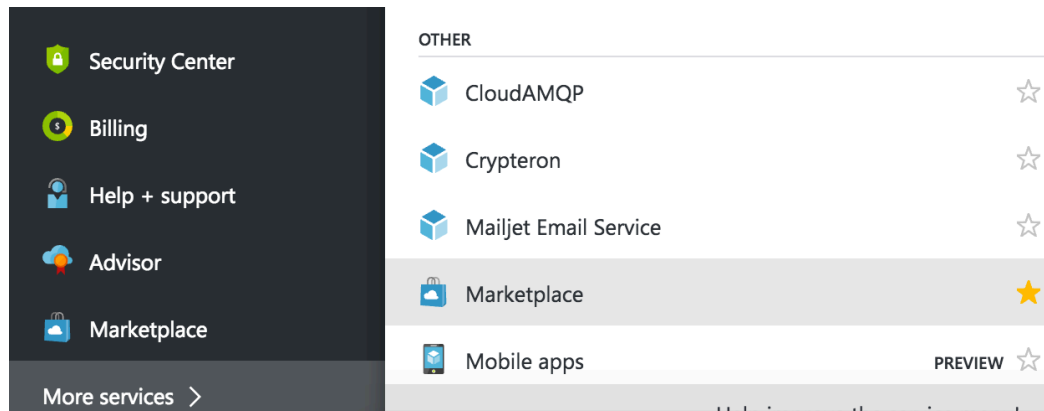








## Lab 1 – Setting up the Azure environment

1. Sign into the Microsoft Azure portal.
2. On the bottom left hand pane, select **More Services** and filter for **Marketplace**.



3. Search/Filter for **Hortonworks** and select **Hortonworks Sandbox with HDP 2.5** than click **Create** in the new pane that opens on the bottom right.

<input type="text" value="hortonworks"/>		
Results		
NAME	PUBLISHER	CATEGORY
 Hortonworks Sandbox with HDP 2.4 (Staged)	Hortonworks	Compute
 Hortonworks Sandbox with HDP 2.5 (Staged)	Hortonworks	Compute
 Hortonworks Data Platform Standard (Staged)	Hortonworks	Compute
 Hortonworks Sandbox with HDP 2.5	Hortonworks	Compute
 Cloudbreak for Hortonworks Data Platform	Hortonworks	Compute
 Hortonworks Sandbox with HDP 2.4	Hortonworks	Compute

4. Follow instructions and fill in values as directed by the Wizard.
  - a. Basics  
Use defaults for all unless otherwise specified below:
    - Name
    - User name
    - Authentication type: Password

- Password
- Confirm password
- Resource group: *new or existing*
- Location

Basics

\* Name

hdf-workshop

✓

VM disk type ⓘ

SSD

▼

\* User name

ppruski

✓

\* Authentication type

SSH public key Password

\* Password

.....

✓

\* Confirm password

.....

✓

Subscription

SE

▼

\* Resource group ⓘ

☒ Create new ☐ Use existing

hdf-workshop

✓






Location

Canada Central

▼

OK

- b. For an appropriate sized environment specify DS3\_V2. (Note: if the size is unavailable an alternate location can be used)


DS3_V2 Standard	
4	Cores
14	GB
 8	Data disks
 12800	Max IOPS
 28 GB	Local SSD
	Load balancing
	Premium disk support
227.66	
USD/MONTH (ESTIMATED)	

#### c. Settings

Use defaults for all unless otherwise specified below:

- Virtual Network: *Select existing or create new*
  - *If new:*

Create virtual network



The address space '10.10.0.0/16' overlaps with '10.10.10.0/24' in virtual network 'resgroup-vnet'.

\* Name

hdf-workshop

\* Address space

10.10.0.0/16

✓

10.10.0.0 - 10.10.255.255 (65536 addresses)

\* Subnet name

default

\* Subnet address range

10.10.0.0/16

✓

10.10.0.0 - 10.10.255.255 (65536 addresses)

#### d. Summary

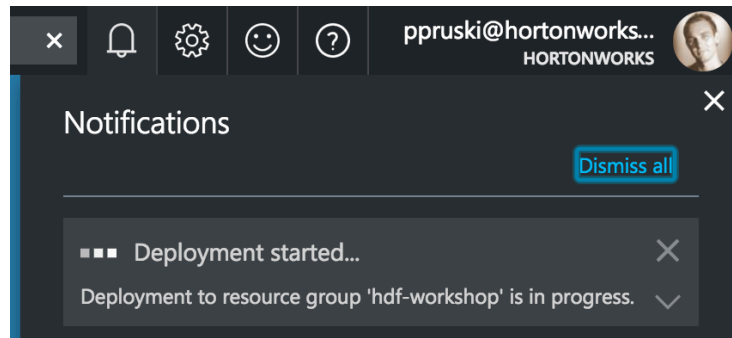
After reviewing, click **OK**.

s

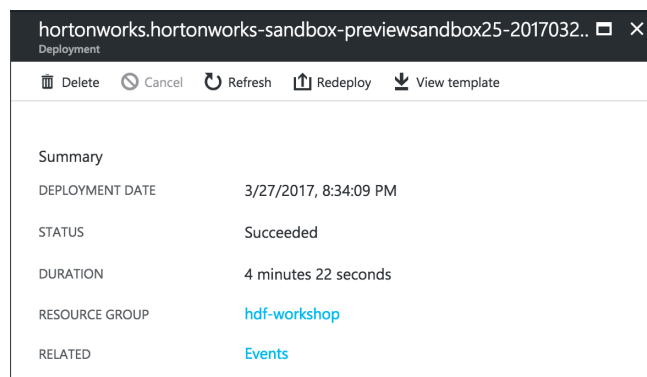
#### e. Buy

After reviewing, click **Purchase**.

