

Please Report any problems or feedback to flowscale-dev@googlegroups.com

Run From Source

Prerequisites :

- Eclipse 3.6.2
- Mysql 5.0 (For this tutorial we're using mysql, but in reality you could use any database)
- [CPAN](#) (optional)

Import Beacon and OpenflowJ

Flowscale's engine is the [Beacon](#) controller. We need to download the source code for both the Beacon controller and OpenflowJ.

The latest Beacon release (1.0.2) can be found [here](#).

The latest OpenflowJ release (1.0.2) can be found [here](#).

- Extract those folders to any location on your computer. (For documentation purposes we'll call the locations <beacon-path> and <openflowj-path>)
- Fire up Eclipse and create a new workstation
- Import the Beacon Project to Eclipse (File -> Import -> General -> Existing Projects into workspace)
- Browse to <beacon-path> select the folder
- Click Deselect all and select the following projects
 - Beacon Libraries
 - Beacon Main Target
 - net.beaconcontroller.core
 - net.beaconcontroller.logging.bridge
 - net.beaconcontroller.logging.config
 - net.beaconcontroller.packet
- Import OpenflowJ to Eclipse the same way. This time keep the select all

Import FlowScale

Get FlowScale from github

```
git clone git://github.com/InCNTRE/FlowScale.git
```

- Import FlowScale into Eclipse * Import -> General -> Existing Projects into workspace
- Browse to <path-to-FlowScale>/backend
- Select All projects and click 'OK'

Resolving Target Definitions

One you move to the Eclipse workbench, you'll find that most of the bundles contain errors. We need to resolve those errors by downloading the dependencies.

- On the left pane Double-Click the FlowScale Target Platform

- Right click on main.target.example select "refactor" and rename it to main.target
- Double click on the file. It will appear on the right, in the bottom it will say "Resolving Target Definitions" wait until it finishes. It is downloading all the necessary libraries. So, it will take a while the first time.
- Once it is done in the top right, click on "Set as Target Platform"

All errors should be cleared now

Deploy the Database

The database carries all the information that needs to be persistent.

- Set up the flowscale_db database
- Access mysql root
- create a new database
 - `create database flowscale_db;`
- Use the script file <path-to-FlowScale>/database/flowscale_db.sql to create the tables. For example:
- `mysql -u root -D flowscale_db -p < flowscale_db.sql`

Update Backend Configuration

We need to update all of the configuration values in the bundles to allow FlowScale to run, most of these values should be customized according to your own environment

- For all FlowScale bundles rename the META-INF/spring/context.xml.example file to context.xml:
- edu.iu.incentre.flowscale
- edu.iu.incentre.flowscaleflowupdate
- edu.iu.incentre.flowscalestatcollector
- edu.iu.incentre.flowscalehttplistener
- org.eclipse.equinox.console.telnet
- You need to edit some of the context.xml files to reflect your own environment as detailed below.

Database configuration

- For documentation purposes I'll assume the database is under localhost with the listening port being 3306. The username and password of the database is "flowscale". The JDBC driver being used is com.mysql.jdbc.Driver.
- Edit the context.xml of edu.iu.incentre.flowscale
 - set the username property to "flowscale"
 - set the password property to "flowscale"
 - set the connectionString property to "jdbc:mysql://127.0.0.1:3306/flowscale_db"
 - set the dbDriverString property to "com.mysql.jdbc.Driver"
- Edit the context.xml of edu.iu.incentre.flowscalestatcollector
 - set the databaseClass property to com.mysql.jdbc.Driver

- set the dbUsername property to flowscale
- set the dbPassword property to flowscale
- set the databaseDriver property to "dbc:mysql://localhost:3306/flowscale_db"
- Edit the context.xml of edu.iu.incentre.flowscaleflowupdate
 - set the databaseClass property to com.mysql.jdbc.Driver
 - set the dbUsername property to flowscale
 - set the dbPassword property to flowscale
 - set the databaseDriver property to "dbc:mysql://localhost:3306/flowscale_db"

Switch Configuration

Some of bundles also need information about the switch or switches being used. FlowScale will need the datapath ID of the switch. Otherwise, it will ignore it (even if the switch connects to the controllers). For documentation purposes assume we have 2 switches.. The first with datapathId abcdefabcdef and another with datapathId 123456123456.

