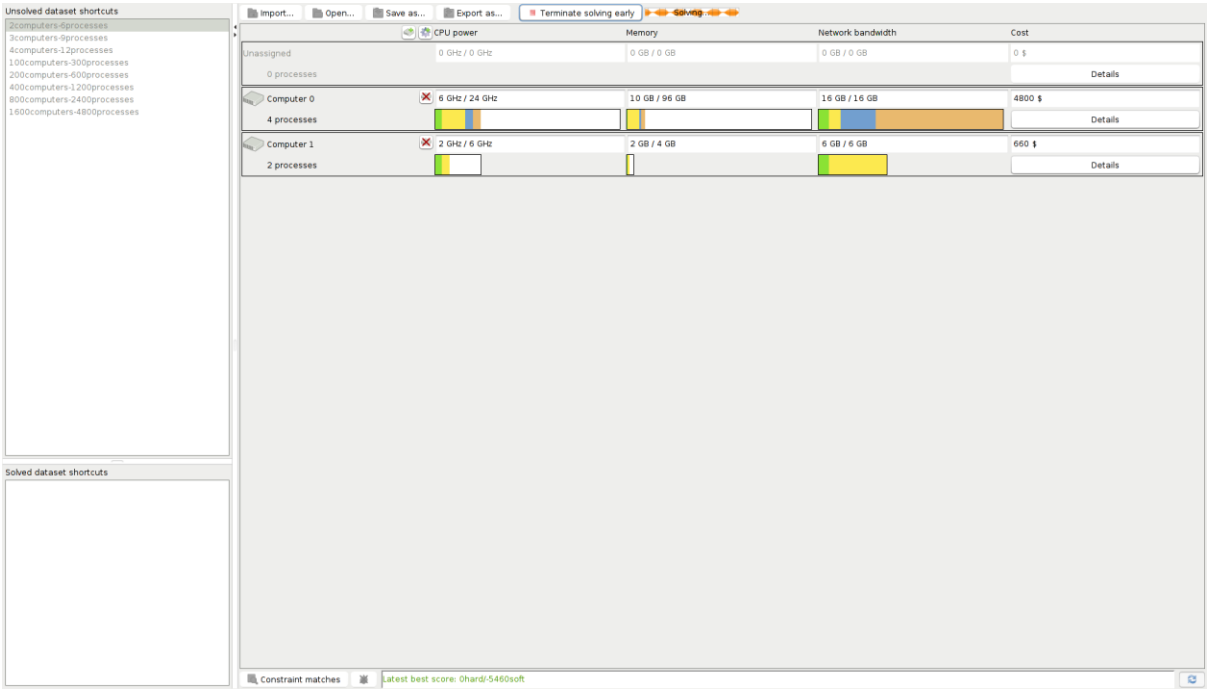


Above: Search Algorithms in Action [Demo]

Cloud Computer Balancing [Demo]



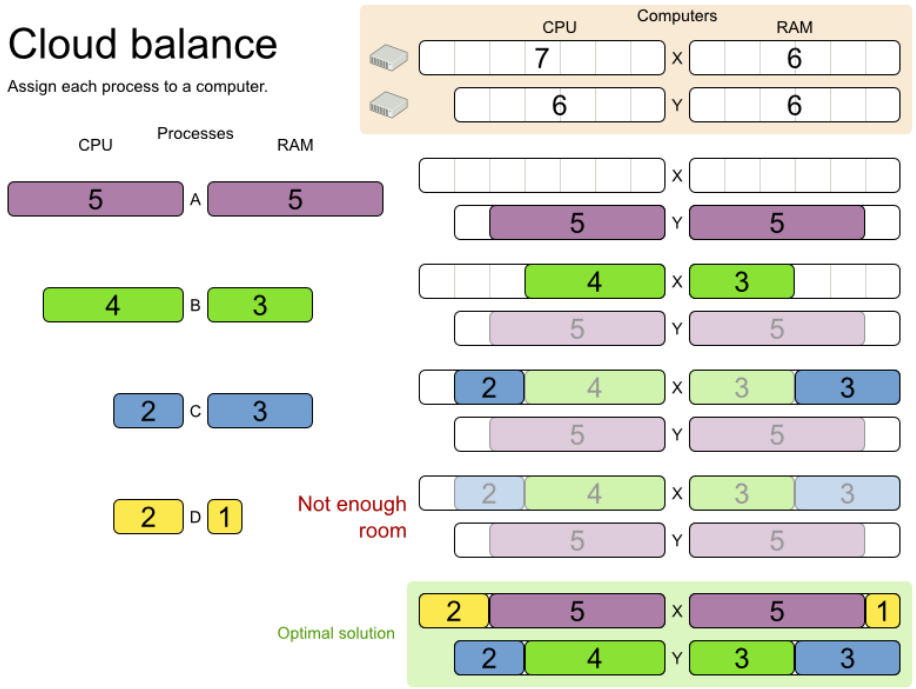
Above: initial solver states

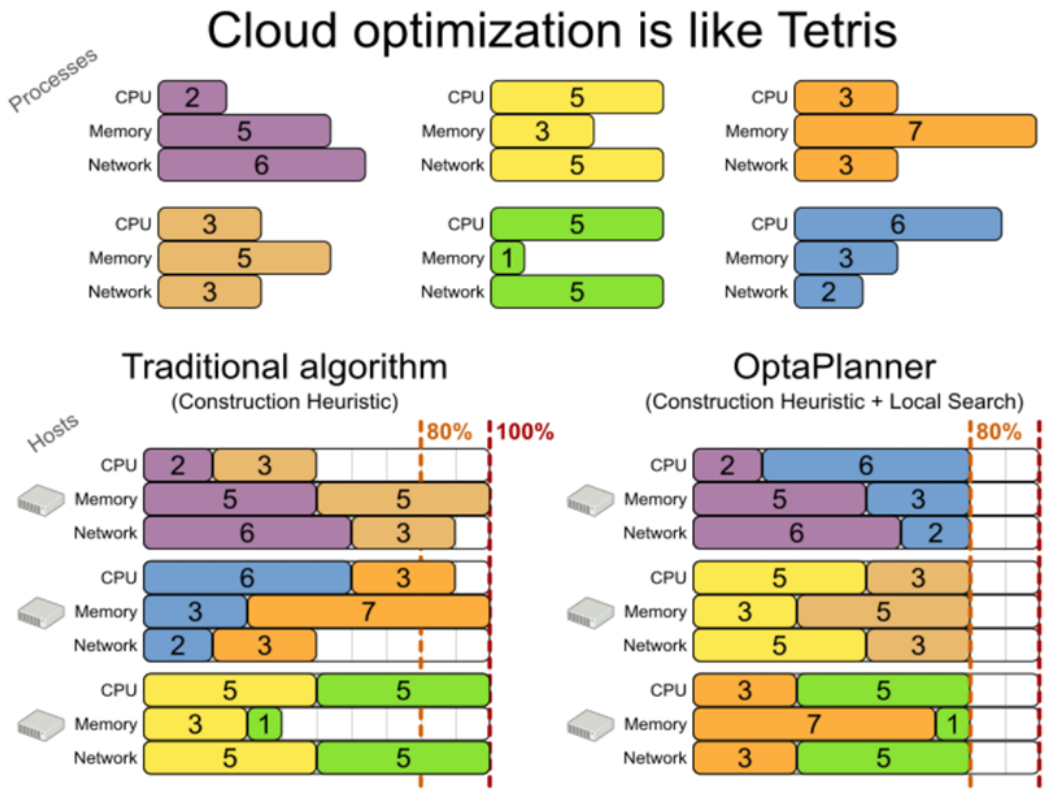


Above: best solution

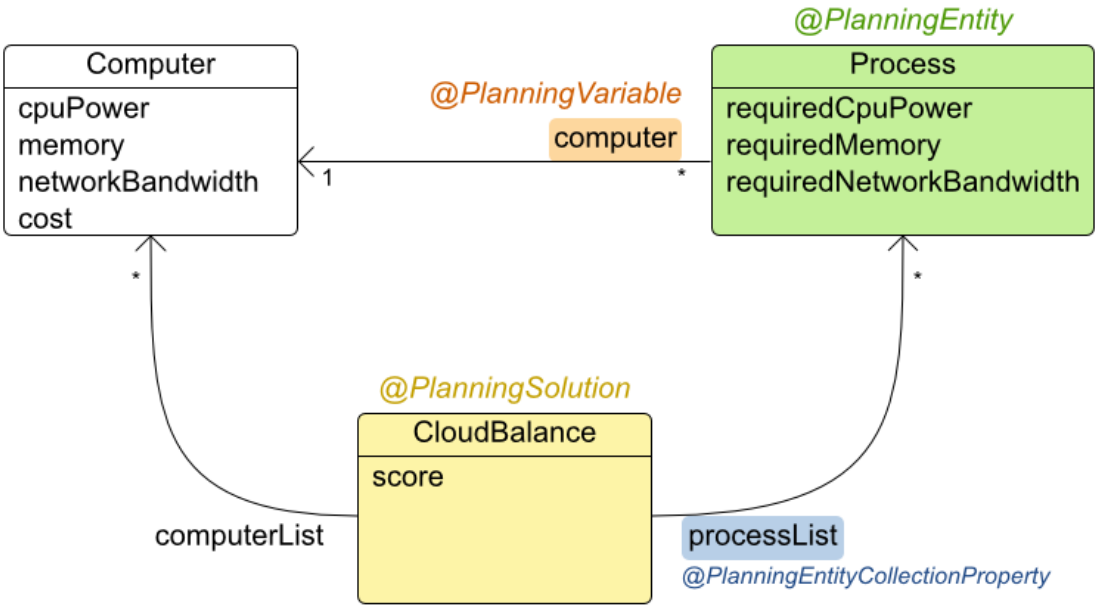
Workshop 2.1 [Individual]

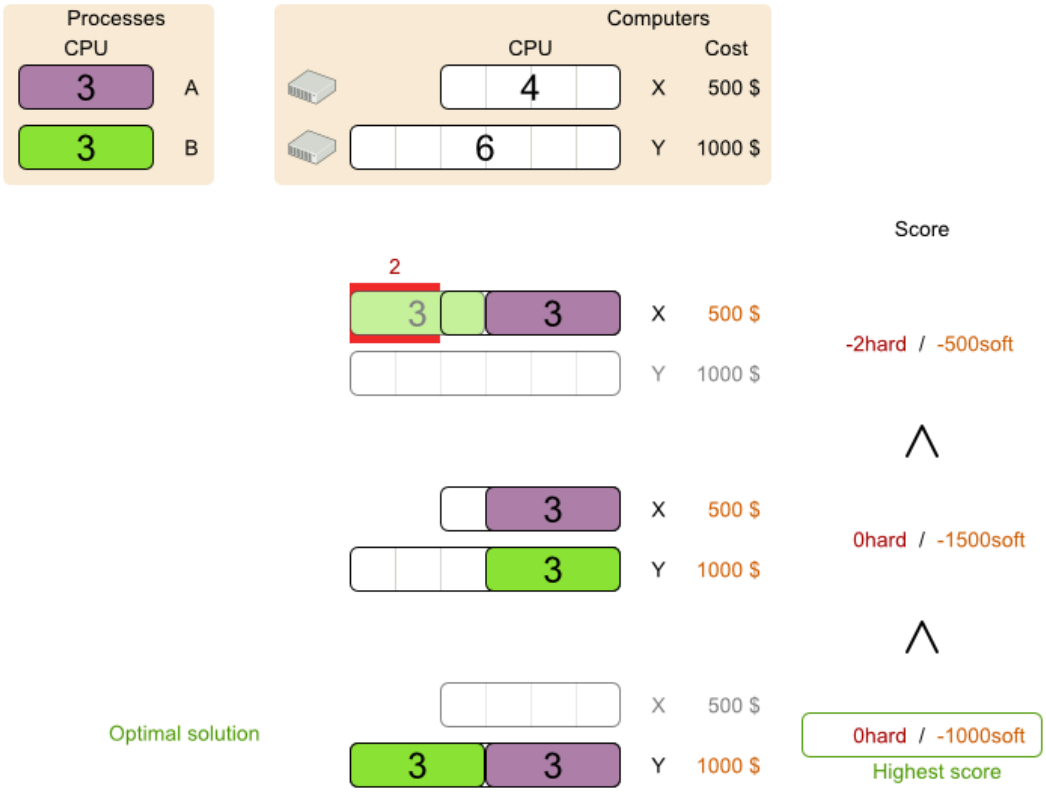
Analyze and execute cloud computer balancing system using both Eclipse and KIE Workbench.