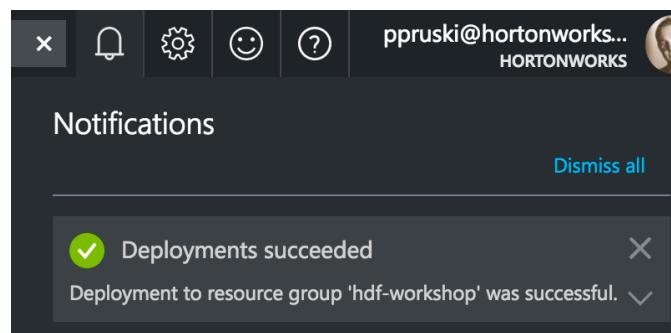
5. You can view the progress of the deployment by clicking the notification icon on the top right of the portal and selecting the deployment.



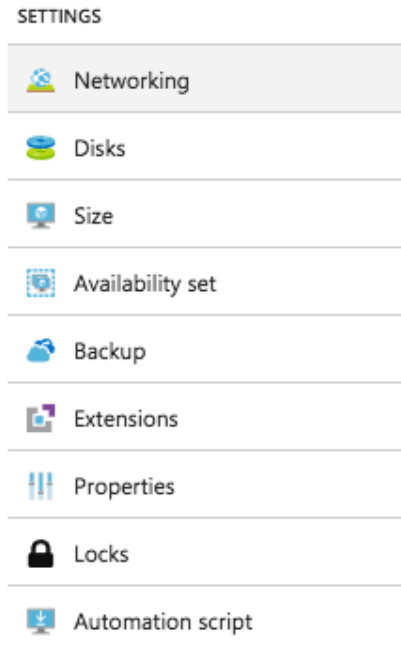
- The deployment can take a few minutes. Once it is complete, you should see a screen similar to the following:



- Navigate to the Sandbox virtual machine by clicking **Deployments succeeded** from the notifications once again.



- Click on **Networking** from the Virtual machine navigation pane under SETTINGS and select the network interface for the Sandbox.



9. Click on the **Add Inbound port rule** button on the right side of the screen and **Add** an inbound rule to allow all ports as the following figure shows and click **OK**.

A screenshot of the 'Add inbound security rule' form in the Azure portal. The form is titled 'Add inbound security rule' and has a subtitle 'hdf-workshop-nsg'. It is set to 'Advanced' mode. The form contains several fields and options: 'Name' is 'allow-all-ports'; 'Priority' is '1010'; 'Source' is 'Any' (selected from a dropdown with 'CIDR block' and 'Tag' as other options); 'Service' is 'Custom'; 'Protocol' is 'Any' (selected from a dropdown with 'TCP' and 'UDP' as other options); 'Port range' is '1-65535'; and 'Action' is 'Allow' (selected from a dropdown with 'Deny' as the other option).

Add inbound security rule  
hdf-workshop-nsg

✕ Advanced

\* Name  
allow-all-ports

\* Priority ⓘ  
1010

\* Source ⓘ  
Any CIDR block Tag

Service ⓘ  
Custom

\* Protocol  
Any TCP UDP

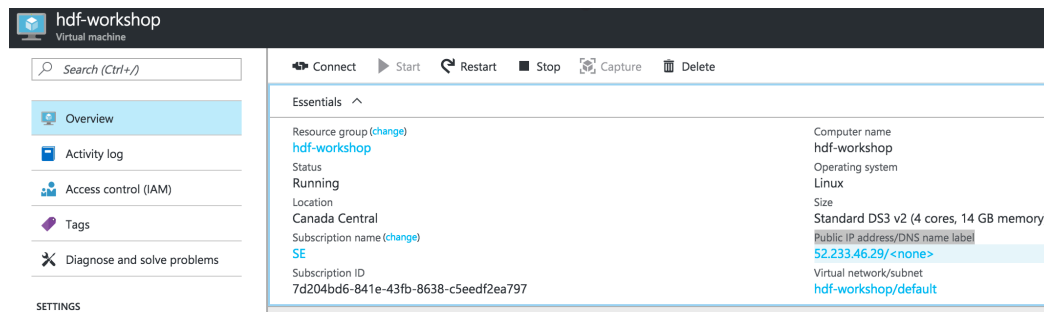
\* Port range ⓘ  
1-65535

\* Action  
Deny Allow

10. Using a terminal (Mac/Linux) or Putty (Windows), SSH into the VM with the public IP address of the Sandbox and the username/password you set during Sandbox creation.

```
ssh <user_name>@<public_ip_addr>
```

You can find the public IP address of the virtual machine in the Virtual Machine Overview section.



11. Change user to sudo and enter your password once again.

```
sudo su -
```

12. Now as the root user SSH into the docker image with the following command.

```
ssh root@sandbox -p 2222
```

13. You will be prompted for the password and to set a new password. The password for root is **hadoop**. You'll be required to enter this password twice before selecting a new password. (HINT: for simplicity it is recommended to change the password to the same as specified above in step 4a)

14. Set the new passwords for Ambari as well. (HINT: for simplicity it is recommended to change the password to the same as specified above in step 4a)

```
ambari-admin-password-reset
```

This will cause ambari-server to restart.