Assume we would like to have the traffic output distribution from the switch on ports 5,6,7,8 for the first switch and the second switch with output distribution 15,16,17,18.

- edit edu.iu.incentre.flowscaleflowupdate
 - set the loadedPorts property to "abcdefabcdef:5,6,7,8-123456123456:15,16,17,18"
- edit edu.iu.incentre.flowscalestatcollector
 - set the datapathIdString property to "abcdefabcdef,123456123456"

Other Optional Configuration

[coming soon]

It is time to run the FlowScale main engine

Running FlowScale

- under edu.iu.incentre.flowscaleproduct
 - Under the configuration tab, on the top right click on the validate tab it should say "no problems detected"
 - To run the application in debug mode click on the debug icon and choose "Launch in Eclipse application mode"

That should startup FlowScale you should see the following ports on your machine listening:

- 6633 : This is the Beacon Controller listening port
- 8080 : This the http interface for FlowScale accepting REST API requests
- 7777: This is to access the OSGI console to manage the bundles
- Try telneting into localhost (port 7777) and type "ss".. The bundles should appear

Setting up the Web User Interface "flowscale_ui"

The FlowScale engine is now running. But you still need to interact with it. Currently the only way is by using REST API calls. The web interface would allow you to make those requests to the engine. The flowscale_ui directory under FlowScale provides the files to run the webui.

The WebUI was tested on Apache2. It should, however, run on any server. We'll assume here the that

the root directory is <path-to-webserver>

First of all, make sure you've got perl on the machine you are running the WebUi on.

You will need the following perl modules to be installed. The best way to get those modules is through CPAN.

Perl Packages Required:

- libhtml-template-perl
- Template
- JSON
- File::ShareDir
- REST::Client
- RRDS
- Under flowscale_ui/conf
 - Rename controller.xml.example to controller.xml and set the hostname and port properties to point to the controller. Assuming you followed this guide to running the FlowScale engine, the hostname should be "127.0.0.1" and the port should be "8080"
 - Rename database.xml.example to database.xml and set the the credentials username property to "flowscale" and password property to "flowscale". Do the same thing for flow_stat credentials. Leave the other values as is.

Updating Absolute Paths in the code

You need to update the absolute paths included in the code.

- Update lib/Flowscale/Graph.pm
 - The **\$imgDir** variable should have the absolute path of where the images should be temporarily stored. Replace absolute path /var/www/img with <path-to-webserver>/flowscale_ui/img
 - **%creds_args** variable should be <path-to-webserver>/flowscale_ui/conf/database.xml
- Update lib/Flowscale/Controller.pm
 - Set the config property in the %args variable to <path-to-webserver>/flowscale_ui/conf/controller.xml
 - e.g.

```
%args = (          config =>
'/var/www/flowscale_ui/conf/controller.xml',
                @_,
            );
```
 - Update lib/Flowscale/Database.pm
 - Set the config property in the %args variable to <path-to-webserver>/flowscale_ui/conf/database.xml
 - e.g.

```
%args = (          config =>
'/var/www/flowscale_ui/conf/database.xml',
                @_,
            );
```

You should be able to access the WebUI through your webserver [url]/flowscale_ui/index.cgi?action=switch

Start registering your switches as described [here](#) .

You will need to populate those switches with flows that will get your traffic going. The default rule is to drop all your traffic. (Not very helpful).

