

Report: Homework 7 - Amazon Flow Framework

Jan SCHLENKER

May 28, 2015

Instructor:	Dipl.-Ing. Dr. Simon Ostermann
Programming language:	Java
Library used:	Java AWS SDK & Jsch (SSH-library)
Total points:	10

1 Task 1

To execute the `AwsFlowFramework` samples from the Java AWS SDK, the `README.md` of the samples is pretty useful, which worked for me (Extract follows):

Prerequisites

You must have a valid Amazon Web Services developer account.

Requires the AWS SDK for Java. For more information on the AWS SDK for Java, see (<http://aws.amazon.com/sdkforjava>).

You must be signed up for the following services:

Amazon Simple Workflow Service (SWF). For more information, see (<http://aws.amazon.com/swf>).

Amazon Simple Storage Service (S3). For more information, see (<http://aws.amazon.com/s3>).

JUnit (version 4.7) is required to run the samples. The jar file must be in the classpath. For more information, see (<http://www.junit.org/>)

`org.springframework.test` (version 3.0) is required to run the samples. The jar file must be in the classpath. For more information, see (<http://www.springsource.org/>)

Log4j (version 1.2.15) is required to run the samples. The jar file must be in the classpath. For more information, see (<http://logging.apache.org/log4j/1.2/>)

Running the Samples

The steps for running the AWS Flow Framework samples are:

1. Create the Samples domain
1. Go to the SWF Management Console
(<https://console.aws.amazon.com/swf/home>).
2. Follow the on-screen instructions to log in.
3. Click Manage Domains and register a new domain with the name Samples.

2. Open the access.properties in the samples folder.

3. Locate the following sections and fill in your Access Key ID and Secret Access Key. You can use the same values for SWF and S3:

```
““
# Fill in your AWS Access Key ID and Secret Access Key for SWF
# http://aws.amazon.com/security-credentials
AWS.Access.ID=;Your AWS Access Key;
AWS.Secret.Key=;Your AWS Secret Key;
AWS.Account.ID=;Your AWS Account ID;
```

```
# Fill in your AWS Access Key ID and Secret Access Key for S3
# http://aws.amazon.com/security-credentials
S3.Access.ID=;Your AWS Access Key;
S3.Secret.Key=;Your AWS Secret Key;
S3.Account.ID=;Your AWS Account ID;
```

4. Some samples upload files to S3. Locate the following section and fill in the name of S3 bucket that you want the samples to use:

```
““
##### FileProcessing Sample Config Values #####
Workflow.Input.TargetBucketName=;Your S3 bucket name;
““
```

5. Save the file.

6. Set the environment variable `AWS_SWF_SAMPLES_CONFIG` to the full

path of the directory containing the access.properties file. For example on windows run this command:

```
“  
set AWS_SWF_SAMPLES_CONFIG=%Your SDK Directory%/src/samples/AwsFlowFramework  
“
```

and on linux use this command to set the environment variable:

```
“  
export AWS_SWF_SAMPLES_CONFIG=%Your SDK Directory%/src/samples/-  
AwsFlowFramework  
“
```

7. Compile the samples by using the Ant build.xml file. This will create binaries in bin directory under the samples directory.

8. To run the samples follow these instructions:

Hello World Sample:

The sample has three executables. You should run each in a separate terminal/console.

- Run: ‘ant -f build.xml -Dmain-class=”com.amazonaws.services.simpleworkflow.-flow.examples.helloworld.ActivityHost” run‘

- Run: ‘ant -f build.xml -Dmain-class=”com.amazonaws.services.simpleworkflow.-flow.examples.helloworld.WorkflowHost” run‘

- Run: ‘ant -f build.xml -Dmain-class=”com.amazonaws.services.simpleworkflow.-flow.examples.helloworld.WorkflowExecutionStarter” run‘

Booking Sample:

The sample has three executables. You should run each in a separate terminal/console. From the samples folder,

- Run: ‘ant -f build.xml -Dmain-class=”com.amazonaws.services.simpleworkflow.-flow.examples.booking.ActivityHost” run‘

- Run: ‘ant -f build.xml -Dmain-class=”com.amazonaws.services.simpleworkflow.-

```
flow.examples.booking.WorkflowHost" run'  
- Run: 'ant -f build.xml -Dmain-class="com.amazonaws.services.simpleworkflow.-  
flow.examples.booking.WorkflowExecutionStarter" run'
```

The HelloWorld and the Booking example worked for me. I used a similiar Ant build file for my own programme. I also tried Maven, but it was not easy to correctly configure the aspectj part, even if there is quite a good documentation under <http://stackoverflow.com/questions/9392655/how-to-consume-amazon-swf>. My problem with (pure) Ant ist, that the uploaded files are pretty big because all libraries are included (time was too short to add dependency management with e.g. Ivy).

2 Task 2

2.1 Requirements

- Java 1.7
- Ant 1.9.2

2.2 How to run the programme

First of all extract the archive file `homework_7.tar.gz` (sry for the big size!):

```
$ tar -xzf homework_7.tar.gz  
$ cd homework_5.2
```

Adapt the settings in `access.properties`:

```
$ vi access.properties &
```

Now you need two terminals. In the first terminal do:

```
$ ant run-workflow-worker
```

In the second do:

```
$ ant run -Darg0=<instances>
```

where `instances` is the number of instances you would like to launch.

2.3 Programme explanation

The files of the the programme are structured as follows:

- The `src` directory contains the source files
- The `lib` directory contains the library files of AWS SDK and JSch
- The `build.xml` file contains build information for Ant
- The `access.properties` contains AWS credential data

The file `src/piEstimator/PiEstimator/PiEstimatorMain` contains the main method for the ant run job. The programme basically creates as many ec2 instances as given, copies an ActivityWorker jar (generated by Ant before run task) and the access.properties to the instance and runs this worker. Afterwards it creates an SWFClient proxy and indirectly creates 10 * given-instance-number activities.