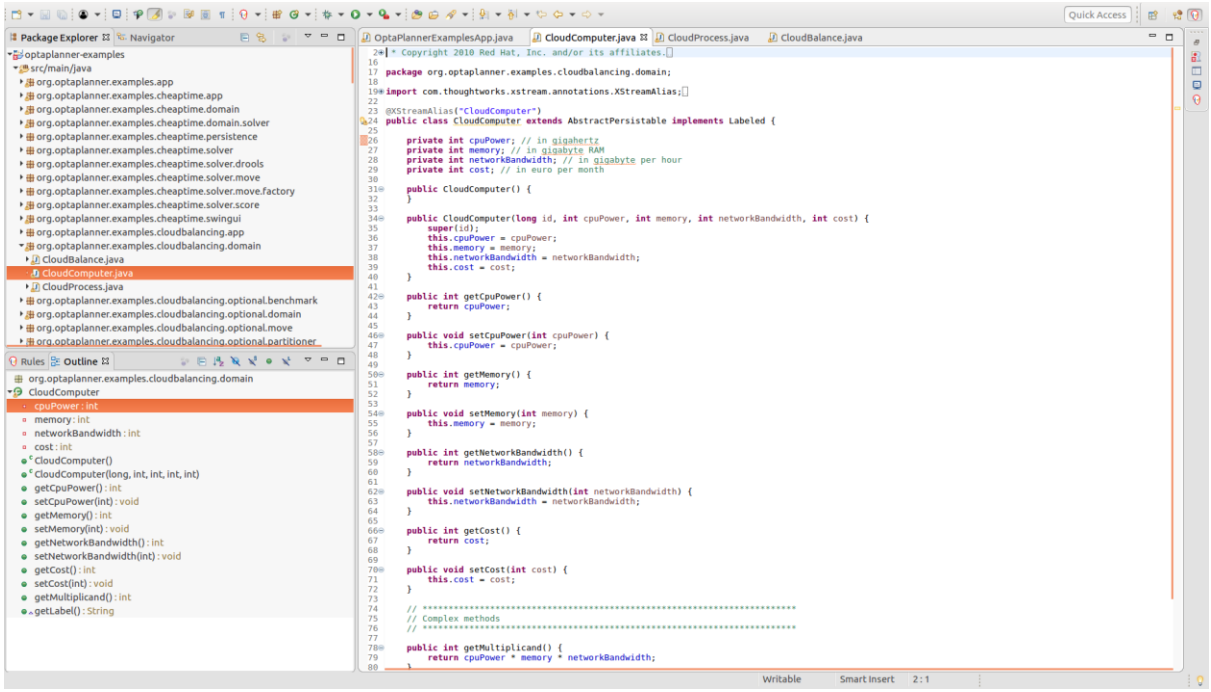


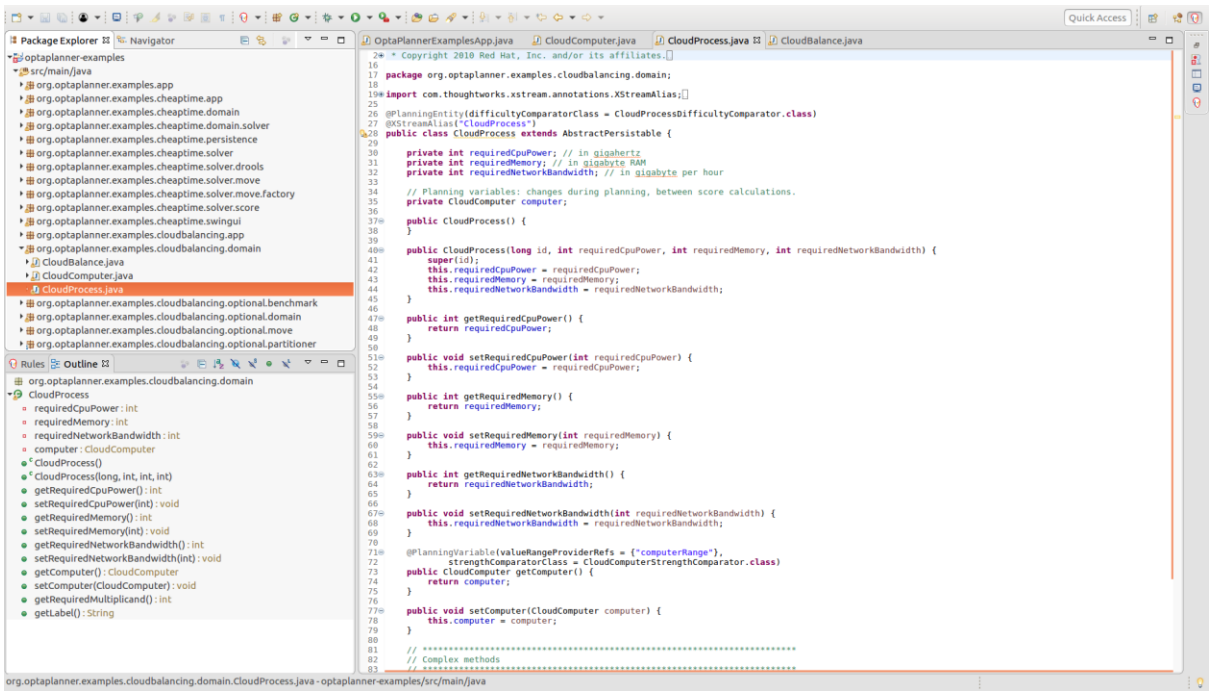
Cloud balance class diagram

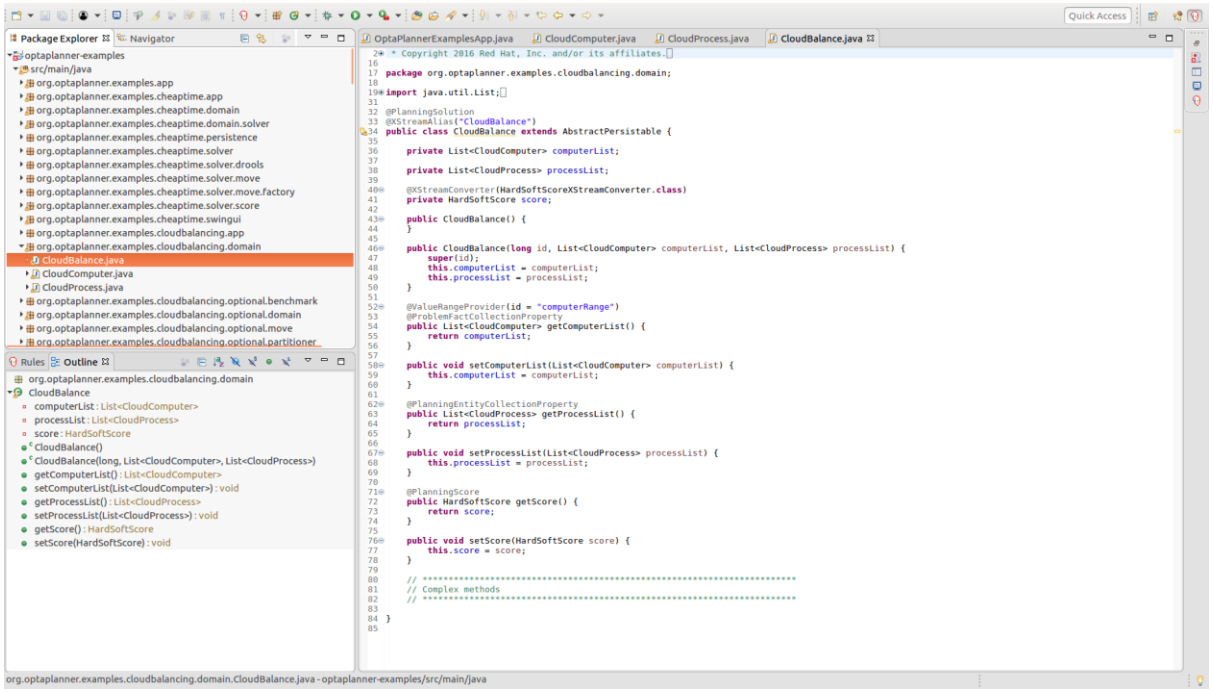


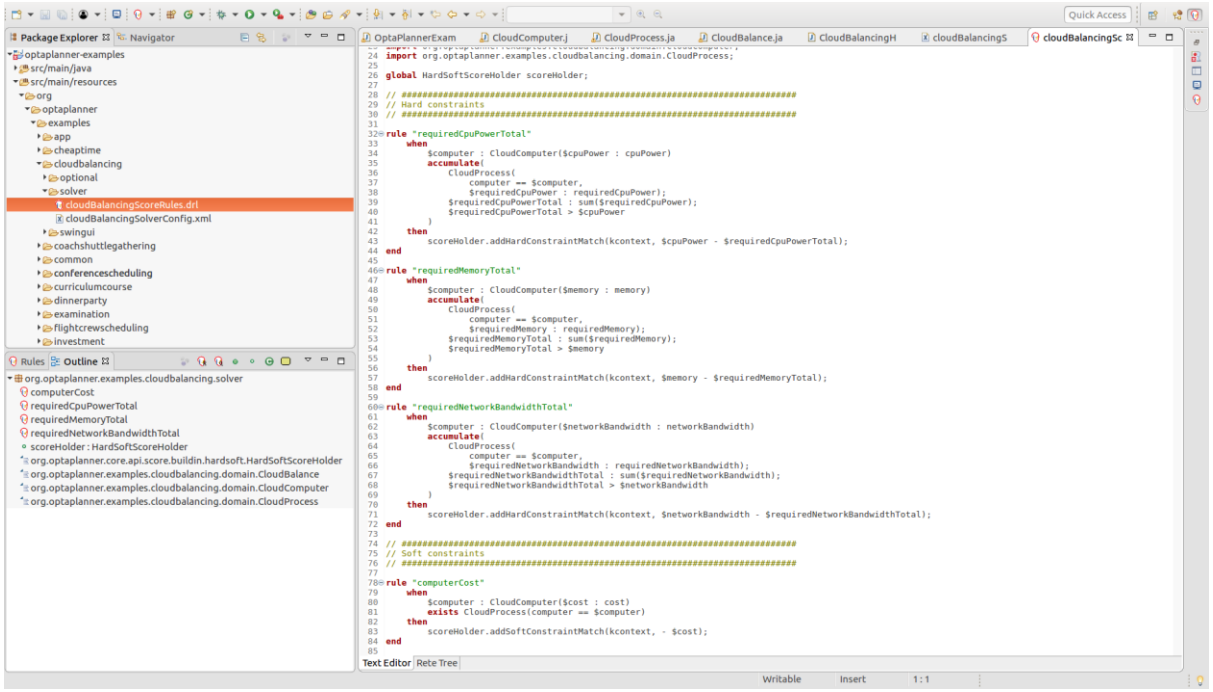


Solver in Java / Eclipse






Package Explorer
Navigator
optaplanner-examples
src/main/java
org.optaplanner.examples.app
org.optaplanner.examples.cheaptime.app
org.optaplanner.examples.cheaptime.domain
org.optaplanner.examples.cheaptime.domain.solver
org.optaplanner.examples.cheaptime.persistence
org.optaplanner.examples.cheaptime.solver
org.optaplanner.examples.cheaptime.solver.drools
org.optaplanner.examples.cheaptime.solver.move
org.optaplanner.examples.cheaptime.solver.move.factory
org.optaplanner.examples.cheaptime.solver.score
org.optaplanner.examples.cheaptime.swingui
org.optaplanner.examples.cloudbalancing.app
org.optaplanner.examples.cloudbalancing.domain
CloudBalance.java
CloudComputer.java
CloudProcess.java
org.optaplanner.examples.cloudbalancing.optional.benchmark
org.optaplanner.examples.cloudbalancing.optional.domain
org.optaplanner.examples.cloudbalancing.optional.move
org.optaplanner.examples.cloudbalancing.optional.partitioner
Rules
Outline
org.optaplanner.examples.cloudbalancing.domain
CloudBalance
computerList : List<CloudComputer>
processList : List<CloudProcess>
score : HardSoftScore
CloudBalance()
CloudBalance(long, List<CloudComputer>, List<CloudProcess>)
getComputerList() : List<CloudComputer>
setComputerList(List<CloudComputer>) : void
getProcessList() : List<CloudProcess>
setProcessList(List<CloudProcess>) : void
getScore() : HardSoftScore
setScore(HardSoftScore) : void
OptaPlannerExamplesApp.java
CloudComputer.java
CloudProcess.java
CloudBalance.java
Quick Access
org.optaplanner.examples.cloudbalancing.domain.CloudBalance.java - optaplanner-examples/src/main/java


Package Explorer
Navigator
optaplanner-examples
src/main/java
src/main/resources
org
optaplanner
examples
app
cheaptime
cloudbalancing
optional
solver
cloudBalancingScoreRules.drl
cloudBalancingSolverConfig.xml
swingui
coachshuttlegathering
common
conferencescheduling
curriculumcourse
dinnerparty
examination
flightcrewscheduling
investment
Rules
Outline
org.optaplanner.examples.cloudbalancing.solver
computerCost
requiredCpuPowerTotal
requiredMemoryTotal
requiredNetworkBandwidthTotal
scoreHolder : HardSoftScoreHolder
org.optaplanner.core.api.score.buildin.hardsoft.HardSoftScoreHolder
org.optaplanner.examples.cloudbalancing.domain.CloudBalance
org.optaplanner.examples.cloudbalancing.domain.CloudComputer
org.optaplanner.examples.cloudbalancing.domain.CloudProcess
OptaPlannerExam
CloudComputer.j
CloudProcess.ja
CloudBalance.ja
CloudBalancingH
cloudBalancingS
cloudBalancingSc
Quick Access
Text Editor
Rete Tree
Writable
Insert
1 : 1

2.5.1. Easy Java Score Configuration

One way to define a score function is to implement the interface `EasyScoreCalculator` in plain Java.

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

Just implement the `calculateScore(Solution)` method to return a `HardSoftScore` instance.

Example 6. CloudBalancingEasyScoreCalculator.java

```
public class CloudBalancingEasyScoreCalculator implements EasyScoreCalculator<CloudBalance> {

    /**
     * A very simple implementation. The double loop can easily be removed by using Maps as sho
     * {@link CloudBalancingMapBasedEasyScoreCalculator#calculateScore(CloudBalance)}.
     */
    public HardSoftScore calculateScore(CloudBalance cloudBalance) {
        int hardScore = 0;
        int softScore = 0;
        for (CloudComputer computer : cloudBalance.getComputerList()) {
            int cpuPowerUsage = 0;
            int memoryUsage = 0;
            int networkBandwidthUsage = 0;
            boolean used = false;

            // Calculate usage
            for (CloudProcess process : cloudBalance.getProcessList()) {
                if (computer.equals(process.getComputer())) {
                    cpuPowerUsage += process.getRequiredCpuPower();
                    memoryUsage += process.getRequiredMemory();
                    networkBandwidthUsage += process.getRequiredNetworkBandwidth();
                    used = true;
                }
            }

            // Hard constraints
            int cpuPowerAvailable = computer.getCpuPower() - cpuPowerUsage;
            if (cpuPowerAvailable < 0) {
                hardScore += cpuPowerAvailable;
            }
            int memoryAvailable = computer.getMemory() - memoryUsage;
            if (memoryAvailable < 0) {
                hardScore += memoryAvailable;
            }
            int networkBandwidthAvailable = computer.getNetworkBandwidth() - networkBandwidthUs
            if (networkBandwidthAvailable < 0) {
                hardScore += networkBandwidthAvailable;
            }

            // Soft constraints
            if (used) {
                softScore -= computer.getCost();
            }
        }
        return HardSoftScore.valueOf(hardScore, softScore);
    }

}
```

Even if we optimize the code above to use `Map`s to iterate through the `processList` only once, *it is still slow* because it does not do incremental score calculation. To fix that, either use incremental Java score calculation or Drools score calculation.

2.5.2. Drools Score Configuration

Drools score calculation uses incremental calculation, where every score constraint is written as one or more score rules.

Using the Drools rule engine for score calculation, allows you to integrate with other Drools technologies, such as decision tables (XLS or web based), the KIE Workbench, ...