Start adding your groups as described [here](#)

Run From Binary

Download the binary for your platform

<https://github.com/InCNTRE/FlowScale/downloads>

Configure FlowScale

There are 5 property files that are involved with FlowScale. Those files

1. beacon.properties : This file belongs to the Beacon Controller. The listening port can be edited in this file (e.g. **controller.listenPort=6655** will make Beacon listen on TCP port 6655)
2. flowscale.properties: Sets the values of the main core bundle (edu.iu.incetre.flowscale) with the following possible properties (NOTE: not all properties need to be set)
 1. flowscaleController.username - This is the username of the account used to access flowscale_db
 2. flowscaleController.password - This is the password of the account used to access flowscale_db
 3. flowscaleController.connectionString - The connection string of the database (e.g. jdbc:mysql://127.0.0.1:3306/flowscale_db)
 4. flowscaleController.driverString - The Java driver used to connect to the database
 5. flowscaleController.defaultRulePriority - The priority of the default rule (drop all traffic). By default, this is set to 5.
 6. flowscaleController.mirrorPriority - Sets the priority of the flows that will mirror traffic Mirroring traffic to other output ports (**only to be used if the switch does not support masking the input ports when mirroring**). See <https://mailman.stanford.edu/pipermail/openflow-indigo/2012-February/000331.html>
 7. flowscaleController.mirroringRules - sets the mirroring rules (**only to be used if the switch does not support masking the input ports when mirroring**)
 8. flowscaleController.flowMirrorPorts - If you want to mirror traffic from outgoing ports to other ports set this property.
 9. flowscaleController.maximumFlowsToPush - How many flows to push onto the switch in one command (some switches crash if a large number of flows are pushed at the same time)
3. collector.properties : This file is used for polling statistics from the OpenFlow switch. The following below values could be set.
 1. statCollector.datapathIdStrings - Set the datapath ID of the OpenFlow switches here
 2. statCollector.isQuery - boolean value if this bundle should be enabled on startup or not. Default is true
 3. statCollector.intervalTime - Time in ms of the interval between each request to the OpenFlow switch. E.g. if set to 30000, the switches will be queried every 30 seconds

4. statCollector.databaseClass - Java driver for the database
5. statCollector.databaseDriver - database connection string
6. statCollector.dbUsername - username of the database use to connect to the statistics tables
7. statCollector.dbPassword - password of the database use to connect to the statistics tables
4. hotswap.properties : FlowScale does hotswapping as well. The bundle edu.iu.incentre.flowscaleflowupdate handles the hotswapping thread, which is basically a daemon performing hot swapping of flows from ports with a high load to ones with a lower load. The following values could be set :
 1. flowUpdate.databaseClass - The java driver of the database
 2. flowUpdate.databaseDriver - The connection string of the database
 - flowUpdate.intervalTime - The time in seconds of how long the daemon should be idle before its next run. Setting this to a really long time might allow a port to receive a high load of traffic for a long period. Setting it too short might result in false positives in that flows will be moved just from a burst.
 3. flowUpdate.loadedPorts - Specify the OpenFlow switch and the outgoing ports used for balancing the outgoing traffic. E.g. 000000000001:5,6,7,8,-000000000002:26,28,30,32,34,36,38
 4. flowUpdate.dbUsername - username that will access the hotswap tables in flowscale_db
 5. flowUpdate.dbPassword - password that will access the hotswap tables in flowscale_db
5. cli.properties : Set the listening port (localhost only) of the OSGI console to manage the bundles
 1. commandInterface.socketPort - set the port here
6. web_interface.properties : Set the listening port of the embedded web interface (used to listen to REST API requests)
 1. flowscaleHttpListener.jettyListenerPort - set the port here

Configure Logging (optional)

You may want to configure the logging level and the location of the logs :

- create a file called **logback.xml** in the main directory.

The below **logback.xml** file reports ERROR for the beacon controller and INFO logs for Flowscale:

```
<configuration>

<appender name="flowscale.log">
  <file>/var/flowscale/logs/flowscale.log</file>

  <rollingPolicy>
    <fileNamePattern>/var/flowscale/logs/flowscale.%d.log.gz</fileNamePattern>
    <maxHistory>60</maxHistory>
  </rollingPolicy>

  <encoder>
    <pattern>%d{dd-MM-yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{35} - %msg
%n</pattern>
  </encoder>
</appender>

<root level="info">
```

```

    <appender-ref ref="flowscale.log" />
</root>
<logger name="net.beaconcontroller" level="INFO"/>
<logger name="edu.iu.incntre.flowscale" level="INFO"/>
<logger name="edu.iu.incntre.flowscalehttplistener" level="INFO"/>
<logger name="edu.iu.incntre.flowscalestatcollector" level="INFO"/>
<logger name="edu.iu.incntre.flowscaleflowupdate" level="INFO"/>

</configuration>

