

ID2209 HT23 Distributed Artificial Intelligence and Intelligent Agents

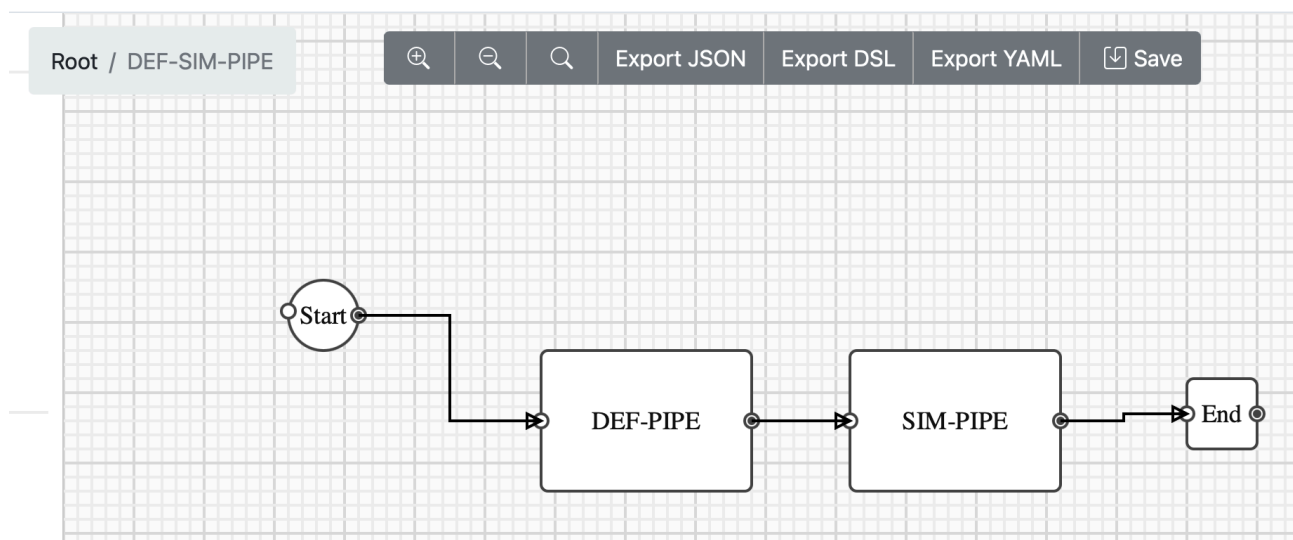
DataCloud Evaluation

by Eugen Lucchiari Hartz

December 01, 2023

Task 1

Screenshot with DEF-SIM-PIPE pipeline graphical view




Generated YAML text from the graphical view of the pipeline

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: def-sim-pipe
spec:
  entrypoint: main-workflow
  templates:
    - name: main-workflow
      dag:
        tasks:
```

- name: task-1
 template: def-pipe
- name: task-2
 dependencies: [task-1]
 template: sim-pipe
- name: def-pipe
 container:
 image: yilinchang/def-pipe-gke
 command: [sh, -c]
 args: ["echo 'Echoing the envParam MQ_HOST: \$MQ_HOST'
 && echo 'Echoing the envParam MQ_PORT: \$MQ_PORT'
 && echo 'Echoing the envParam MQ_USERNAME: \$MQ_USERNAME'
 && echo 'Echoing the envParam MQ_PASSWORD: \$MQ_PASSWORD'"]
 env:
 - name: MQ_HOST
 value: "HOST1"
 - name: MQ_PORT
 value: "PORT1"
 - name: MQ_USERNAME
 value: "def-pipe"
 - name: MQ_PASSWORD
 value: "****"
- name: sim-pipe
 container:
 image: sintef/sim-pipe
 command: [sh, -c]
 args: ["echo 'Echoing the envParam MQ_HOST: \$MQ_HOST'
 && echo 'Echoing the envParam MQ_PORT: \$MQ_PORT'
 && echo 'Echoing the envParam MQ_USERNAME: \$MQ_USERNAME'
 && echo 'Echoing the envParam MQ_PASSWORD: \$MQ_PASSWORD'"]
 env:
 - name: MQ_HOST
 value: "HOST"
 - name: MQ_PORT
 value: "PORT"
 - name: MQ_USERNAME
 value: "SIM-PIPE"
 - name: MQ_PASSWORD
 value: "*****"

Task 2

Protocol of invocation and output of invocation of the API from <http://crowdserv.sys.kth.se:8082/docs>


 **Swagger**
powered by SMARTBEAR

Select a definition **v1**

DEF-PIPE **v1** **OAS3**

/swagger/v1/swagger.json

For the protected end points, you need to generate an access token (e.g. using postman) and provide it by opening "Authorize"

Authorize 

Export

POST /api/export/dsl Transform workflow to dsl