Prerequisite To use the Drools rule engine as a score function, simply add a `scoreDrl` resource in the classpath:

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

1. We want to make sure that all computers have enough CPU, RAM and network bandwidth to support all their processes, so we make these hard constraints:

Example 7. cloudBalancingScoreRules.drl - Hard Constraints

```
...

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

global HardSoftScoreHolder scoreHolder;

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

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

rule "requiredMemoryTotal"
    ...
end

rule "requiredNetworkBandwidthTotal"
    ...
end
```

2. If those constraints are met, we want to minimize the maintenance cost, so we add that as a soft constraint:

Example 8. cloudBalancingScoreRules.drl - Soft Constraints

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

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

WORKSHOP SEARCH REASONING
KIE OptaPlanner Deep Dive – Cloud Computer Balancing

Example 4. CloudBalancingHelloWorld.java

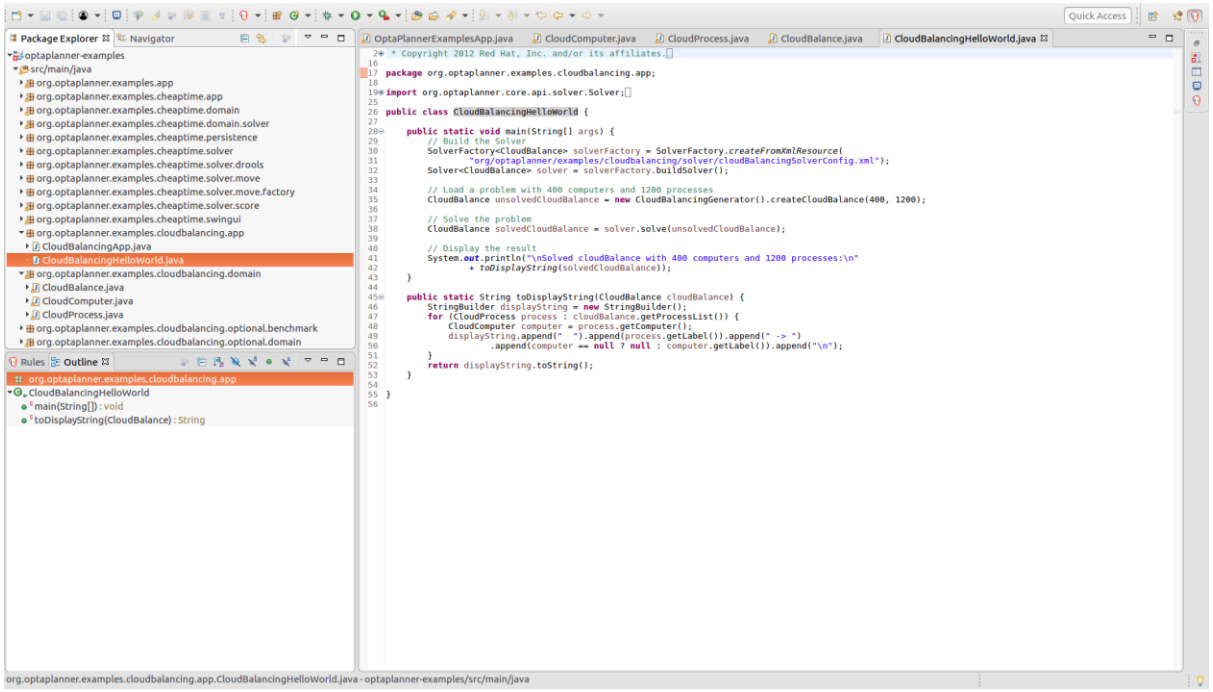
```
public class CloudBalancingHelloWorld {

    public static void main(String[] args) {
        // Build the Solver
        SolverFactory<CloudBalance> solverFactory = SolverFactory.createFromXmlResource(
                "org/optaplanner/examples/cloudbalancing/solver/cloudBalancingSolverConfig.xml"
        Solver<CloudBalance> solver = solverFactory.buildSolver();

        // Load a problem with 400 computers and 1200 processes
        CloudBalance unsolvedCloudBalance = new CloudBalancingGenerator().createCloudBalance(40

        // Solve the problem
        CloudBalance solvedCloudBalance = solver.solve(unsolvedCloudBalance);

        // Display the result
        System.out.println("\nSolved cloudBalance with 400 computers and 1200 processes:\n"
                + toDisplayString(solvedCloudBalance));
    }

    ...
}
```

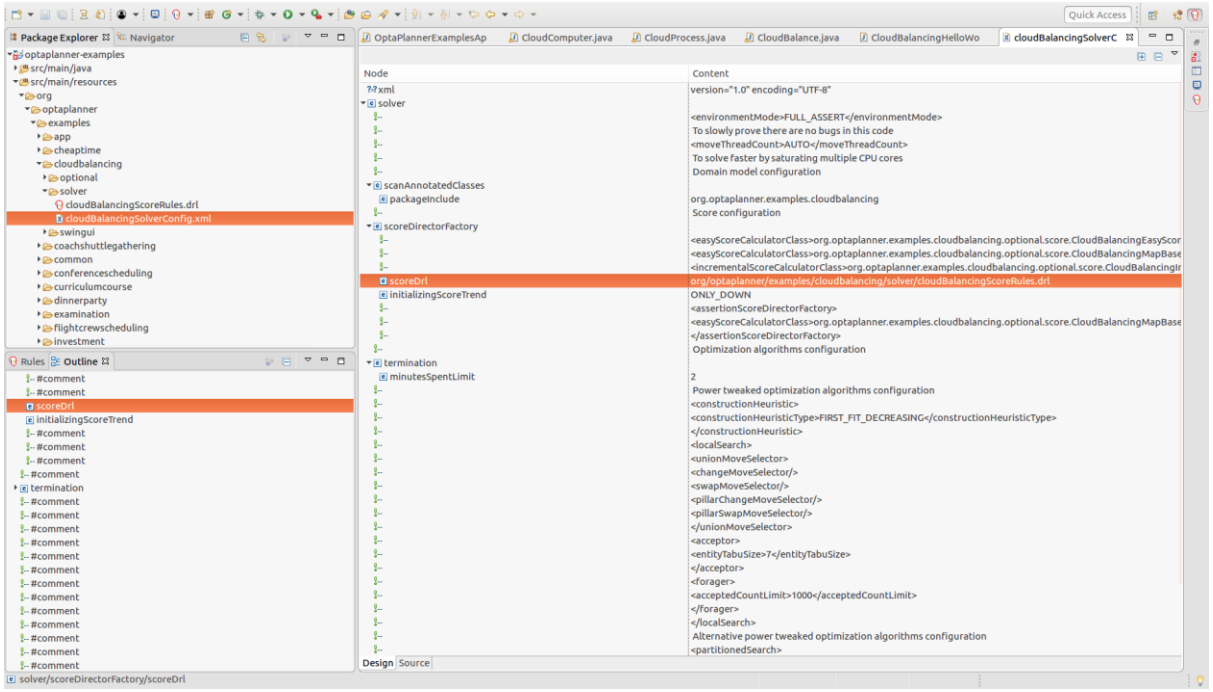
WORKSHOP SEARCH REASONING
KIE OptaPlanner Deep Dive – Cloud Computer Balancing

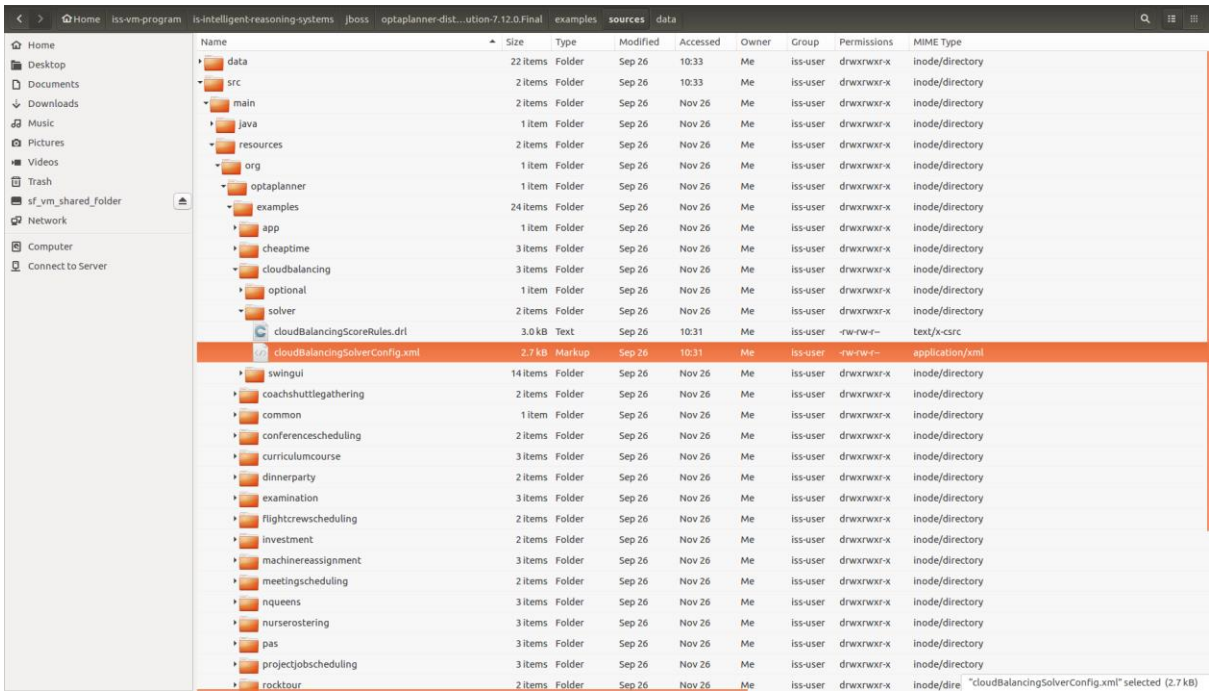
The solver configuration file determines how the solving process works; it is considered a part of the code. The file is named `cloudBalancingSolverConfig.xml`.

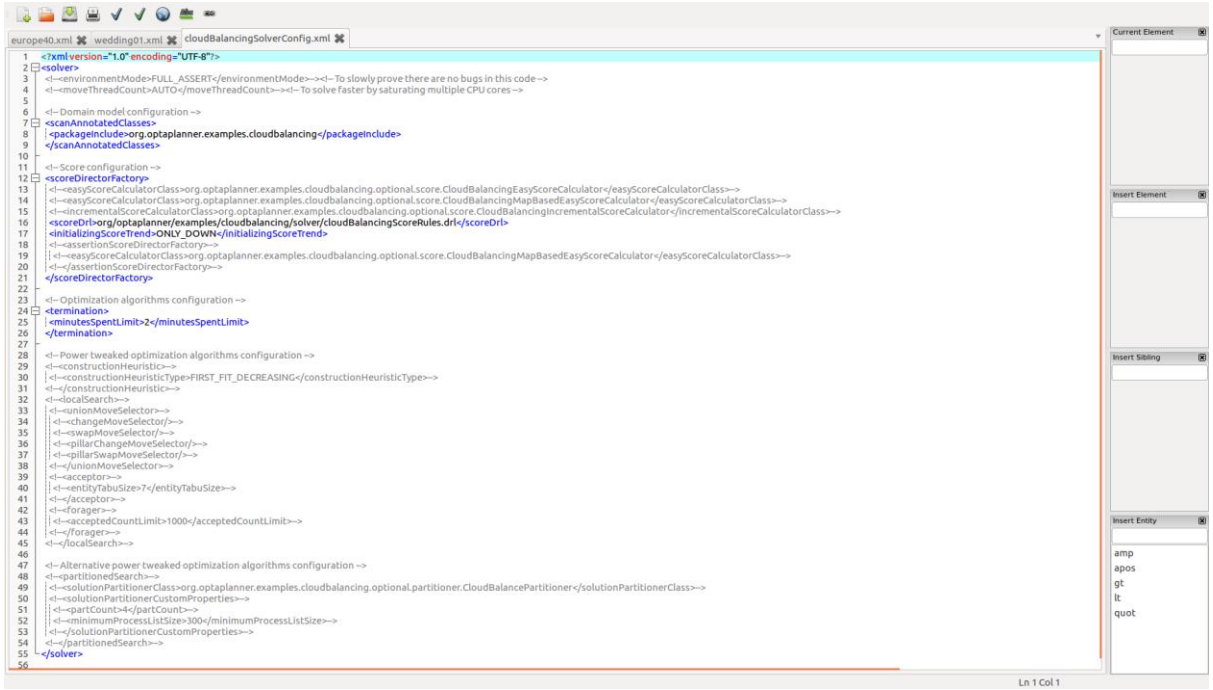
Example 5. cloudBalancingSolverConfig.xml

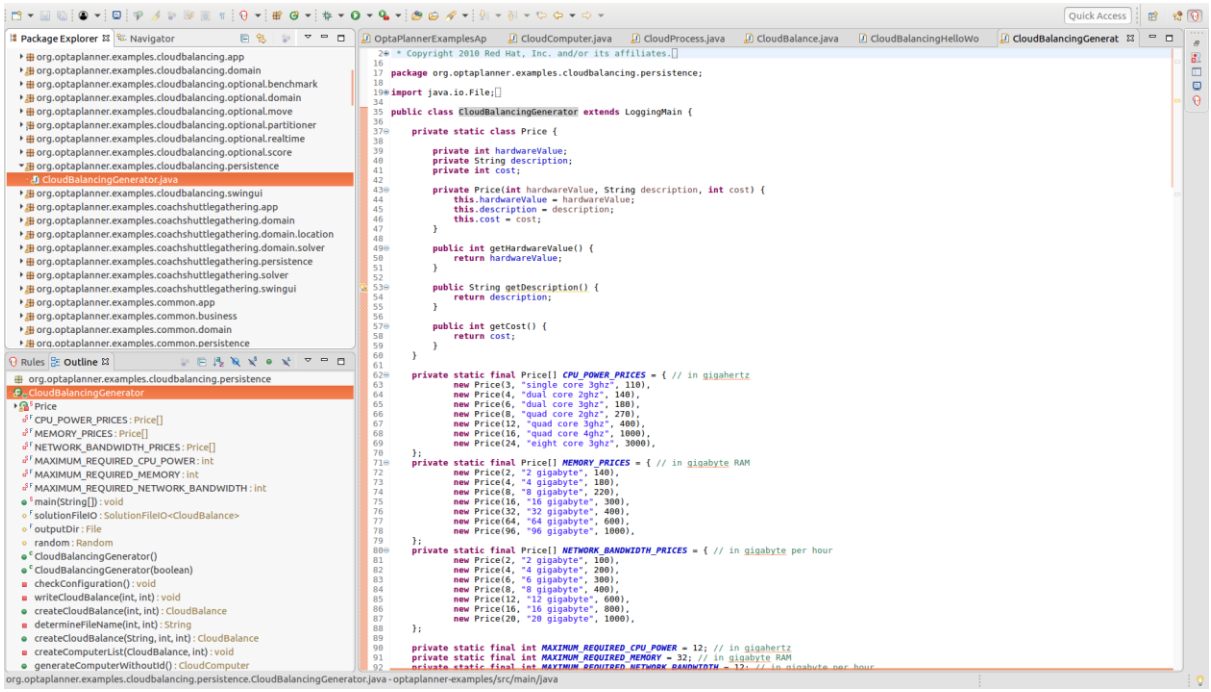
```
<?xml version="1.0" encoding="UTF-8"?>
<solver>
  <!-- Domain model configuration -->
  <scanAnnotatedClasses/>

  <!-- Score configuration -->
  <scoreDirectorFactory>
    <easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalan
    <!--<scoreDrl>org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl</
  </scoreDirectorFactory>

  <!-- Optimization algorithms configuration -->
  <termination>
    <secondsSpentLimit>30</secondsSpentLimit>
  </termination>
</solver>
```