```

In **flowscale-controller.ini** set the path for logback.xml

Dlogback.configurationFile=logback.xml

Run the Binary

- cd into the root directory (flowscale-controller-binary)
 cd flowscale-controller-binary
- run the flowscale binary
 ./flowscale-controller &
- Confirm that flowscale is running by running a netstat and verify that the jetty, beacon, and console ports are listening.
- Do a telnet to the OSGI console to confirm that the necessary bundles are in a running state.

```
telnet localhost 7777
```

```
telnet localhost 7777
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^J'.
```

```
osgi>
```

Type in ss to display the bundles

```
osgi> ss
```

Framework is launched.

```

id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.0.v20100517
Fragments=16, 32
2 ACTIVE org.mortbay.jetty.server_6.1.25
3 ACTIVE org.openflow_1.0.2
4 ACTIVE com.mysql.jdbc_1.0.0
6 ACTIVE org.eclipse.equinox.log_1.2.100.v20100503

```

```
7 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
9 ACTIVE org.springframework.asm_3.0.4.RELEASE
10 ACTIVE edu.iu.incntre.flowscaleflowupdate_1.0.0
11 ACTIVE org.mortbay.jetty.util_6.1.25
13 ACTIVE org.springframework.core_3.0.4.RELEASE
14 ACTIVE org.springframework.context.source_3.0.4.RELEASE
15 ACTIVE org.springframework.osgi.io_2.0.0.M1
16 RESOLVED org.eclipse.equinox.transforms.hook_1.0.200.v20100503
Master=0
17 ACTIVE org.springframework.context_3.0.4.RELEASE
18 ACTIVE org.json.simple_1.0.0
19 ACTIVE edu.iu.incntre.flowscale_1.0.0
20 ACTIVE ch.qos.logback.core_1.0.7
21 ACTIVE slf4j.api_1.7.2
22 ACTIVE com.springsource.net.sf.cglib_2.1.3
23 ACTIVE org.springframework.beans_3.0.4.RELEASE
24 ACTIVE org.apache.commons.io_2.0.0
25 ACTIVE org.springframework.osgi.extensions.annotations_2.0.0.M1
26 ACTIVE org.springframework.expression_3.0.4.RELEASE
27 ACTIVE ch.qos.logback.classic_1.0.7
Fragments=38
29 ACTIVE org.eclipse.equinox.ds_1.2.0.v20100507
31 ACTIVE org.springframework.osgi.extender_2.0.0.M1
32 RESOLVED org.eclipse.equinox.weaving.hook_1.0.0.v20100503
Master=0
33 ACTIVE jcl.over.slf4j_1.7.2
34 ACTIVE com.springsource.org.aopalliance_1.0.0
35 ACTIVE org.springframework.context.support_3.0.4.RELEASE
36 ACTIVE javax.servlet_2.5.0.v200910301333
37 ACTIVE net.beaconcontroller.core_1.0.2
38 RESOLVED net.beaconcontroller.logging.config_1.0.2
Master=27
39 ACTIVE edu.iu.incntre.flowscalehttplistener_0.1.0
40 ACTIVE org.springframework.osgi.core_2.0.0.M1
41 ACTIVE org.springframework.aop_3.0.4.RELEASE
42 ACTIVE net.beaconcontroller.packet_1.0.2
43 ACTIVE grnoc.net.util_1.0.0
44 ACTIVE org.eclipse.equinox.console.telnet_1.0.0.qualifier
45 ACTIVE org.eclipse.equinox.util_1.0.200.v20100503
46 ACTIVE edu.iu.incntre.flowscalestatcollector_1.0.0
```

Type disconnect to exit the console