POST /api/export/yaml

Repo

GET /api/repo/s Search for public workflow

GET /api/repo/exportyaml/{user}/{pipeline} Export pipeline in YAML

GET /api/repo/exportyaml/{user}/{pipeline} Export pipeline in YAML

Cancel

Name	Description
user <small>required</small> string (path)	<input type="text" value="testuser"/>
pipeline <small>required</small> string (path)	<input type="text" value="DEF-SIM-PIPE"/>

Execute

Responses

Curl

```
curl -X 'GET' \
'http://crowdserv.sys.kth.se:8082/api/repo/exportyaml/testuser/DEF-SIM-PIPE' \
-H 'accept: text/plain'
```

Request URL

```
http://crowdserv.sys.kth.se:8082/api/repo/exportyaml/testuser/DEF-SIM-PIPE
```

Server response

Code	Details
200	<div><p>Response body</p><pre>{ "data": { "apiVersion": "argoproj.io/v1alpha1", "kind": "Workflow", "metadata": { "generateName": "def-sim-pipe", "spec": { "entrypoint": "main-workflow", "templates": { "name": "main-workflow", "dag": { "tasks": { "name": "task-1", "template": "def-pipe", "container": { "image": "yllinchang/def-pipe-gke", "command": ["sh", "-c"], "args": ["echo 'Echoing the envParam MQ_HOST: \$MQ_HOST' && echo 'Echoing the envParam MQ_PORT: \$MQ_PORT' && echo 'Echoing the envParam MQ_USERNAME: \$MQ_USERNAME' && echo 'Echoing the envParam MQ_PASSWORD: \$MQ_PASSWORD'"] } } } } } } }, "success": true, "errorMessage": null }</pre><p>Response headers</p><pre>content-length: 1625 content-type: application/json; charset=utf-8 date: Thu, 30 Nov 2023 14:40:24 GMT server: Kestrel</pre></div>

Responses		
Code	Description	Links
200	DSL formatted pipeline	No links
<div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <pre>{ "success": true, "errorMessage": "string" }</pre>		

```
curl -X 'GET' \
'http://crowdserv.sys.kth.se:8082/api/repo/exporthyaml/testuser/DEF-SIM-PIPE' \
-H 'accept: text/plain'
```

<http://crowdserv.sys.kth.se:8082/api/repo/exporthyaml/testuser/DEF-SIM-PIPE>