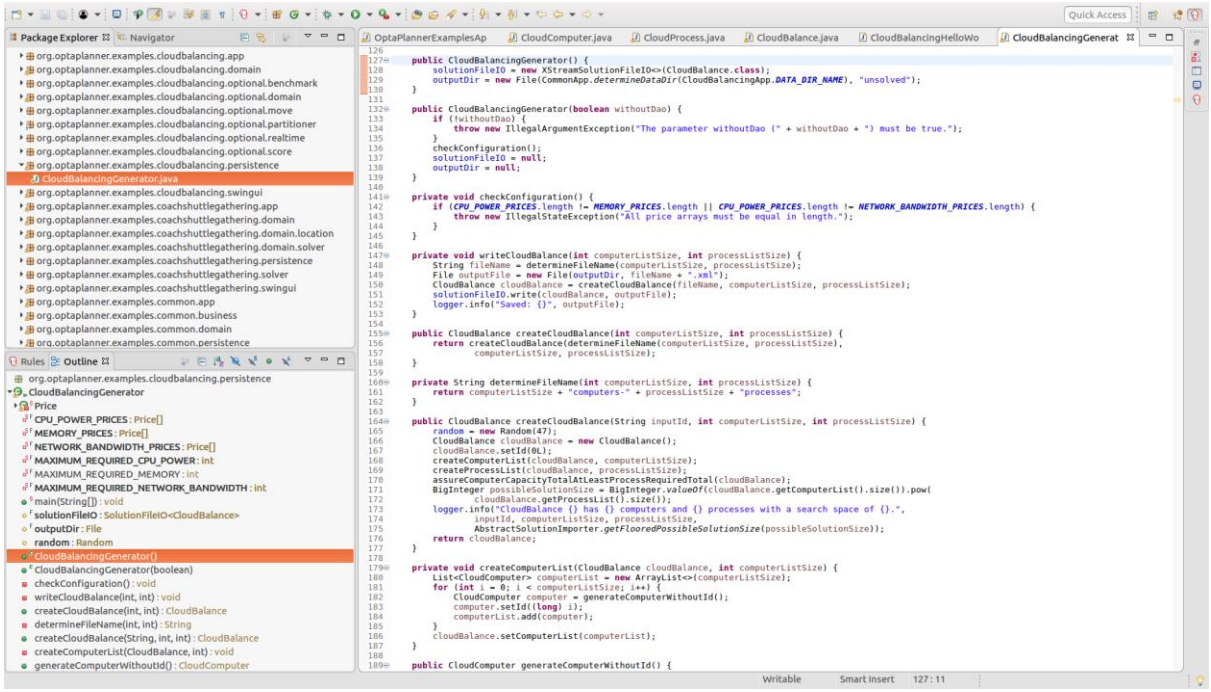

Package Explorer
Navigator
optaplanner-examples
src/main/java
org.optaplanner.examples.app
org.optaplanner.examples.cheaptime.app
org.optaplanner.examples.cheaptime.domain
org.optaplanner.examples.cheaptime.domain.solver
org.optaplanner.examples.cheaptime.persistence
org.optaplanner.examples.cheaptime.solver
org.optaplanner.examples.cheaptime.solver.drools
org.optaplanner.examples.cheaptime.solver.move
org.optaplanner.examples.cheaptime.solver.move.factory
org.optaplanner.examples.cheaptime.solver.score
org.optaplanner.examples.cheaptime.swingui
org.optaplanner.examples.cloudbalancing.app
CloudBalancingApp.java
CloudBalancingHelloWorld.java
org.optaplanner.examples.cloudbalancing.domain
CloudBalance.java
CloudComputer.java
CloudProcess.java
org.optaplanner.examples.cloudbalancing.optional.benchmark
org.optaplanner.examples.cloudbalancing.optional.domain
Rules
Outline
org.optaplanner.examples.cloudbalancing.app
CloudBalancingHelloWorld
main(String[]) : void
toDisplayString(CloudBalance) : String
OptaPlannerExamplesApp.java
CloudComputer.java
CloudProcess.java
CloudBalance.java
CloudBalancingHelloWorld.java
org.optaplanner.examples.cloudbalancing.app.CloudBalancingHelloWorld.java - optaplanner-examples/src/main/java

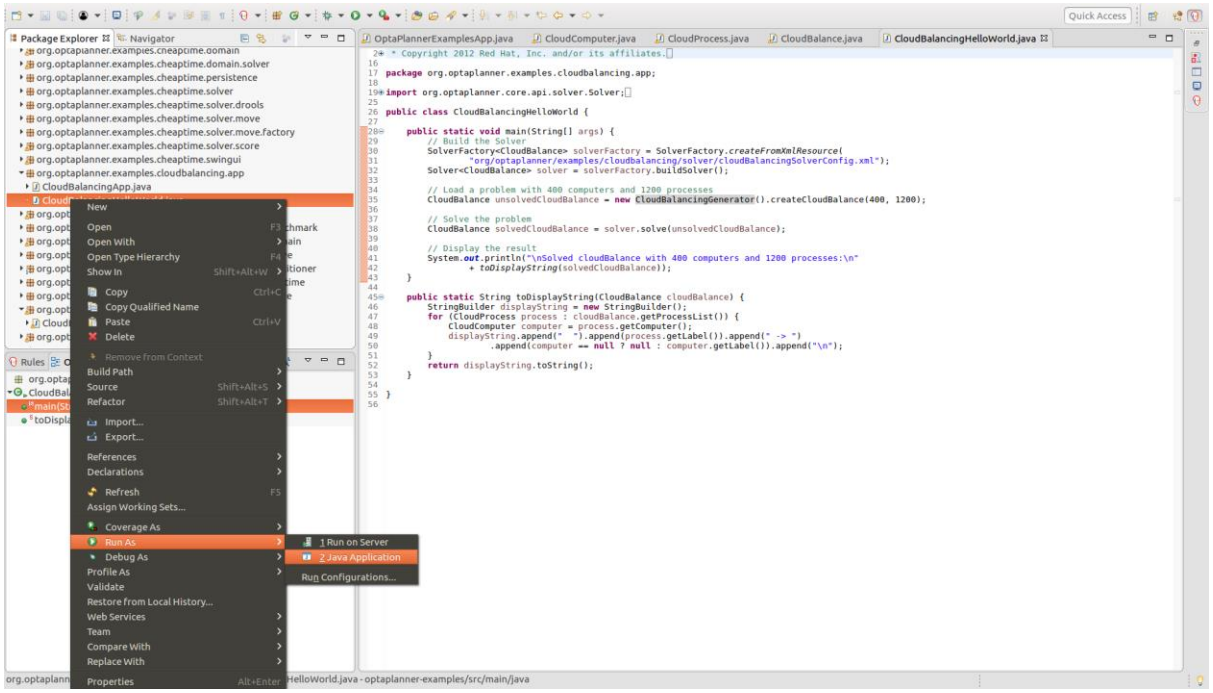

Package Explorer
Navigator
optaplanner-examples
src/main/java
src/main/resources
org
optaplanner
examples
app
cheaptime
cloudbalancing
optional
solver
cloudBalancingScoreRules.drl
cloudBalancingSolverConfig.xml
swingui
coachshuttlegathering
common
conferencescheduling
curriculumcourse
dinnerparty
examination
flightcrewscheduling
investment
Rules
Outline
#comment
scoreDrl
initializingScoreTrend
termination
minutesSpentLimit
Node
Content
xml
solver
scanAnnotatedClasses
packageInclude
scoreDirectorFactory
scoreDrl
initializingScoreTrend
termination
minutesSpentLimit
Design
Source
solver/scoreDirectorFactory/scoreDrl

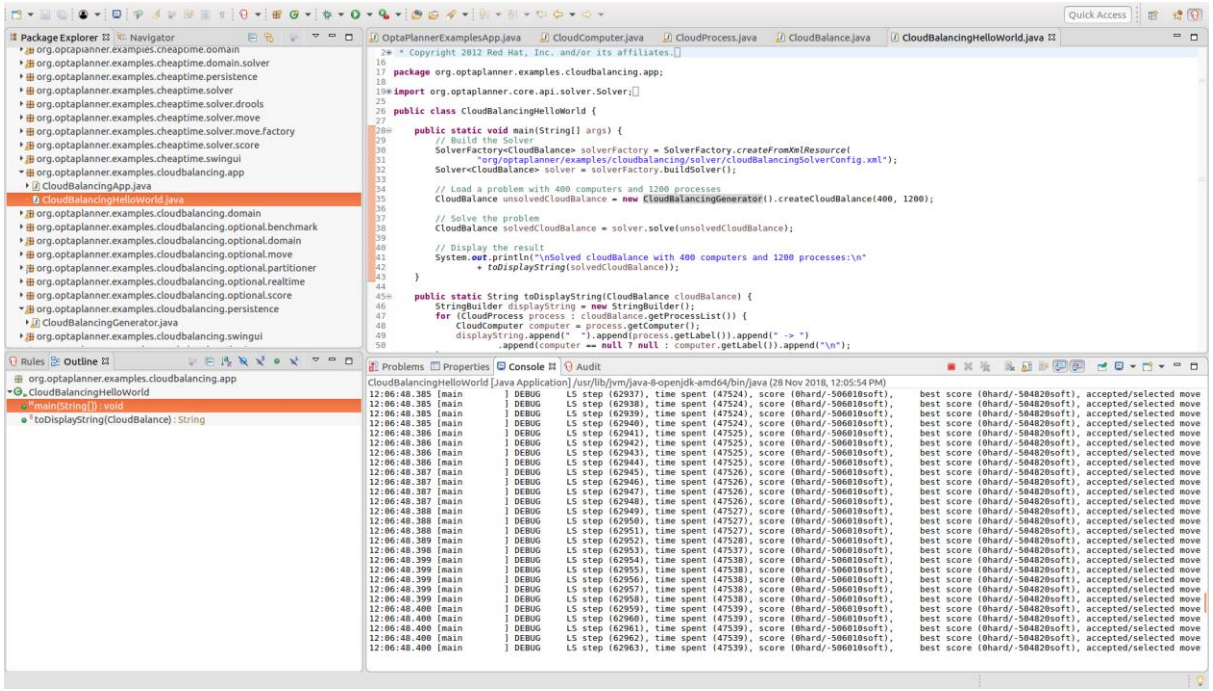

Home
iss-vm-program
is-intelligent-reasoning-systems
jboss
optaplanner-dist...ution-7.12.0.Final
examples
sources
data
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
sf_vm_shared_folder
Network
Computer
Connect to Server
Name
Size
Type
Modified
Accessed
Owner
Group
Permissions
MIME Type
data
src
main
java
resources
org
optaplanner
examples
app
cheaptime
cloudbalancing
optional
solver
cloudBalancingScoreRules.drl
cloudBalancingSolverConfig.xml
swingui
coachshuttlegathering
common
conferencescheduling
curriculumcourse
dinnerparty
examination
flightcrewscheduling
investment
machinereassignment
meetingscheduling
nqueens
nurserostering
pas
projectjobscheduling
rocktour
"cloudBalancingSolverConfig.xml" selected (2.7 kB)

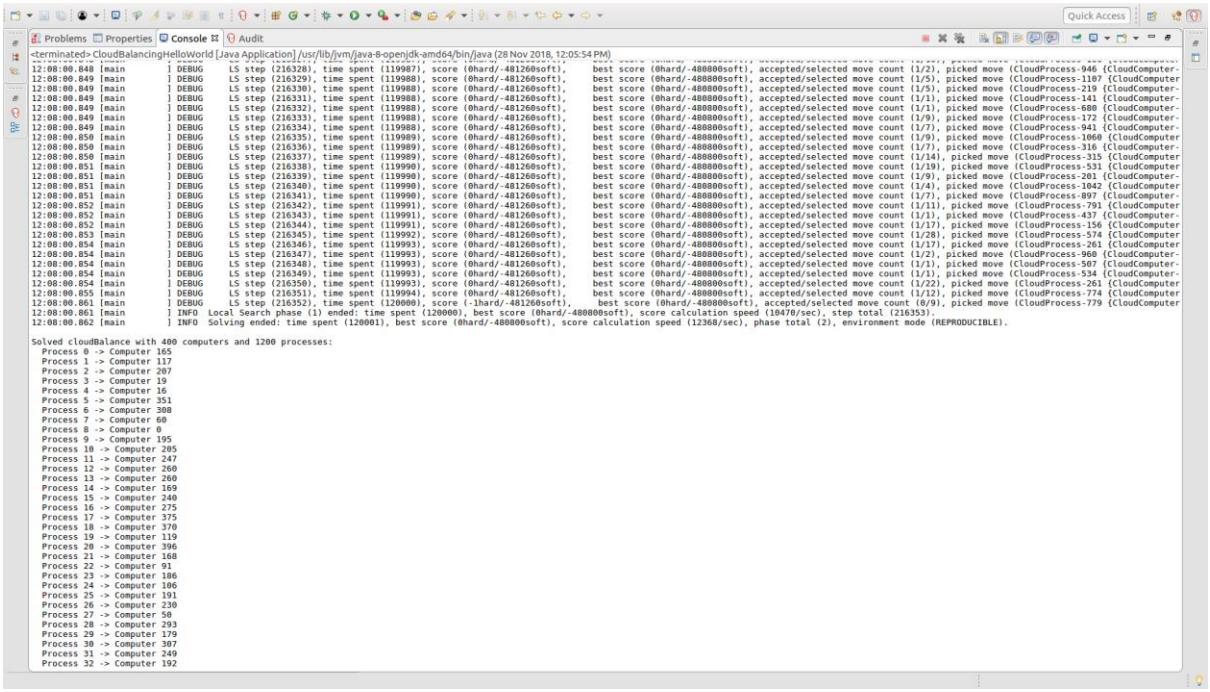

europe40.xml
wedding01.xml
cloudBalancingSolverConfig.xml
Current Element
Insert Element
Insert Sibling
Insert Entity
amp
apos
gt
lt
quot
Ln 1 Col 1