Install NiFi 1.1.0, MiNiFi, and MiNiFi-toolkit:

15. Download the tarball's for the appropriate OS (CentOS 6). The below script can be copied and pasted into your terminal.

```
wget http://public-repo-1.hortonworks.com/HDF/centos6/2.x/updates/2.1.2.0/HDF-2.1.2.0-centos6-tars-tarball.tar.gz

tar zxvf HDF-2.1.2.0-centos6-tars-tarball.tar.gz

cp ./HDF/centos6/tars/nifi/nifi-1.1.0.2.1.2.0-10-bin.tar.gz /usr/

cp ./HDF/centos6/tars/minifi/minifi-1.0.2.1.2.0-10-bin.tar.gz /usr/

cp ./HDF/centos6/tars/minifi/minifi-toolkit-1.0.2.1.2.0-10-bin.tar.gz /usr/

cd /usr/

tar zxvf nifi-1.1.0.2.1.2.0-10-bin.tar.gz

tar zxvf minifi-1.0.2.1.2.0-10-bin.tar.gz

tar zxvf minifi-toolkit-1.0.2.1.2.0-10-bin.tar.gz

cd nifi-1.1.0.2.1.2.0-10

bin/nifi.sh install
```

16. Change the NiFi web properties so that the port is set to 9090 (versus the default of 8080 which is being used by Ambari).

```
vi conf/nifi.properties
```

Change the value for **nifi.web.http.port** from 8080 to **9090**.

17. Start the NiFi service with the following command.

```
service nifi start
```

18. Point your web browser to the NiFi UI console. The address is:  
[http://<public\\_IP\\_addr>:9090/nifi/](http://<public_IP_addr>:9090/nifi/)

## Lab 2 – Consuming Data

### Review use case

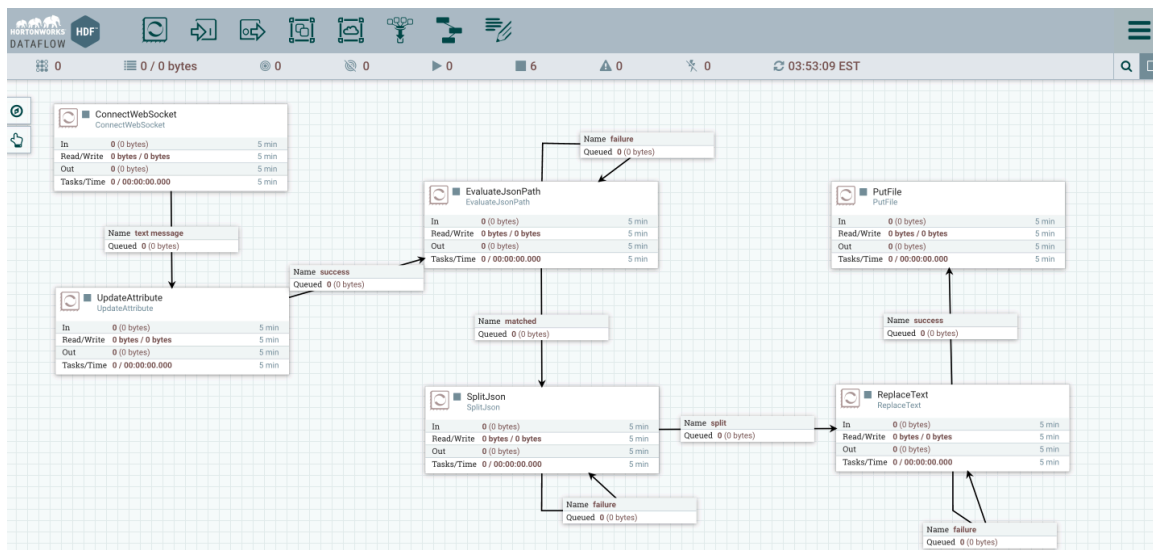
Use case: We work for a social analytics company, the first product we are working on is a Meetup Analytics dashboard. To do this we want to calculate the Top N

meetups happening right now and display them on a dashboard. To get started we need to consume the data from the Meetup RSVP stream, extract what we need, spilt the content and save it to a file:

### Goals:

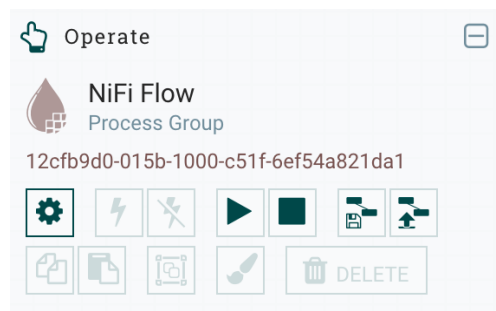
- Consume Meetup RSVP stream
- Extract the JSON elements we are interested in
- Split the JSON into smaller fragments
- Write the JSON files to disk

Our final flow for this lab will look like the following:



**Note:** To use the template versus manually performing all the actions below see the instructions in the Appendix of this document!

1. With a blank canvas, click on the Configuration gear icon in the Operate box on the left side of the UI.



2. Under the CONTROLLER SERVICES tab, Add a JettyWebSocketClient service and click on the pencil icon to edit the configure the controller service.



GENERAL

CONTROLLER SERVICES

- Under the PROPERTIES tab add the value for WebSocket URI as ws://stream.meetup.com/2/rsvps. Your configuration should look like this:

Configure Controller Service

SETTINGS

PROPERTIES

COMMENTS

Required field

Property	Value
Input Buffer Size	4 kb
Max Text Message Size	64 kb
Max Binary Message Size	64 kb
WebSocket URI	ws://stream.meetup.com/2/rsvps
SSL Context Service	No value set
Connection Timeout	3 sec

- Notice that the state for the Controller Service is Disabled. Click on the lightning icon on the right to enable it.

Name	Type	State	Process Group
JettyWebSocketClient	JettyWebSocketClient	Disabled	12c9b9d0-015b-1000-c51f-6ef54a821da1

- Add a ConnectWebSocket processor to the canvas by dragging the icon on the page . It will need to be configured as type "ConnectWebSocket".