```
{
  "data": "apiVersion: argoproj.io/v1alpha1\nkind: Workflow\nmetadata:\n  generateName: def-sim-pipe\nspec:\n  entrypoint: main-workflow\n  templates:\n    - name: main-workflow\n      dag:\n        tasks:\n          - name: task-1\n            template: def-pipe\n          - name: task-2\n            dependencies: [task-1]\n            template: sim-pipe\n          - name: def-pipe\n            container:\n              image: yilinchang/def-pipe-gke\n              command: [sh, -c]\n              args: [\"echo 'Echoing the envParam MQ_HOST: $MQ_HOST'\n&& echo 'Echoing the envParam MQ_PORT: $MQ_PORT'\n&& echo 'Echoing the envParam MQ_USERNAME: $MQ_USERNAME'\n&& echo 'Echoing the envParam MQ_PASSWORD: $MQ_PASSWORD']\n              env:\n                - name: MQ_HOST\n                  value: \"HOST1\"\n                - name: MQ_PORT\n                  value: \"PORT1\"\n                - name: MQ_USERNAME\n                  value: \"def-pipe\"\n                - name: MQ_PASSWORD\n                  value: \"***\"\n          - name: sim-pipe\n            container:\n              image: sintef/sim-pipe\n              command: [sh, -c]\n              args: [\"echo 'Echoing the envParam MQ_HOST: $MQ_HOST'\n&& echo 'Echoing the envParam MQ_PORT: $MQ_PORT'\n&& echo 'Echoing the envParam MQ_USERNAME: $MQ_USERNAME'\n&& echo 'Echoing the envParam MQ_PASSWORD: $MQ_PASSWORD']\n              env:\n                - name: MQ_HOST\n                  value: \"HOST\"\n                - name: MQ_PORT\n                  value: \"PORT\"\n                - name: MQ_USERNAME\n                  value: \"SIM-PIPE\"\n                - name: MQ_PASSWORD\n                  value: \"****\"\n            \"success\": true,\n            \"errorMessage\": null\n        }\n    }
```

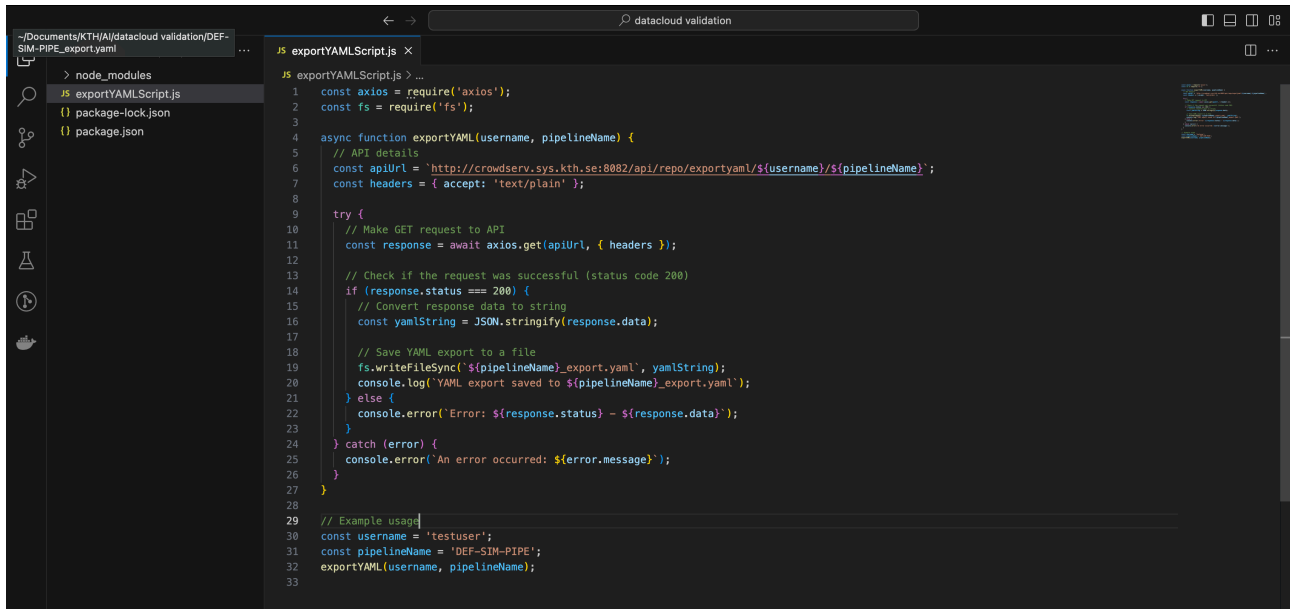
```
content-length: 1625
content-type: application/json; charset=utf-8
date: Thu,30 Nov 2023 14:40:24 GMT
server: Kestrel
```