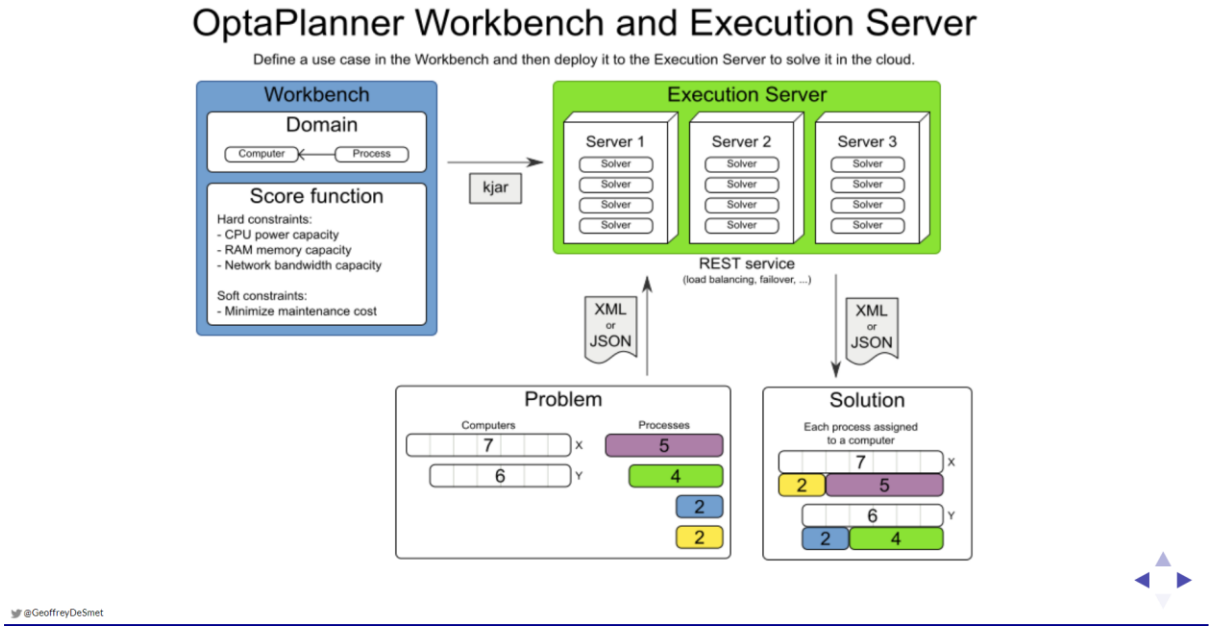


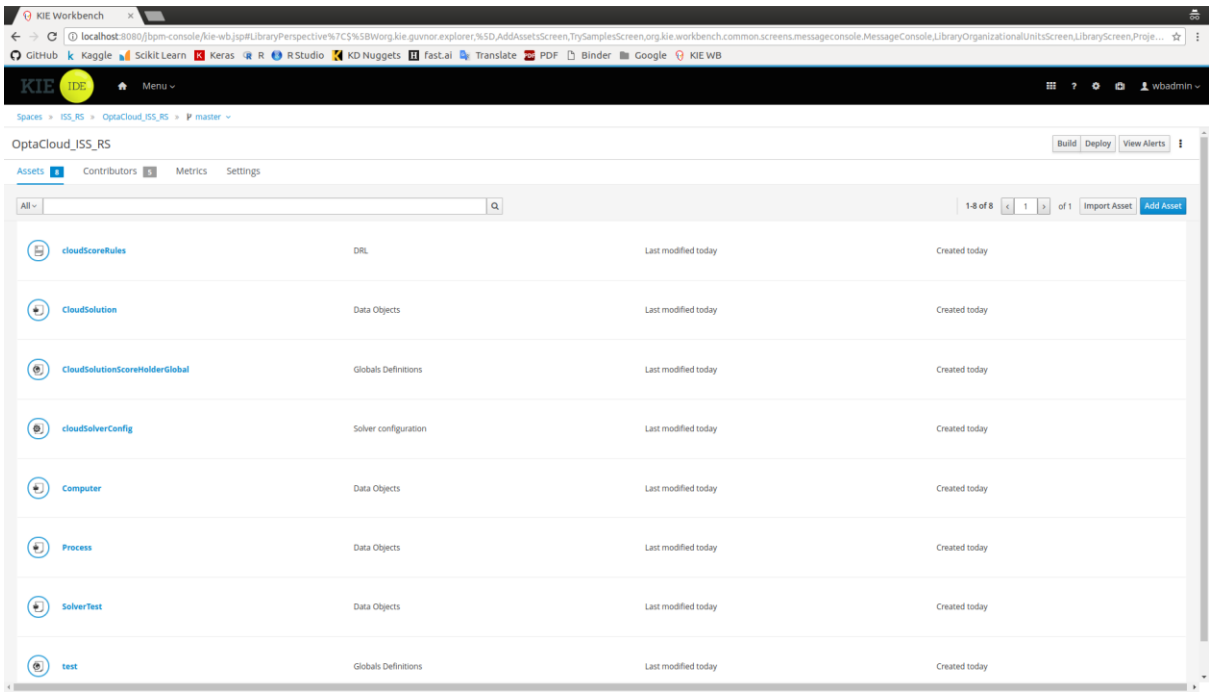

Package Explorer
Navigator
OptaPlannerExamplesApp.java
CloudComputer.java
CloudProcess.java
CloudBalance.java
CloudBalancingHelloWorld.java
CloudBalancingApp.java
New
Open
Open With
Open Type Hierarchy
Show In
Copy
Copy Qualified Name
Paste
Delete
Remove From Context
Build Path
Source
Refactor
Import...
Export...
References
Declarations
Refresh
Assign Working Sets...
Coverage As
Run As
Debug As
Profile As
Validate
Restore from Local History...
Web Services
Team
Compare With
Replace With
Properties
1 Run on Server
2 Java Application
Run Configurations...


Package Explorer
Navigator
Outline
Problems
Properties
Console
Audit
CloudBalancingHelloWorld [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (28 Nov 2018, 12:05:54 PM)

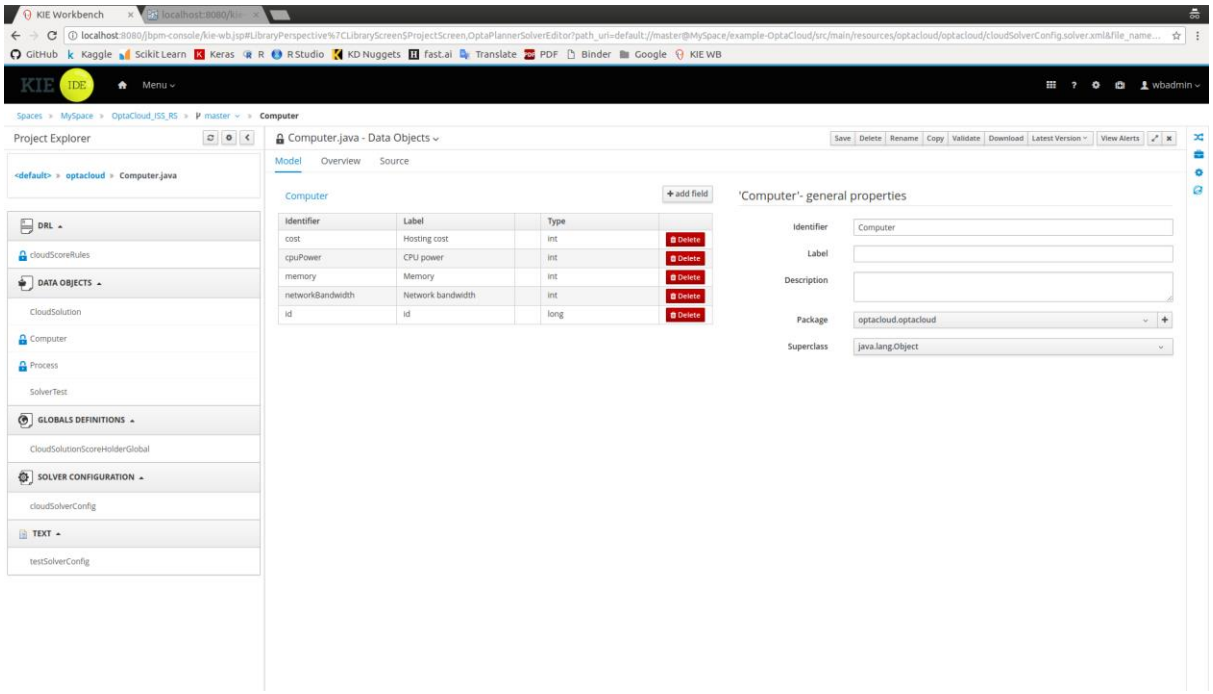

Problems
Properties
Console
Audit
<terminated> CloudBalancingHelloWorld [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (28 Nov 2018, 12:05:54 PM)
12:08:00.848 [main] DEBUG LS step (216328), time spent (119987), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/2), picked move (CloudProcess-946 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216329), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/5), picked move (CloudProcess-1107 {CloudComputer
12:08:00.849 [main] DEBUG LS step (216330), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/5), picked move (CloudProcess-219 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216331), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-141 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216332), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-680 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216333), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/9), picked move (CloudProcess-172 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216334), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/7), picked move (CloudProcess-941 {CloudComputer-
12:08:00.850 [main] DEBUG LS step (216335), time spent (119989), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/9), picked move (CloudProcess-1060 {CloudComputer
12:08:00.850 [main] DEBUG LS step (216336), time spent (119989), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/7), picked move (CloudProcess-316 {CloudComputer-
12:08:00.850 [main] DEBUG LS step (216337), time spent (119989), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/14), picked move (CloudProcess-315 {CloudComputer
12:08:00.851 [main] DEBUG LS step (216338), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/19), picked move (CloudProcess-531 {CloudComputer
12:08:00.851 [main] DEBUG LS step (216339), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/9), picked move (CloudProcess-201 {CloudComputer-
12:08:00.851 [main] DEBUG LS step (216340), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/4), picked move (CloudProcess-1042 {CloudComputer
12:08:00.851 [main] DEBUG LS step (216341), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/7), picked move (CloudProcess-897 {CloudComputer-
12:08:00.852 [main] DEBUG LS step (216342), time spent (119991), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/11), picked move (CloudProcess-791 {CloudComputer
12:08:00.852 [main] DEBUG LS step (216343), time spent (119991), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-437 {CloudComputer-
12:08:00.852 [main] DEBUG LS step (216344), time spent (119991), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/17), picked move (CloudProcess-156 {CloudComputer
12:08:00.853 [main] DEBUG LS step (216345), time spent (119992), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/28), picked move (CloudProcess-574 {CloudComputer
12:08:00.854 [main] DEBUG LS step (216346), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/17), picked move (CloudProcess-261 {CloudComputer
12:08:00.854 [main] DEBUG LS step (216347), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/2), picked move (CloudProcess-960 {CloudComputer-
12:08:00.854 [main] DEBUG LS step (216348), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-507 {CloudComputer-
12:08:00.854 [main] DEBUG LS step (216349), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-534 {CloudComputer-
12:08:00.854 [main] DEBUG LS step (216350), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/22), picked move (CloudProcess-261 {CloudComputer
12:08:00.855 [main] DEBUG LS step (216351), time spent (119994), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/12), picked move (CloudProcess-774 {CloudComputer
12:08:00.861 [main] DEBUG LS step (216352), time spent (120000), score (-1hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (0/9), picked move (CloudProcess-779 {CloudComputer
12:08:00.861 [main] INFO Local Search phase (1) ended: time spent (120000), best score (0hard/-480800soft), score calculation speed (10470/sec), step total (216353).
12:08:00.862 [main] INFO Solving ended: time spent (120001), best score (0hard/-480800soft), score calculation speed (12368/sec), phase total (2), environment mode (REPRODUCIBLE).
Solved cloudBalance with 400 computers and 1200 processes:
Process 0 -> Computer 165
Process 1 -> Computer 117
Process 2 -> Computer 207
Process 3 -> Computer 19
Process 4 -> Computer 16
Process 5 -> Computer 351
Process 6 -> Computer 308
Process 7 -> Computer 60
Process 8 -> Computer 0
Process 9 -> Computer 195
Process 10 -> Computer 205
Process 11 -> Computer 247
Process 12 -> Computer 260
Process 13 -> Computer 260
Process 14 -> Computer 169
Process 15 -> Computer 240
Process 16 -> Computer 275
Process 17 -> Computer 375
Process 18 -> Computer 370
Process 19 -> Computer 119
Process 20 -> Computer 396
Process 21 -> Computer 168
Process 22 -> Computer 91
Process 23 -> Computer 186
Process 24 -> Computer 106
Process 25 -> Computer 191
Process 26 -> Computer 230
Process 27 -> Computer 50
Process 28 -> Computer 293
Process 29 -> Computer 179
Process 30 -> Computer 307
Process 31 -> Computer 249
Process 32 -> Computer 192