Add Processor

Tag Cloud:

Displaying 1 of 188

conn|

Type	Tags
ConnectWebSocket	subscribe, consume, listen, WebSocket

- Configure the ConnectWebSocket Processor so it looks like below.
  - Under the properties tab set the WebSocket Client Controller Service
  - Set the WebSocket Client ID to AGP-HDF-WS-TEST
  - Set the automatic termination relationships as shown.

Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property	Value
WebSocket Client ControllerService	JettyWebSocketClient →
WebSocket Client Id	AGP-HDF-WS-TEST

Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Name

ConnectWebSocket

Enabled

Id

1614dba3-015b-1000-69a3-ecce9290d99d

Type

ConnectWebSocket

Penalty Duration

30 sec

Yield Duration

1 sec

Bulletin Level

WARN

Automatically Terminate Relationships

☒ binary message  
The WebSocket binary message output

☒ connected  
The WebSocket session is established

☐ text message  
The WebSocket text message output

## 7. Add an UpdateAttribute Processor

Add Processor

Tag Cloud:

Displaying 1 of 188

UpdateA|

amazon attributes

avro aws

consume database

fetch files

filesystem get

hadoop http

ingest input

insert json listen

logs message

nut remote

Type

UpdateAttribute

Tags

Attribute Expression Language, update, attribu...

Selected Processor:

UpdateAttribute

Updates the Attributes for a FlowFile by using the Attribute Expression Language and/or

- Configure it to have a custom property called mime.type with the value of application/json.

Configure Processor

SETTINGS

SCHEDULING

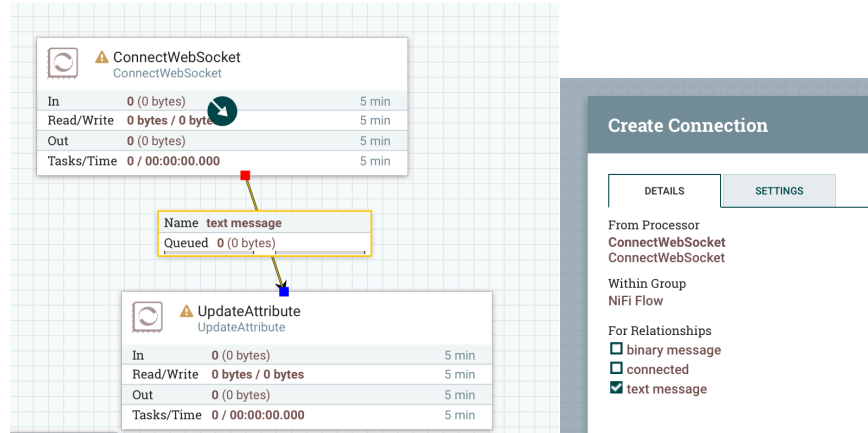
PROPERTIES

COMMENTS

Required field

Property	Value
Delete Attributes Expression	No value set
mime.type	application/json

- Join ConnectWebSocket Processor and the UpdateAttribute Processor using a text message for relationships.



9. Add an EvaluateJsonPath processor and configure it as shown below

### Configure Processor

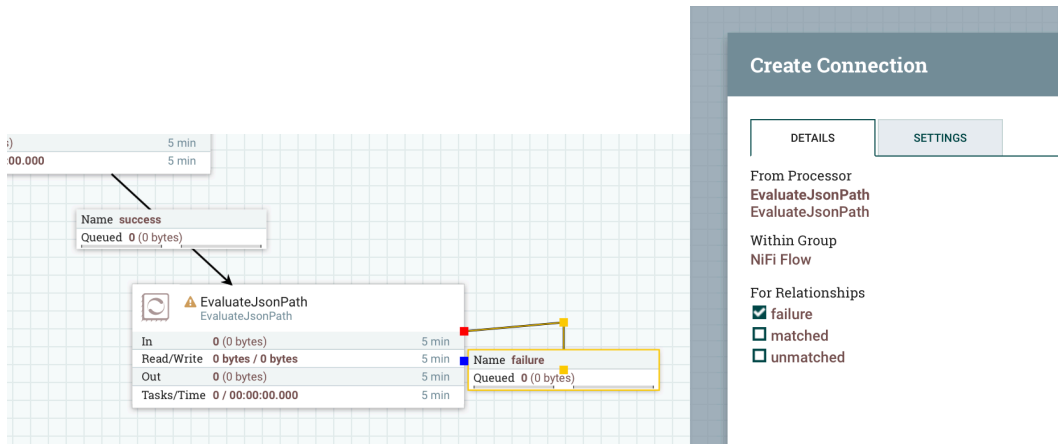
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS

Required field +

Property	Value
Destination	flowfile-attribute
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
event.name	\$.event.event_name
event.url	\$.event.event_url
group.city	\$.group.group_city
group.country	\$.group.group_country
group.name	\$.group.group_name
group.state	\$.group.group_state
venue.lat	\$.venue.lat
venue.lon	\$.venue.lon
venue.name	\$.venue.venue_name

CANCEL
APPLY

10. Join the UpdateAttribute processor and EvaluateJsonPath processor. Also add a failure relationship (Note: recursive join)



11. Add a SplitJson processor and configure the JsonPath Expression to be `$.group.group_topics`. Also the Original relationship needs to be automatically terminated. Your configuration should look like below.

**Add Processor**

Tag Cloud: **amazon** **attributes** **avro** **aws** **consume** **database** **fetch** **files** **filesystem** **get** **hadoop** **http**

Displaying 1 of 188

Type	Tags
Split.Json	split, jsonpath, json

**Configure Processor**

SETTINGS SCHEDULING PROPERTIES COMMENTS

Name: Split.Json ☒ Enabled

Id: 01581004-1e10-14fa-16f7-c8bdfde85fdf

Type: Split.Json

Penalty Duration: 30 sec

Yield Duration: 1 sec

Automatically Terminate Relationships

☐ failure  
If a FlowFile fails processing for any reason (for example, the FlowFile is not valid JSON or the specified path does not exist), it will be routed to this relationship

☒ original  
The original FlowFile that was split into segments. If the FlowFile fails processing, nothing will be sent to this relationship

☐ split  
All segments of the original FlowFile will be routed to this relationship

**Configure Processor**

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field

Property	Value
JsonPath Expression	<code>\$.group.group_topics</code>
Null Value Representation	empty string

12. Join the EvaluateJsonPath processor and the SplitJson processor. In addition, create a failure recursive join on the SplitJson Processor. Should look like the below.

