

[Deploying the simple Two Component application](#)

[Deployment observations](#)

[Deploying a Spark application](#)

[Deployment observations](#)

## Session III

### 3.1. Deploying the simple Two Component application

- 3.1.1. If you are not connected with the Melodic machine, open a terminal and login.

```
ssh -i {{nameOfKey}} ubuntu@{{MELODIC-IP}}
```

- 3.1.2. Check if all component are ready for deploying using alias

```
mping
```

The result should be:

```
[ubuntu@ip-172-31-6-240:~$ mping
cdoserver: 2036: OK 3306: OK
mule: 8088: OK 8089: OK
adapter: 8097: OK 5018: OK
generator: 8091: OK 5015: OK
cpsolver: 8093: OK 5016: OK
camunda: 8095: OK
memcache: 11211: OK
ldap: 389: OK 636: OK
metasolver: 8092: OK
jwtserver: 8094: OK
authdb: 3308: OK
auth-server: 8098: OK
dlmswebservice: 8090: OK
dlmscontroller: 8079: OK
ems: 8111: OK 61616: OK 2222: OK 2099: OK
portainer: 9002: OK

CLOUDIATOR SERVICES
interface: 8080: OK
rest-server: 9000: OK
etcd: 4001: OK 2380: OK 2379: OK
etcd-browser: 8000: OK
influx: 8086: OK
chronograf: 8088: OK
kafka-manager: 8082: OK
kibana: 5601: OK
cadvisor: 8081: OK
portainer: 8083: OK
ubuntu@ip-172-31-6-240:~$
```

3.1.3. Create a directory models and copy the .xmi file

```
mkdir ~/models

cd ~/models/

wget
https://s3-eu-west-1.amazonaws.com/melodic.testing.data/ccgrid/TwoComponentApp.xmi

cd
```

3.1.4. Run the jar with following command and wait for model successfully stored into CDO

```
java
-Deu.paasage.configdir=/home/ubuntu/utls/cdo-uploader/src/main/resources -jar
cdo-uploader-1.0.1-SNAPSHOT-jar-with-dependencies.jar
```

3.1.5. Kill the CDO Uploader process by ctrl+C

3.1.6. Open the following link

<https://reqbin.com/nwcfx5p2>

3.1.7. In bookmark Content update values from the table. If you created a different credentials for the user, please update the username and password accordingly.

variable name	value
melodic-host	your machine public ip

aws-user	notes
aws-secret	notes
nodeGroup	your name*
user	your name*

\* minimum 4 signs, lower case, without special signs and spaces

3.1.8. Alternatively, you can send the request using curl. To do that, replace variables accordingly to the table and use following command.

```
curl -X POST -H 'x-api-key: secure' -H 'Cache-Control: no-cache' -H
'Content-Type: application/json' -d '{
  "applicationId": "TwoComponentApp",
  "username": "user1",
  "password": "ccgrid",
  "cloudDefinitions":
  [

    {
      "endpoint": null,
      "cloudType": "PUBLIC",
      "api": {
        "providerName": "aws-ec2"
      },
      "credential": {
        "user": "{{AWS KEY}}",
        "secret": "{{AWS SECRET}}"
      },
      "cloudConfiguration": {
        "nodeGroup": "{{YOUR NAME}}",
        "properties": {
          "sword.ec2.ami.query": "image-id=ami-09f0b8b3e41191524",
          "sword.ec2.ami.cc.query": "image-id=ami-09f0b8b3e41191524",
          "sword.default.securityGroup": "sg-000c94554210d1820"
        }
      },
      "id": "f009efe1c9d7dfbf962fd13d03b1498e"
    }
  ],
  "watermark": {
    "user": "{{YOUR NAME}}",
    "system": "UI",
    "date": "2016-02-28T16:41:41+0000",
    "uuid": "fb6280ec-1ab8-11e7-93ae-92361f002AAA"
  }
}' -v -i 'http://{{MELODIC IP}}:8088/api/frontend/deploymentProcess'
```

3.1.9. If configurations are set properly, send the request.

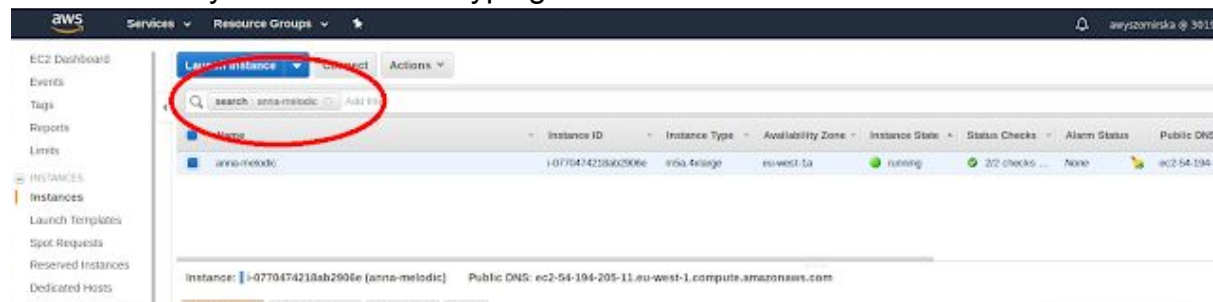
## 3.2. Deployment observations

3.2.1. It is possible to see the process view using Camunda on  
{{MELODIC-IP}}:8095  
admin/admin

3.2.2. Check the logs of the components. For example in the CP Solver log  
you can find the chosen configuration that will be deployed:

```
tail -300f logs/cpsolver.log
```