Solver in KIE Workbench & Server



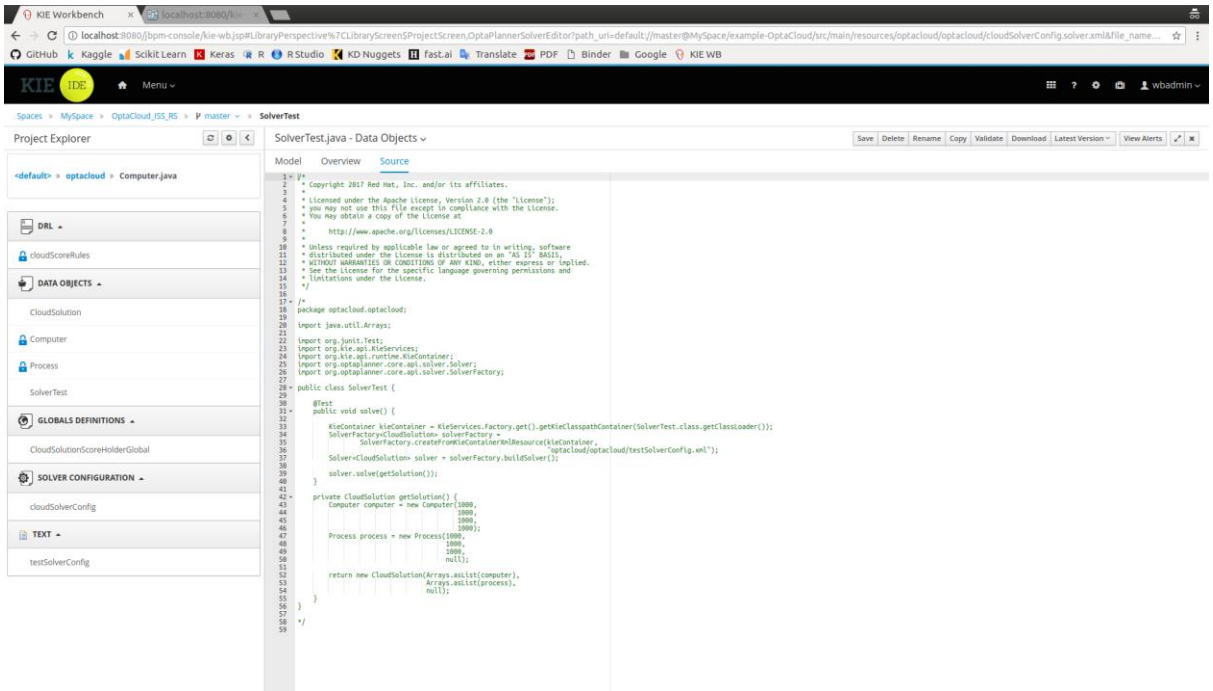


Above: try-sample project *OptaCloud_ISS_RS* in KIE Workbench



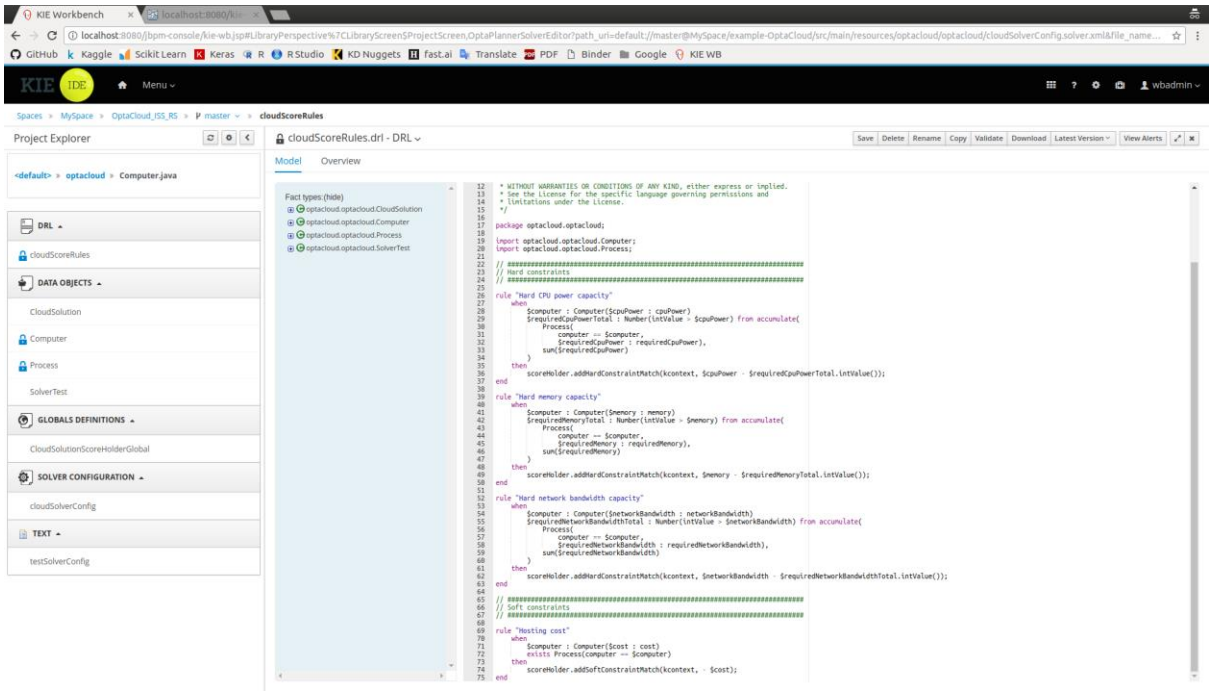
Above: domain objects based on class diagram

[Action] Add field: **id** (type: **long**) to Data Objects: **Computer**, **Process**, **CloudSolution**

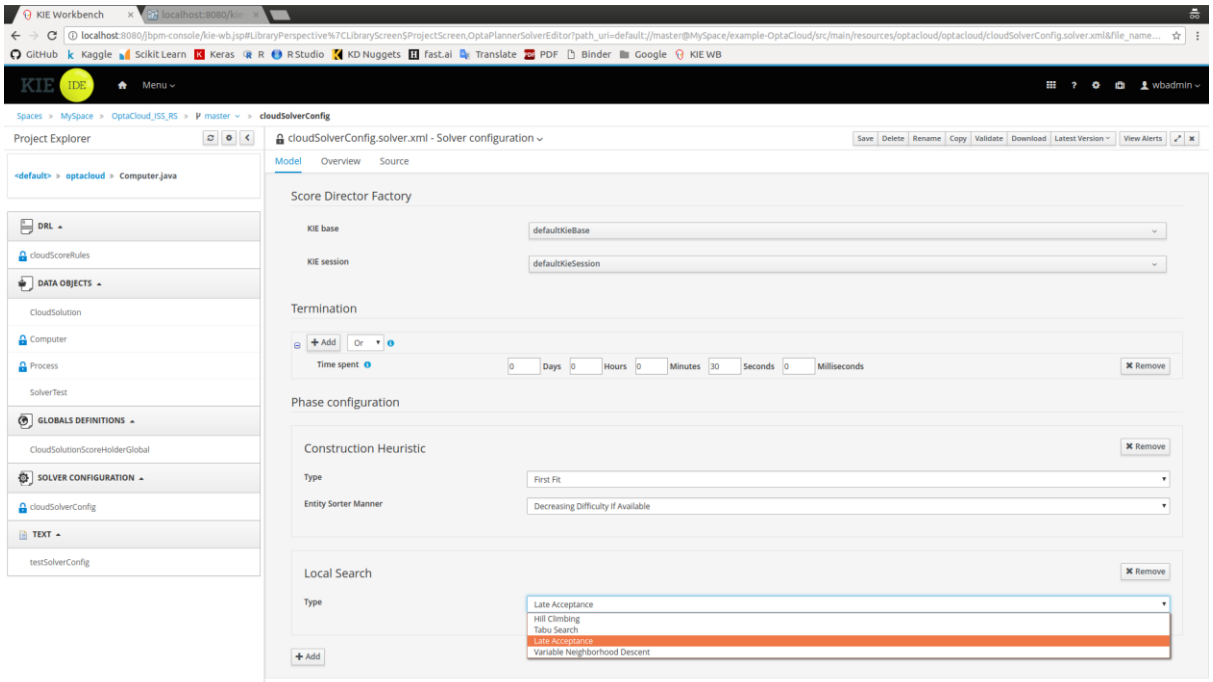


Above: unused SolverTest java program

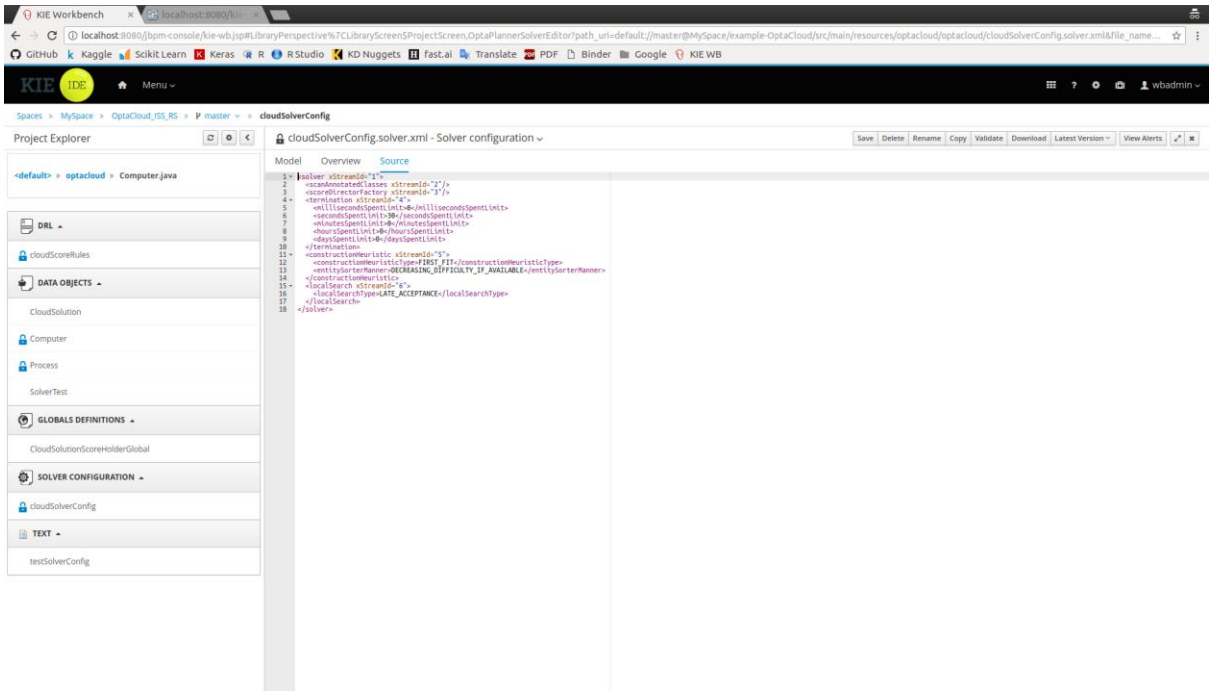
[Action] To prevent Build/Deploy error, comment all code in Java script:
example-OptaCloud/src/test/java/optacloud/optacloud/SolverTest.java



Above: constraints using Drools rule

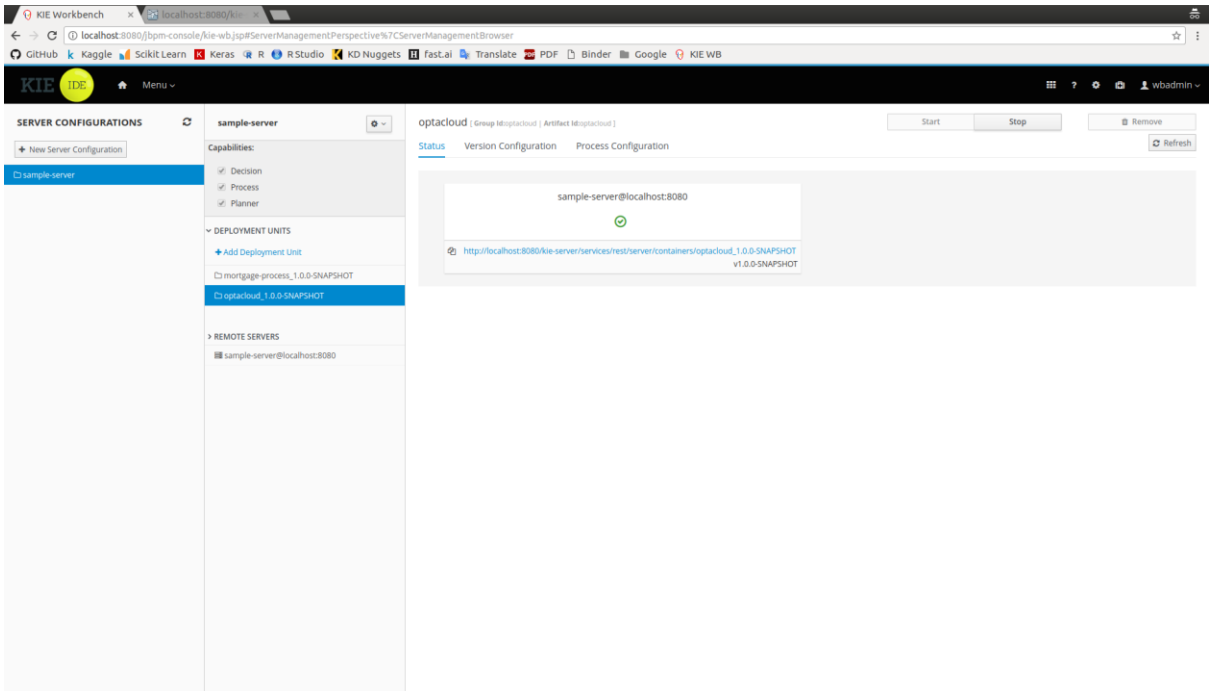


Above: solver configuration

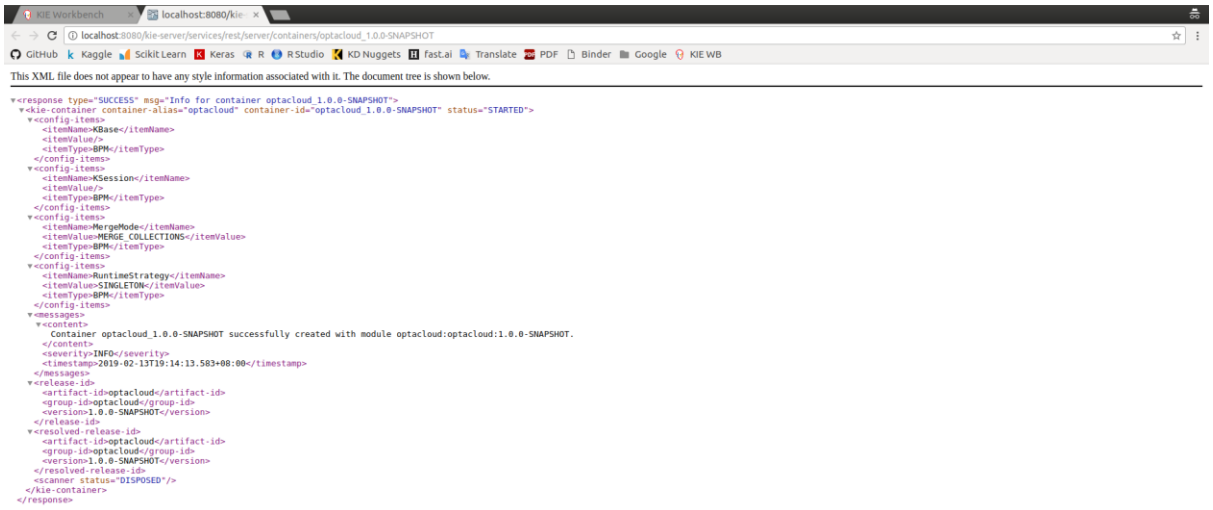


Above: solver configuration (xml)

example-OptaCloud/src/main/resources/optacloud/optacloud/cloudSolverConfig.solver.xml