### Configure Connection

DETAILS

SETTINGS

From Processor  
EvaluateJsonPath  
EvaluateJsonPath

Within Group  
NIFI Flow

For Relationships  
☐ failure  
☒ matched  
☐ unmatched

### Create Connection

DETAILS

SETTINGS

From Processor  
SplitJson  
SplitJson

Within Group  
NIFI Flow

For Relationships  
☒ failure  
☐ original  
☐ split

**SplitJson**  
 SplitJson

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name failure  
 Queued 0 (0 bytes)

Name matched  
 Queued 0 (0 bytes)

### 13. Add a ReplaceText processor

Displaying 2 of 188

replace

attributes	Type	Tags
ReplaceText	ReplaceText	Regular Expression, Replace, Regex, Text, Modi...
ReplaceTextWithMapping	ReplaceTextWithMapping	Regular Expression, Replace, Regex, Text, Modi...

- a. In the properties tab configure the Search Value to be `{{[]}}([\\S\\s+)]{{[]}}` and the Replacement Value to be:

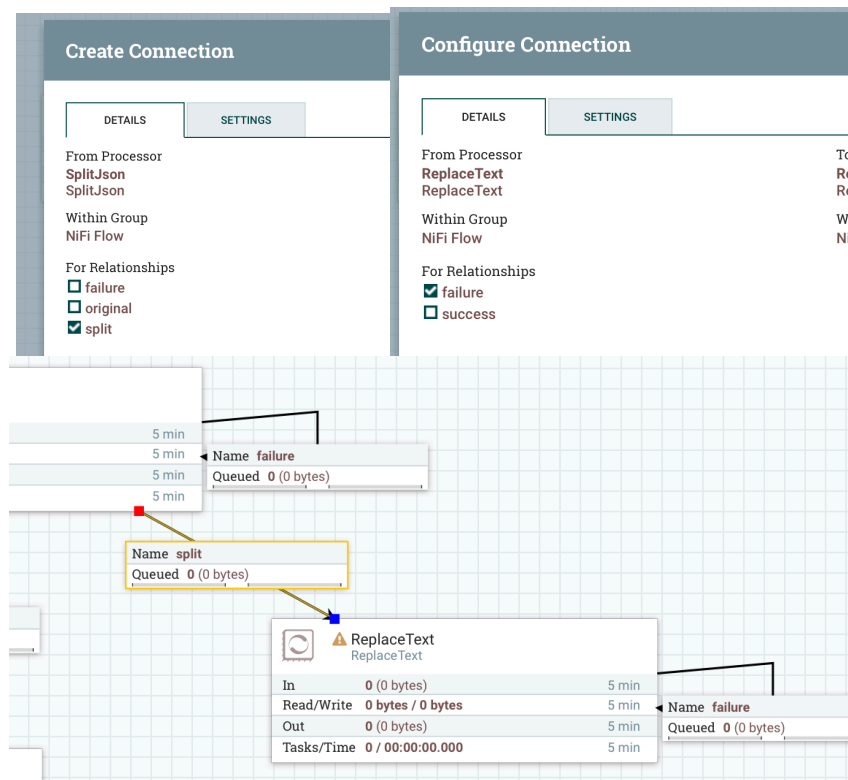
```
{
  "event_name": "${event.name}",
  "event_url": "${event.url}",
  "venue" : {
    "lat": "${venue.lat}",
    "lon": "${venue.lon}",
    "name": "${venue.name}"
  },
  "group" : {
    "group_city" : "${group.city}",
    "group_country" : "${group.country}",
    "group_name" : "${group.name}",
    "group_state" : "${group.state}",
  }
}
```

```
$2  
}  
}
```

The processor should look like the below.

Property	Value
Search Value	{0}(\Ss)+{0}
Replacement Value	{ "event_name": "\${event.name}", "event_url": "\${e...
Character Set	UTF-8
Maximum Buffer Size	1 MB
Replacement Strategy	Regex Replace
Evaluation Mode	Entire text

14. Join the SplitJson processor and the ReplaceText processor. In addition add an on Failure recursive join.



15. Add a PutFile processor to the canvas and configure it to write the data out to /temp/rsvp-data. Automatically terminate both on Success and Failure. The configuration should look like below.

### Add Processor

Tag Cloud: amazon attributes avro aws consume database fetch files Displaying 1 of 188

Type	Tags
PutFile	restricted, files, archive, copy, put, local, filesystem...

### Configure Processor

SETTINGS SCHEDULING PROPERTIES COMMENTS

Name: PutFile ☒ Enabled

Id: 0158100a-1e10-14fa-767c-7998db75bd3a

Type: PutFile

Penalty Duration: Yield Duration:

Automatically Terminate Relationships ☒ failure

Files that could not be written to the output directory for some reason are transferred to this relationship

☒ success

Files that have been successfully written to the output directory are transferred to this relationship

### Configure Processor

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field

Property	Value
Directory	/temp/rsvp
Conflict Resolution Strategy	fail
Create Missing Directories	true
Maximum File Count	No value set
Last Modified Time	No value set
Permissions	No value set

16. Join the ReplaceText processor and the PutFile processor for successful relationships.

bytes) 5 min

10:00:00.000 5 min

Name failure

Queued 0 (0 bytes)

Name success

Queued 0 (0 bytes)

PutFile

PutFile

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

### Create Connection

DETAILS SETTINGS

From Processor: ReplaceText

Within Group: NiFi Flow

For Relationships: ☐ failure ☒ success

## Questions to Answer

- What does a full RSVP Json object look like?
- How many output files do you end up with?
- How can you change the file name that Json is saved as from PutFile?