```
{
  "success": true,
  "errorMessage": "string"
```

```
}
```

Task 3

Code for invocation of API from your program and invocation result



```
1  const axios = require('axios');
2  const fs = require('fs');
3
4  async function exportYAML(username, pipelineName) {
5    // API details
6    const apiUrl = 'http://crowdserv.sys.kth.se:8082/api/repo/exportyaml/${username}/${pipelineName}';
7    const headers = { accept: 'text/plain' };
8
9    try {
10     // Make GET request to API
11     const response = await axios.get(apiUrl, { headers });
12
13     // Check if the request was successful (status code 200)
14     if (response.status === 200) {
15       // Convert response data to string
16       const yamlString = JSON.stringify(response.data);
17
18       // Save YAML export to a file
19       fs.writeFileSync(`${pipelineName}_export.yaml`, yamlString);
20       console.log(`YAML export saved to ${pipelineName}_export.yaml`);
21     } else {
22       console.error(`Error: ${response.status} - ${response.data}`);
23     }
24   } catch (error) {
25     console.error(`An error occurred: ${error.message}`);
26   }
27 }
28
29 // Example usage
30 const username = 'testuser';
31 const pipelineName = 'DEF-SIM-PIPE';
32 exportYAML(username, pipelineName);
33
```

exportYAMLScript.js

```
const axios = require('axios');
const fs = require('fs');
```

```
async function exportYAML(username, pipelineName) {
  // API details
  const apiUrl = `http://crowdserv.sys.kth.se:8082/api/repo/
exportyaml/${username}/${pipelineName}`;
  const headers = { accept: 'text/plain' };
```

```
  try {
    // Make GET request to API
    const response = await axios.get(apiUrl, { headers });
```

```
    // Check if the request was successful (status code 200)
    if (response.status === 200) {
      // Convert response data to string
      const yamlString = JSON.stringify(response.data);
```

```
      // Save YAML export to a file
      fs.writeFileSync(`${pipelineName}_export.yaml`, yamlString);
      console.log(`YAML export saved to ${pipelineName}
_export.yaml`);
    } else {
```

```

        console.error(`Error: ${response.status} - ${response.data}`);
    });
}
} catch (error) {
    console.error(`An error occurred: ${error.message}`);
}
}
}

```