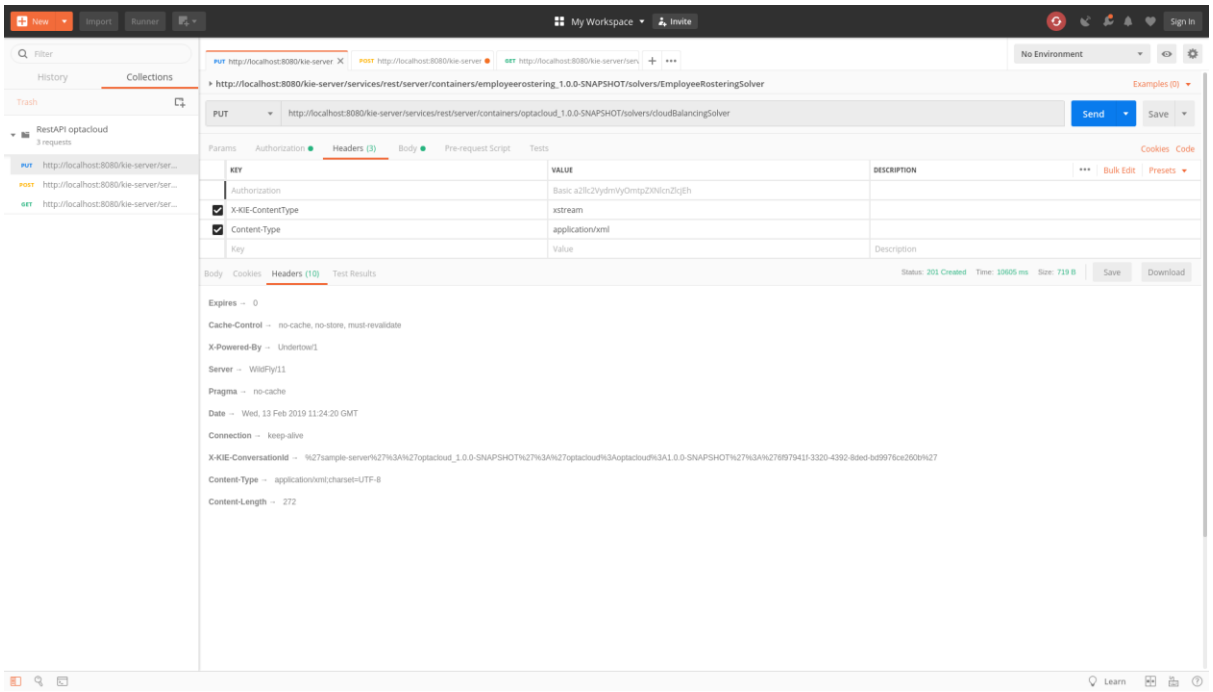


Above: solver deployment to KIE Server

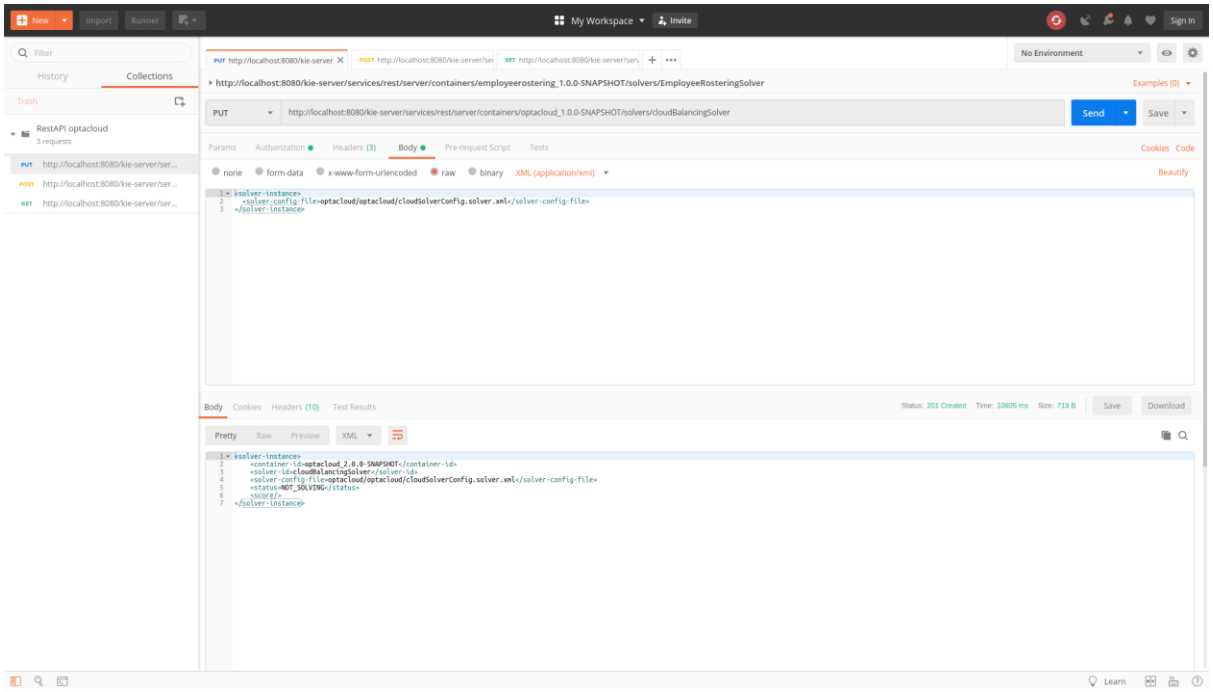


Above: solver web service end point
User: kieserver
Password: kieserver1!