- d. Why do you think we are splitting out the RSVP's by group?
- e. Why are we using the Update Attribute processor to add a mime.type?
- f. How can you change the flow to get the member photo from the Json and download it.

## Lab 3 – Getting Started with MiNiFi

### Getting Started with MiNiFi

In this lab, we will learn how configure MiNiFi to send data to NiFi:

- Setting up the Flow for NiFi
- Setting up the Flow for MiNiFi
- Preparing the flow for MiNiFi
- Configuring and starting MiNiFi
- Enjoying the data flow!

**NOTE:** Before starting NiFi we need to enable Site-to-Site communication. To do that we need to edit the config file. Make the following changes:

Open `/usr/nifi-1.1.0.2.1.2.0-10/conf/nifi.properties` in your favorite editor:

Change:

```
nifi.remote.input.host=  
nifi.remote.input.socket.port=
```

To:

```
nifi.remote.input.host=localhost  
nifi.remote.input.socket.port=10001
```


Restart the NiFi service with the following command.

```
# service nifi restart
```

Now that we have NiFi up and running and MiNiFi installed and ready to go, the next thing to do is to create our data flow. To do that we are going to first start with creating the flow in NiFi.

1. Point your web browser to the NiFi UI console. The address is:  
<http://<public IP addr>:9090/nifi/>



2. The first thing we are going to do is setup an Input Port. This is the port that MiNiFi will be sending data to. To do this drag the Input Port icon  to the canvas and call it "From MiNiFi".


**Add Port**

Input Port Name

From MiNiFi

CANCEL

ADD

3. Now that the Input Port is configured we need to have somewhere for the data to go once we receive it. In this case we will keep it very simple and just log the attributes. To do this drag the Processor icon  to the canvas and choose the LogAttribute processor.

**Add Processor**

Tag Cloud: Displaying 1 of 188 LogAttribute

Type	Tags
LogAttribute	logging, attributes

amazon attributes

avro aws

consume

database fetch

files filesystem

get hadoop http

ingest input

insert json listen

logs message

put remote

restricted source

split update

Selected Processor:  
LogAttribute  
No description specified

CANCEL

ADD

- a. On the Settings tab, under Auto-terminate relationships, select the checkbox next to Success. This will terminate FlowFiles after this processor has successfully processed them.

### Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Name

LogAttribute

Enabled

Automatically Terminate Relationships

success

All FlowFiles are routed to this relationship

Id

13183bd4-015b-1000-d27c-61e4247d402e

- b. Also on the Settings tab, set the Bulletin level to Info. This way, when the dataflow is running, this processor will display the bulletin icon (see [Anatomy of a Processor](#)), and the user may hover over it with the mouse to see the attributes that the processor is logging.

### Configure Processor

SETTINGS

SCHEDULING

Name

LogAttribute

Id

13183bd4-015b-1000-d27c-61e4247d402e

Type

LogAttribute

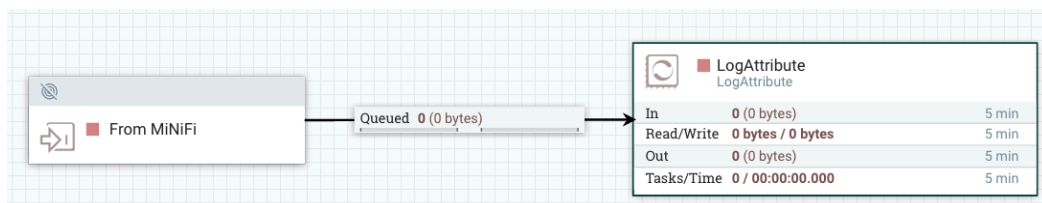
Penalty Duration

30 sec

Bulletin Level

INFO

5. Now that we have the input port and the processor to handle our data, we need to connect them. After creating the connection your data flow should look like this:



6. We are now ready to build the MiNiFi side of the flow. To do this, do the following:
- o Add a GenerateFlowFile processor to the canvas.
    - a) On the Scheduling tab, set Run schedule to: 5 sec. Note that the GenerateFlowFile processor can create many FlowFiles very quickly; that's why setting the Run schedule is important so that this flow does not overwhelm the system NiFi is running on.

SETTINGS

SCHEDULING

PROPERTIES

CO

Scheduling Strategy ?

Timer driven ▼

Concurrent Tasks ?

1

Run Schedule ?

5 sec

b) On the Properties tab, set File Size to: 10 kb

SETTINGS

SCHEDULING

PROPERTIES

Required field

Property	Value
File Size	10 kb

- Drag and drop the Remote Processor Group icon  to the canvas and for the URL use **http://localhost:9090/nifi**.