```

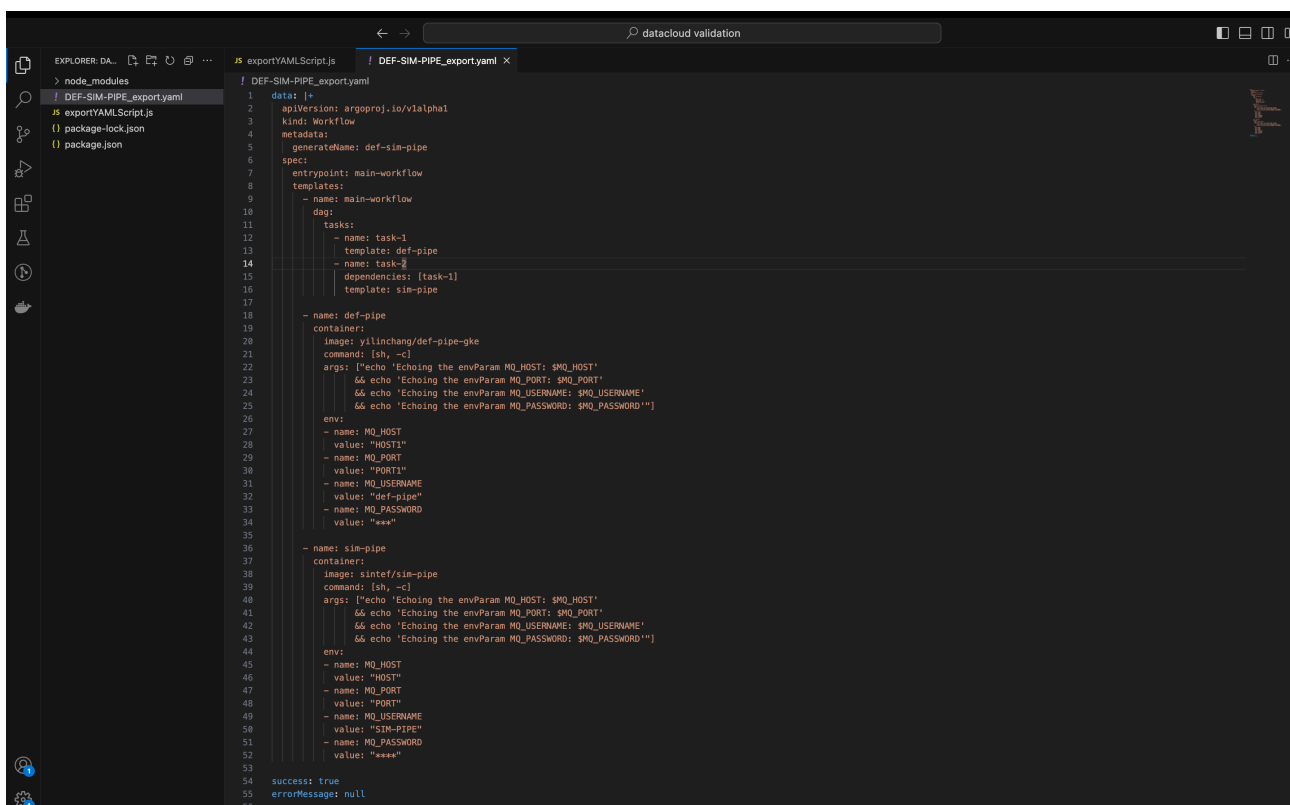
// Example usage
const username = 'testuser';
const pipelineName = 'DEF-SIM-PIPE';
exportYAML(username, pipelineName);

```

Invocation result

you get the result after running `node exportYAMLScript.js` in the terminal

DEF-SIM-PIPE_export.yaml



```

data: |+
  apiVersion: argoproj.io/v1alpha1
  kind: Workflow
  metadata:
    generateName: def-sim-pipe

```

```
spec:
  entrypoint: main-workflow
  templates:
    - name: main-workflow
      dag:
        tasks:
          - name: task-1
            template: def-pipe
          - name: task-2
            dependencies: [task-1]
            template: sim-pipe
```

```
    - name: def-pipe
      container:
        image: yilinchang/def-pipe-gke
        command: [sh, -c]
        args: ["echo 'Echoing the envParam MQ_HOST: $MQ_HOST'
          && echo 'Echoing the envParam MQ_PORT: $MQ_PORT'
          && echo 'Echoing the envParam MQ_USERNAME:
$MQ_USERNAME'
          && echo 'Echoing the envParam MQ_PASSWORD:
$MQ_PASSWORD'"]
        env:
          - name: MQ_HOST
            value: "HOST1"
          - name: MQ_PORT
            value: "PORT1"
          - name: MQ_USERNAME
            value: "def-pipe"
          - name: MQ_PASSWORD
            value: "***"
```

```
    - name: sim-pipe
      container:
        image: sintef/sim-pipe
        command: [sh, -c]
        args: ["echo 'Echoing the envParam MQ_HOST: $MQ_HOST'
          && echo 'Echoing the envParam MQ_PORT: $MQ_PORT'
          && echo 'Echoing the envParam MQ_USERNAME:
$MQ_USERNAME'
          && echo 'Echoing the envParam MQ_PASSWORD:
$MQ_PASSWORD'"]
        env:
          - name: MQ_HOST
            value: "HOST"
          - name: MQ_PORT
            value: "PORT"
          - name: MQ_USERNAME
            value: "SIM-PIPE"
          - name: MQ_PASSWORD
            value: "****"
```

```
success: true  
errorMessage: null
```