PUT – Initiate a solver object/instance



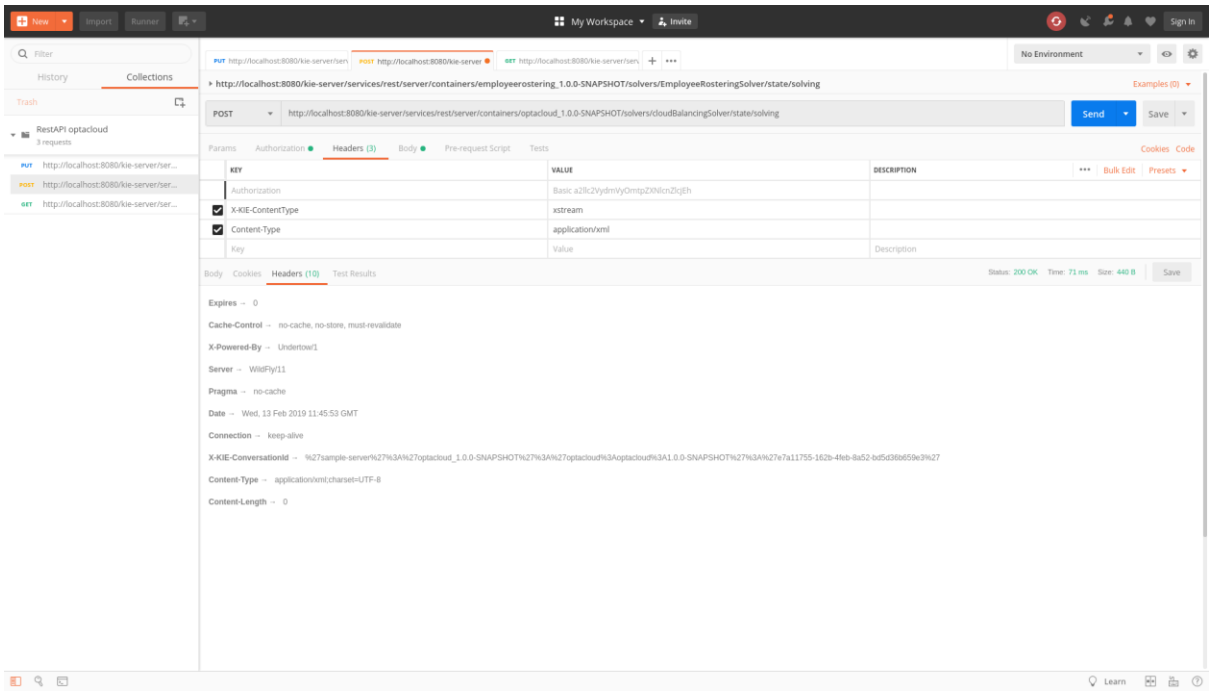
Above: REST-API PUT via Postman tool



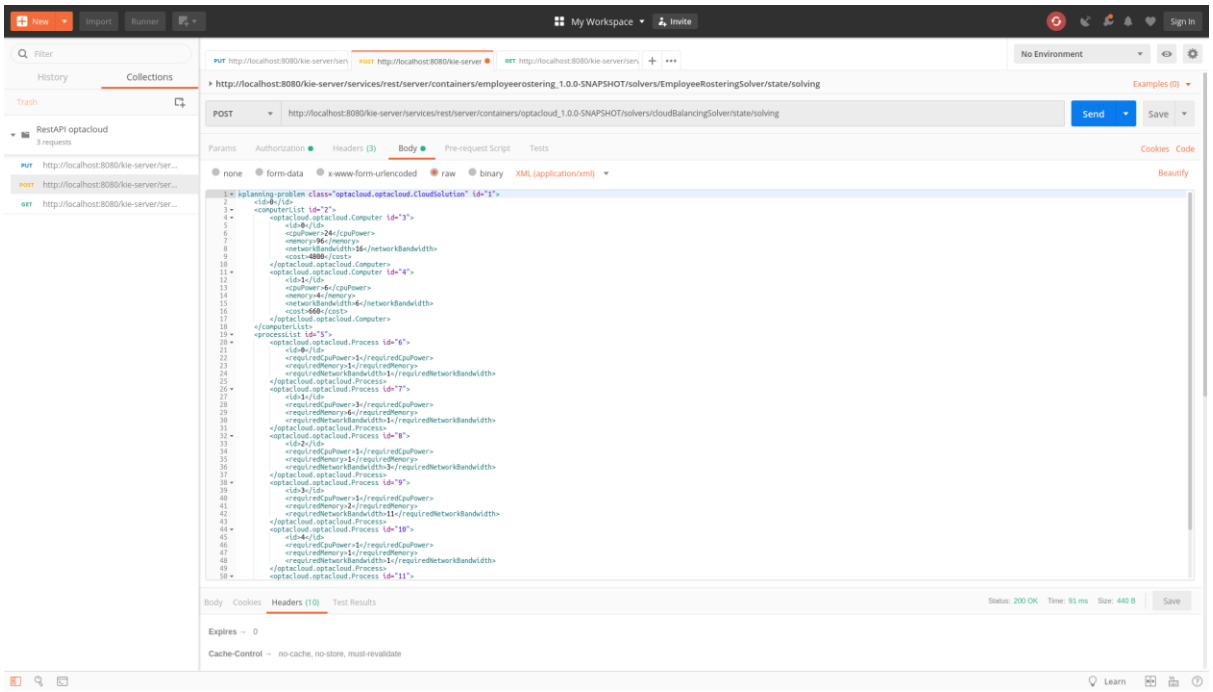
Above: REST-API PUT via Postman tool

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

POST – Start the solver with biz context



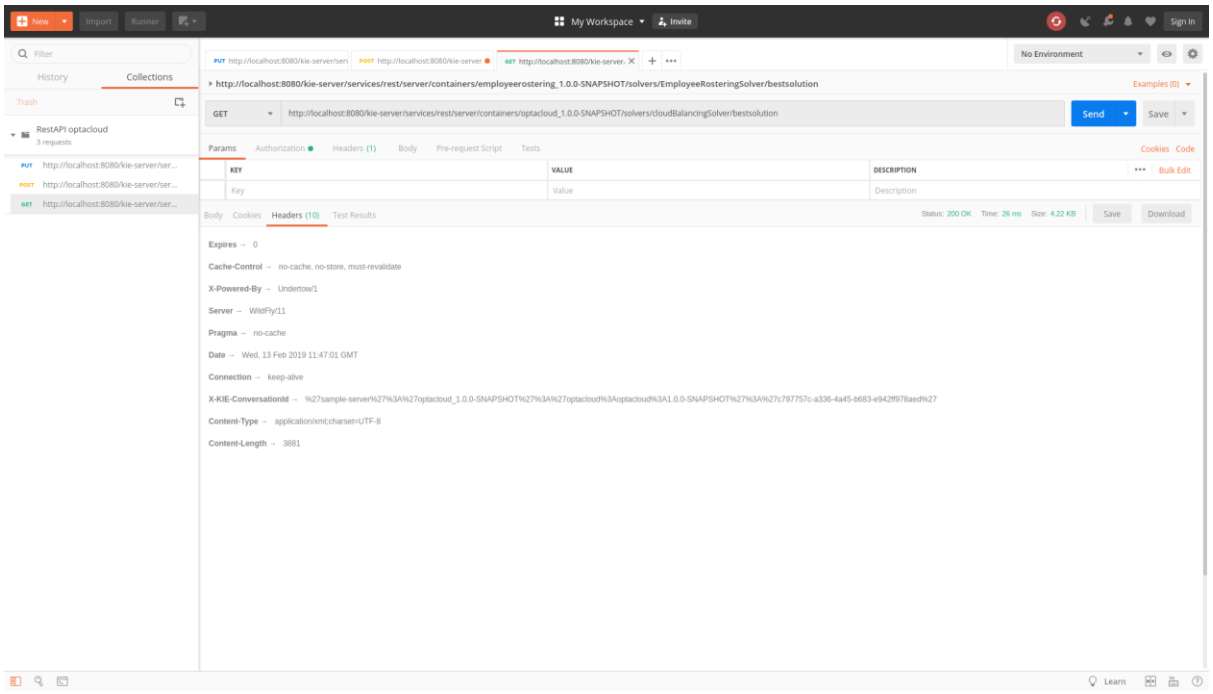
Above: REST-API POST via Postman tool



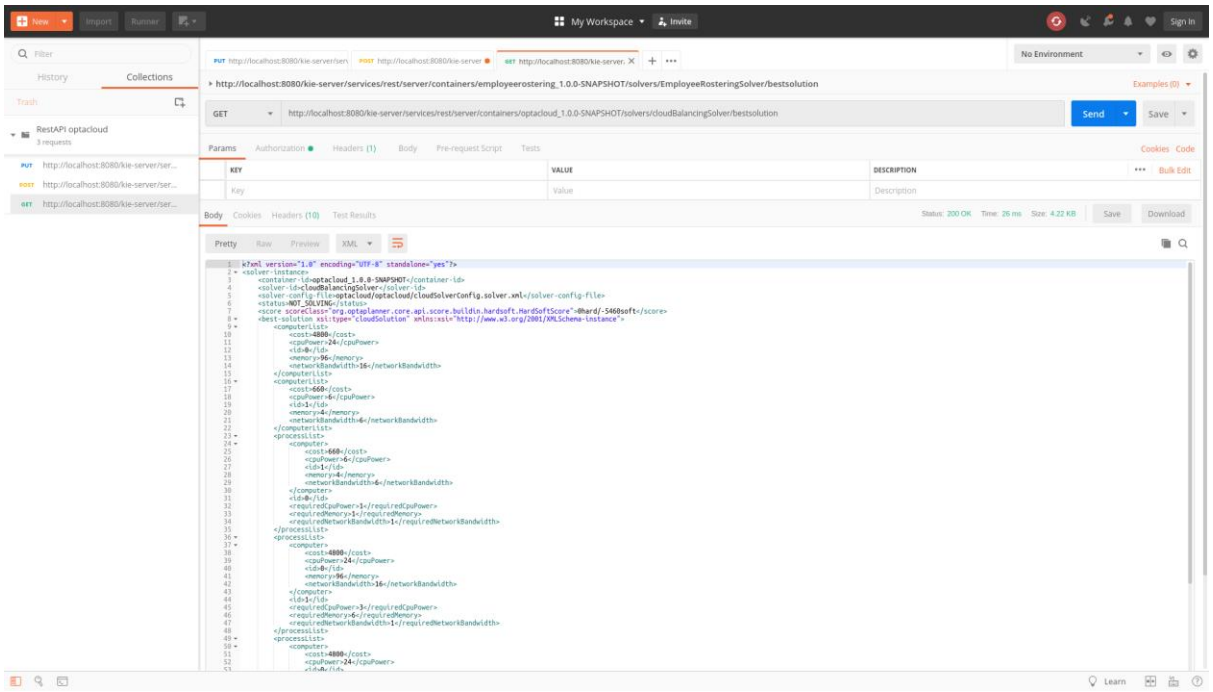
Above: REST-API POST via Postman tool

```
<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>
```

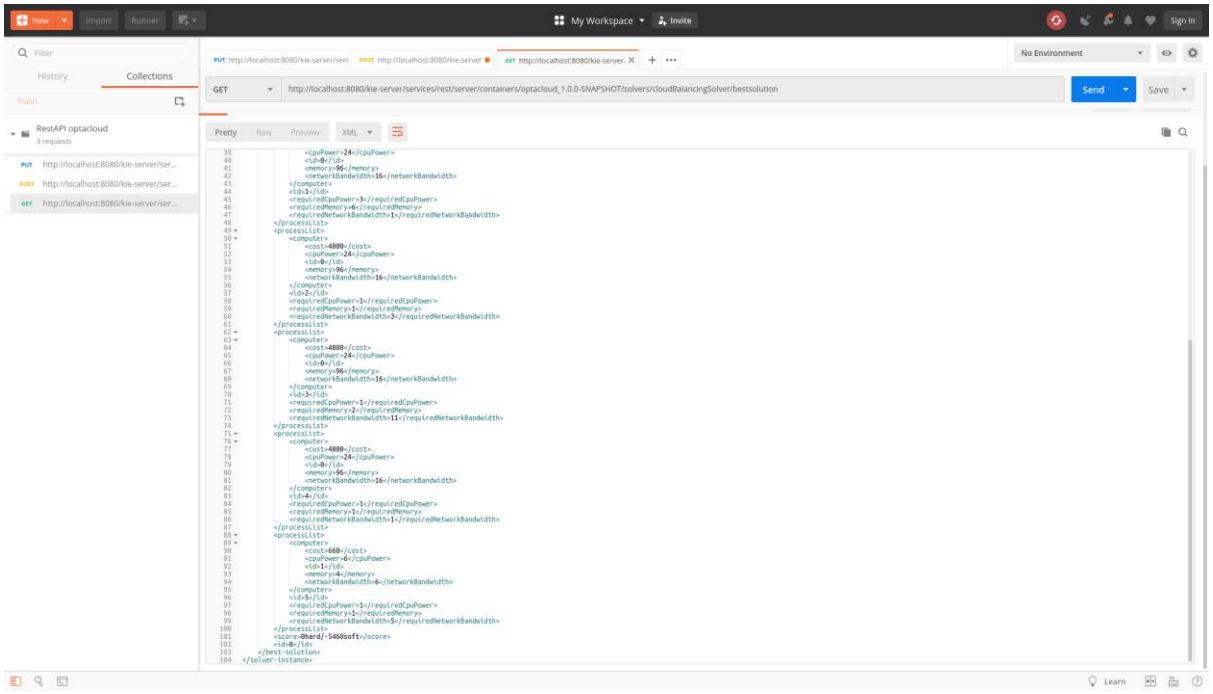
GET – Obtain best solution (xml)



Above: REST-API GET via Postman tool



Above: REST-API GET via Postman tool



Above: REST-API GET 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/

Workshop 2.2 [Individual]

Choose one use case; propose relevant enhancements; then import, analyze, adapt, enhance, and solve/execute.

Enhancement example 1: [Reuse KIE Workbench OptaCloud REST-API project]
Enrich the Cloud Balancing domain model and add extra constraints such as:

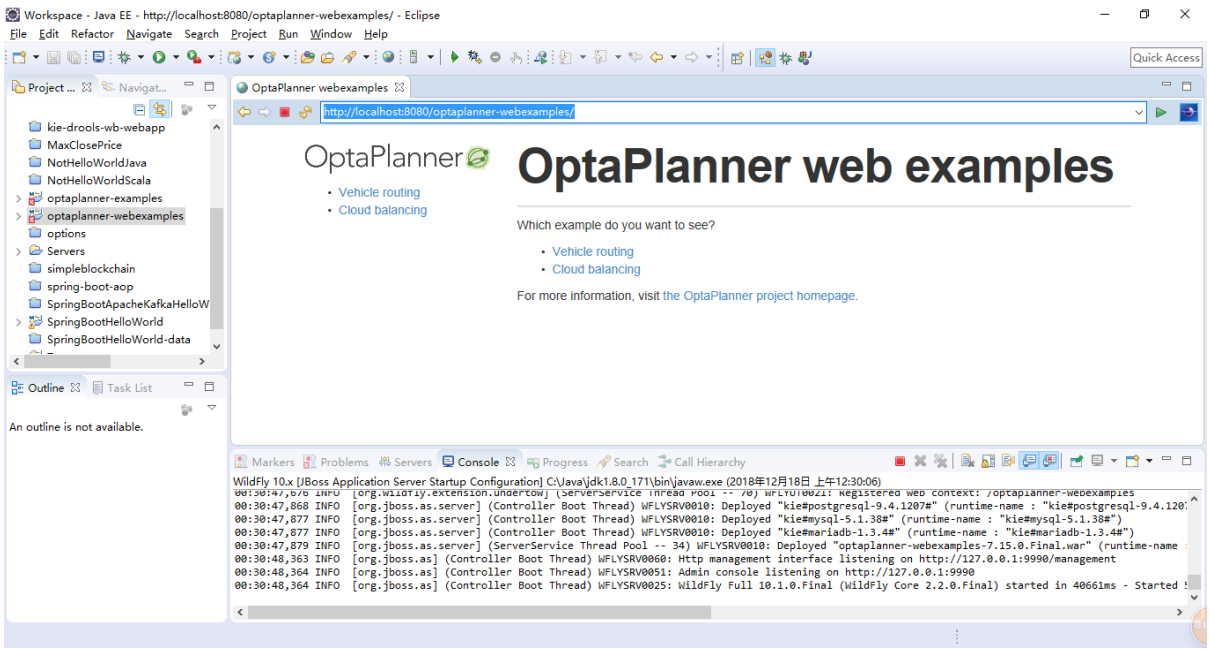
- Some Process running deep learning neural network models can require graphical processing units GPU chips, so these processes should (or must) be assigned to computers with sufficient number of GPU chips.
- Each Process belongs to a Service. A computer might crash, so processes running the same service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

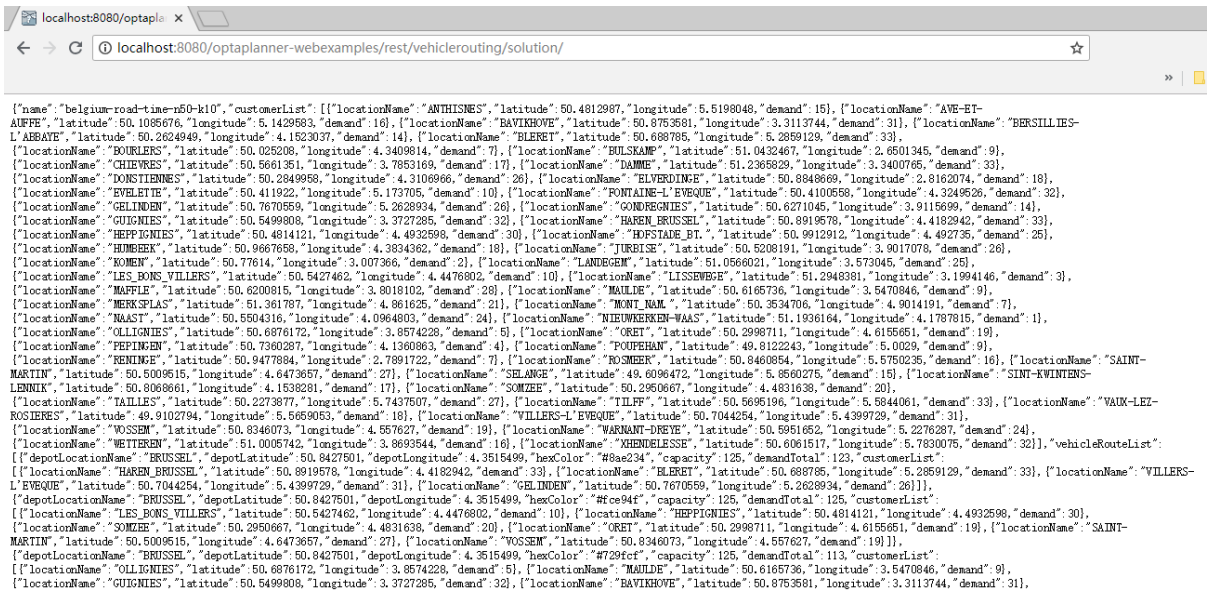
Enhancement example 2: [Reuse any KIE Workbench project or Eclipse project]
Propose and implement/enhance your own relevant business use case.

Enhancement example 3: [Reuse KIE Workbench OptaCloud REST-API project]
Enhance the Cloud Balancing system by introducing a comprehensive user interface for end user to

1. [Input] Key in problem configuration:
    a. Computer instances with: cpuPower, memory, networkBandwidth, cost.
    b. Process instances with: requiredCpuPower, requiredMemory, and requiredNetworkBandwidth.
2. [Output] Display graphical final solution of cloud computer assignments:
    a. Use REST-API for (web based) UI to communicate with KIE web server's solver engine.
    b. Hint: OptaPlanner web example application as reference: https://github.com/kiegroup/optaplanner/tree/master/optaplanner-webexamples



Above: example web applications



Above: example solution enquiry via RESTful API

index.jsp | leaflet.jsp | VehicleRoutingSolver... | DefaultVehicleRouting... | VehicleRoutingRestSe...

```
/**
 * @see DefaultVehicleRoutingRestService
 */
@Path("/vehiclerouting")
public interface VehicleRoutingRestService {

    @GET
    @Path("/solution")
    @Produces("application/json")
    JsonVehicleRoutingSolution getSolution();

    @POST
    @Path("/solution/solve")
    @Produces("application/json")
    JsonMessage solve();

    @POST
    @Path("/solution/terminateEarly")
    @Produces("application/json")
    JsonMessage terminateEarly();
```

Above: example KIE server interaction via RESTful API

index.jsp | leaflet.jsp | VehicleRoutingSolver... | DefaultVehicleRouting... | VehicleRoutingRestSe... | CloudB

```
};

loadSolution = function() {
  $.ajax({
    url: "<%=application.getContextPath()%>/rest/vehiclerouting/solution",
    type: "GET",
    dataType : "json",
    success: function(solution) {
      var markers = [];
      $.each(solution.customerList, function(index, customer) {
        var customerIcon = L.divIcon({
          iconSize: new L.Point(20, 20),
          className: "vehicleRoutingCustomerMarker",
          html: "<span>" + customer.demand + "</span>"
        });
        var marker = L.marker([customer.latitude, customer.longitude], {icon: customerIcon});
        marker.addTo(map).bindPopup(customer.locationName + "</br>Deliver " + customer.demand + " items.");
        markers.push(marker);
      });
      map.fitBounds(L.featureGroup(markers).getBounds());
    }, error : function(jqXHR, textStatus, errorThrown) {ajaxError(jqXHR, textStatus, errorThrown)}
  });
};
```

Above: example server end point

Reference

- https://github.com/kiegroup/optaplanner/tree/master/optaplanner-webexamples
- https://docs.optaplanner.org/latest/optaplanner-docs/html_single/index.html#quickStart

The End of Workshop Project Guide