7. Connect the GenerateFlowFile to the Remote Process Group as shown below. (You may have to refresh the Remote Processor Group, before the input port will be available)

Create Connection

DETAILS

SETTINGS

From Processor

GenerateFlowFile

GenerateFlowFile

Within Group

NiFi Flow

For Relationships

☒ success

To Input

From MiNiFi ▼

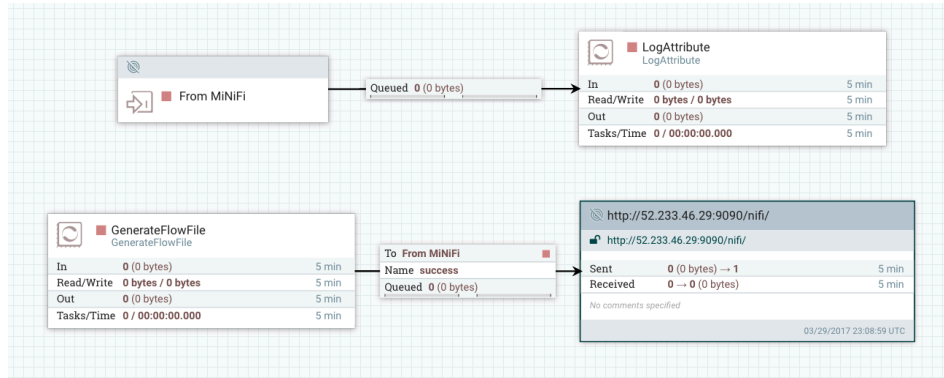
Within Group

http://52.233.46.29:9090/nifi/

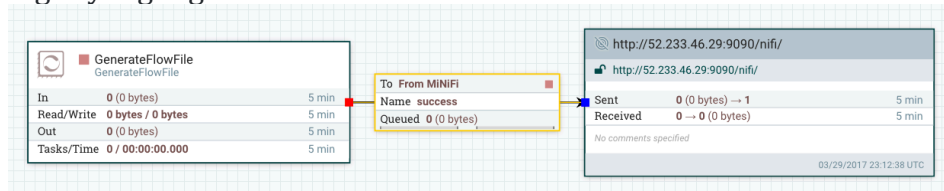
CANCEL


ADD

- Your canvas should now look similar to what is shown below.



8. The next step is to generate the flow we need for MiNiFi. To do this do the following steps:
  - Select the GenerateFlowFile processor and the NiFi Flow Remote Processor Group as well as the connection between. (Note you can select multiple entities with the shift key). The entities should be slightly highlighted.



- Create a template for MiNiFi by clicking the create template icon  within the Operate box on the left hand side of the UI. Specify MiNiFi Flow for the name.

Create Template

Name

MiNiFi Flow

Description

CANCEL

CREATE

9. Now we need to download the template. We will do this locally within the VM by executing the following command:

```
OUTPUT="$(grep Template `ls -t /usr/nifi-1.1.0.2.1.2.0-10/logs/nifi-app* | head -1` | tail -1 | grep -o -P '(?<=id=).*?(?=\])' )"
```

```
wget http://localhost:9090/nifi-api/templates/`echo "${OUTPUT}`"/download -O MiNiFi_Flow.xml
```

10. We are now ready to setup MiNiFi. However, before doing that we need to convert the template to YAML format which MiNiFi uses. To do this we need to do the following:

- Transform the template that we downloaded using the following command:

```
/usr/minifi-toolkit-1.0.2.1.2.0-10/bin/config.sh  
transform MiNiFi_Flow.xml config.yml
```

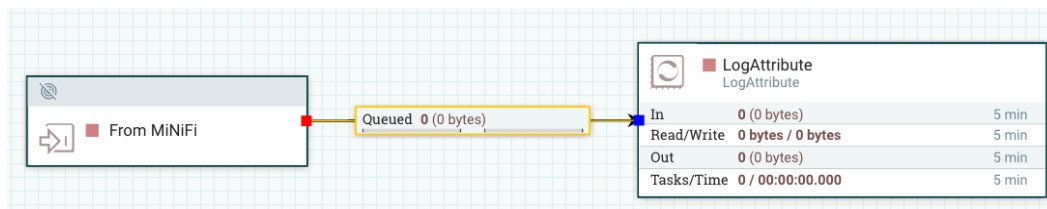
11. Next copy the config.yml to the minifi conf directory. That is the file that MiNiFi uses to generate the nifi.properties file and the flow.xml.gz for MiNiFi.


```
mv config.yml /usr/minifi-1.0.2.1.2.0-10/conf/
```

12. That is it. We are now ready to start MiNiFi. To start MiNiFi from a command prompt execute the following:

```
cd /usr/minifi-1.0.2.1.2.0-10/  
bin/minifi.sh start
```

13. From the NiFi UI, select the Input Port, the LogAttribute processor, and the connection between them. Hold shift to select multiple entities.



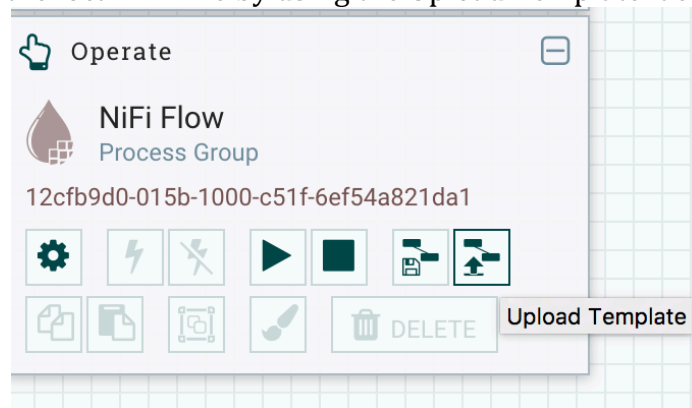
14. Click the start icon  within the Operate box on the left hand side of the UI to begin the flow.

You should be able to now go to your NiFi flow and see data coming in from MiNiFi.