3.2.3. In the AWS console new components should be visible after few  
minutes. Find your instance out typing the instance name



3.2.4. Let's do a simple test if application works properly:

<http://{{application-ip-host}}:9999/demo/all>

3.2.5. Save name and e-mail to the database

<http://{{application-ip-host}}:9999/demo/add?name=Name&email=email@melodic.com>

3.2.6. Check if it has been saved.

<http://{{application-ip-host}}:9999/demo/all>

### 3.3. Deploying a Spark application

- 3.3.1. If you deployed the TwoComponentApplication previously, you need to restart the Melodic. Use command:

```
drestart
```

- 3.3.2. If you are not connected with the Melodic machine, open a terminal and login.

```
ssh -i {{nameOfKey}} ubuntu@{{MELODIC-IP}}
```

- 3.3.3. Wait until or check if all components are ready to deploy. You can check the status of each component using command:

```
mping
```

- 3.3.4. Open directory models/ (if you do not have one, create: \$mkdir ~/models) and download the .xmi file of the Genome Spark application

```
cd ~/models/  
  
wget  
https://s3-eu-west-1.amazonaws.com/melodic.testing.  
data/ccgrid/Genomnew.xmi
```

- 3.3.5. Set AWSKEY and AWSSECRET from the note and run commands

```
sudo sed -i "s/AWSKEY/{{YOUR AWS KEY}}/g"  
~/models/Genomnew.xmi
```

```
sudo sed -i "s/AWSSECRET/{{YOUR AWS SECRET}}/g"  
~/models/Genomnew.xml
```

- 3.3.6. Run the jar with following command and wait for model successfully stored into CDO

```
cd  
  
java  
-Deu.paasage.configdir=/home/ubuntu/utis/cdo-uploader/src/main/resources -jar  
~/cdo-uploader-1.0.1-SNAPSHOT-jar-with-dependencies.jar
```

- 3.3.7. Kill the CDO Uploader process by ctrl+C

- 3.3.8. Open the following link

<https://reqbin.com/147fhy9g>

- 3.3.9. In bookmark Content update values from table. If you created a different credentials for the user, please update the username and password accordingly.

variable name	value
melodic-host	your machine public ip
aws-user	notes
aws-secret	notes
nodeGroup	your name*
user	your name*

\* minimum 4 signs, lower case, without special signs and spaces

- 3.3.10. Alternatively, you can send the request using curl. To do that, replace variables accordingly to the table and use following command.

```
curl -X POST -H 'x-api-key: secure' -H 'Content-Type: application/json' -H
'Cache-Control: no-cache' -d '{
  "applicationId": "Genomnew",
  "username": "user1",
  "password": "ccgrid",
  "cloudDefinitions":
  [
    {
      "endpoint": null,
      "cloudType": "PUBLIC",
      "api": {
        "providerName": "aws-ec2"
      },
      "credential": {
        "user": "{{AWS USER}}",
        "secret": "{{AWS SECRET}}"
      },
      "cloudConfiguration": {
        "nodeGroup": "{{YOUR NAME}}",
        "properties": {
          "sword.ec2.ami.query": "image-id=ami-08a0a7bee3f024aeb",
          "sword.ec2.ami.cc.query": "image-id=ami-08a0a7bee3f024aeb"
        }
      },
      "id": "f009efelc9d7dfbf962fd13d03b1498e"
    }
  ],
  "watermark": {
    "user": "{{YOUR NAME}}",
    "system": "UI",
    "date": "2016-02-28T16:41:41+0000",
    "uuid": "fb6280ec-1ab8-11e7-93ae-92361f002AAA"
  }
}' -v -i 'http://{{MELODIC IP}}:8088/api/frontend/deploymentProcess'
```

3.3.11. If all configurations are set properly, send the request.

## 3.4. Deployment observations

- 3.4.1. It is possible to see the process view using Camunda on **{{MELODIC-IP}}:8095**
- 3.4.2. Check the logs of the components. For example in the CP Solver log you can find the chosen configuration that will be deployed:

```
tail -300f logs/cpsolver.log
```

- 3.4.3. In the AWS console new components should be visible after few minutes. Find your instance out typing the instance name



- 3.4.4. To check if workers are correctly connected with the Spark Master, check the
- `{{[MELODIC-IP]}}:8181`
- 3.4.5. Check that the metrics are being correctly calculated and passed to the MELODIC in the logs/metasolver.log. Logs with values of MinimumCores metric should be visible there.
- 3.4.6. After around 10 minutes, the reconfiguration should start and more instances should be visible in the AWS console and in the Spark Master UI.