## Appendix

### Lab 2 Import Template

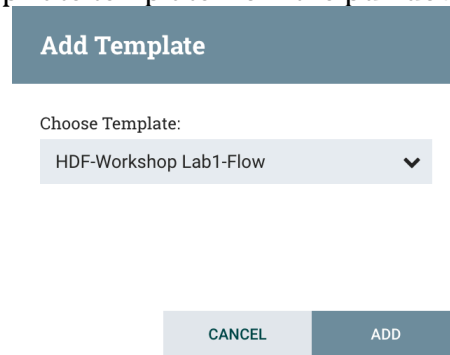
1. To use the template that is provided on github for the workflow (versus manually performing all the actions) you'll first need to download the XML template from GitHub and then upload it to NiFi.
  - a. The template can be found here ([https://raw.githubusercontent.com/apsaltis/HDF-Workshop/master/templates/HDF-Workshop\\_Lab1-Flow.xml](https://raw.githubusercontent.com/apsaltis/HDF-Workshop/master/templates/HDF-Workshop_Lab1-Flow.xml)). Save the xml as a local file.
  - b. Upload the local xml file by using the Upload Template facility



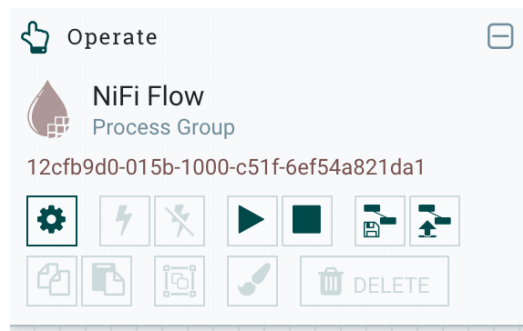
2. Next you'll need to drag and drop the template onto the canvas and select the one you just uploaded.
  - a. Grab the below icon from the top of the window and drag it on the canvas



- b. Select the appropriate template from the pull down selection.



3. Once the template has been added, click on the Configuration gear icon in the Operate box on the left side of the UI.



- Under the CONTROLLER SERVICES tab, click on the pencil icon to edit the configure the controller service.

GENERAL

CONTROLLER SERVICES

</

- Under the PROPERTIES tab copy the value for URI for the last property, WebSocket URI, and paste it for the empty value for WebSocket URI in bold. You can then delete the last property in the list for WebSocket URI. Your configuration should look like this:

### Configure Controller Service

SETTINGS

PROPERTIES

COMMENTS

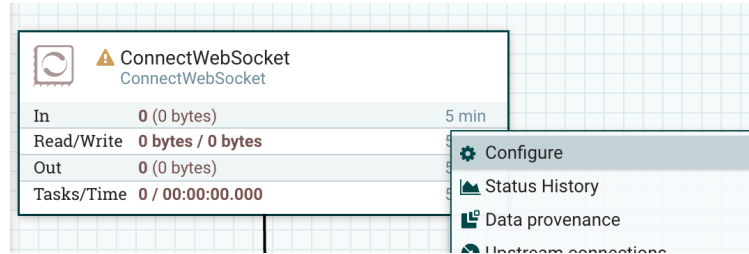
Required field

Property	Value
Input Buffer Size	4 kb
Max Text Message Size	64 kb
Max Binary Message Size	64 kb
WebSocket URI	<b>ws://stream.meetup.com/2/rsvps</b>
SSL Context Service	No value set
Connection Timeout	3 sec

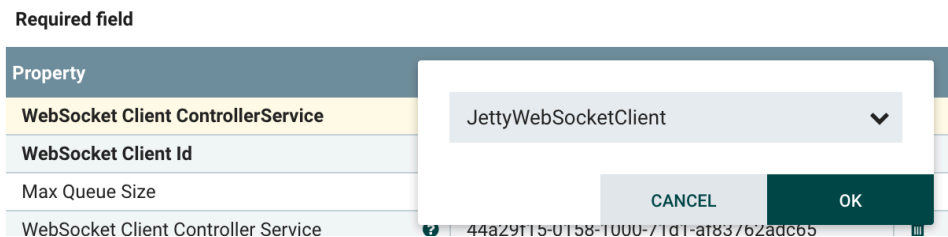
- Notice that the state for the Controller Service is Disabled. Click on the lightning icon on the right to enable it.

Name	Type	State	Process Group
JettyWebSocketClient	JettyWebSocketClient	Disabled	12cfb9d0-015b-1000-c51f-6ef54a821da1

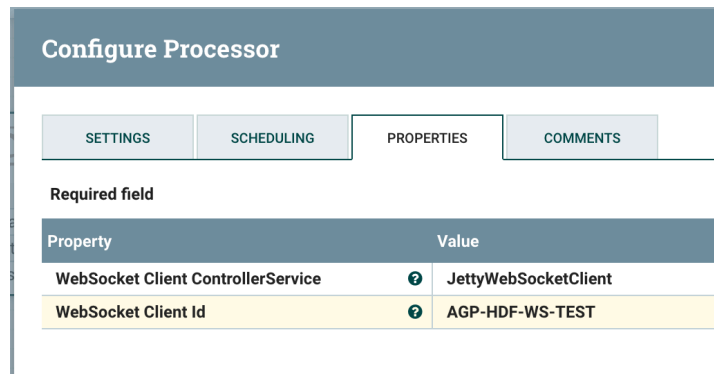
- Right-click on the ConnectWebSocket processor and select Configure.



8. Under the PROPERTIES tab click on the value for first WebSocketClientService property at the top of the list and select JettyWebSocketClient as shown.






9. Under the PROPERTIES tab copy the value for URI for the last property, WebSocket Client Id, and paste it for the empty value for WebSocket Client Id in bold. You can then delete all the other properties within this. Your configuration should look like this:










10. Now you should be able to run the template and view the output in /temp/rsvp-data/. Select the back canvas and hit the start button. The label should say "NiFi Flow – Process Group". If you have an individual component selected, then the play button will only run that component and the label will reflect the component selected.









 Operate 

 **NiFi Flow**  
Process Group

1575d8d2-015b-1000-c2fb-7ee47c8cc7a0



 DELETE

 Up  
Up

In (

Read/Write (

Out (

Tasks/Time (