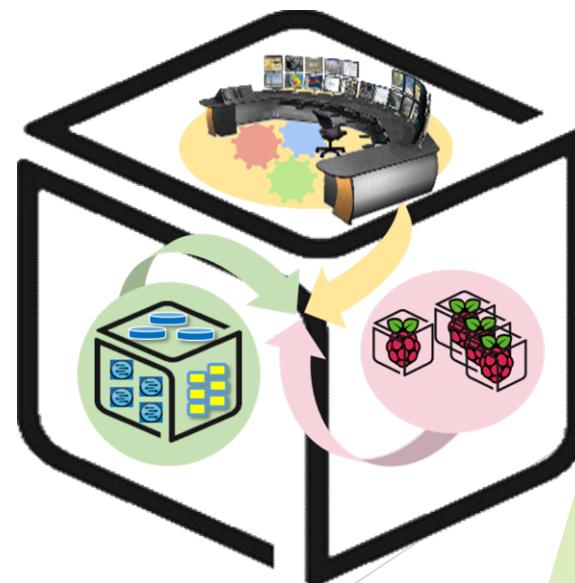


# SmartX Labs for Computer Systems

InterConnect  
Lab v01

(2016, Spring)

NetCS Lab



# InterConnect LAB

## -Goals-

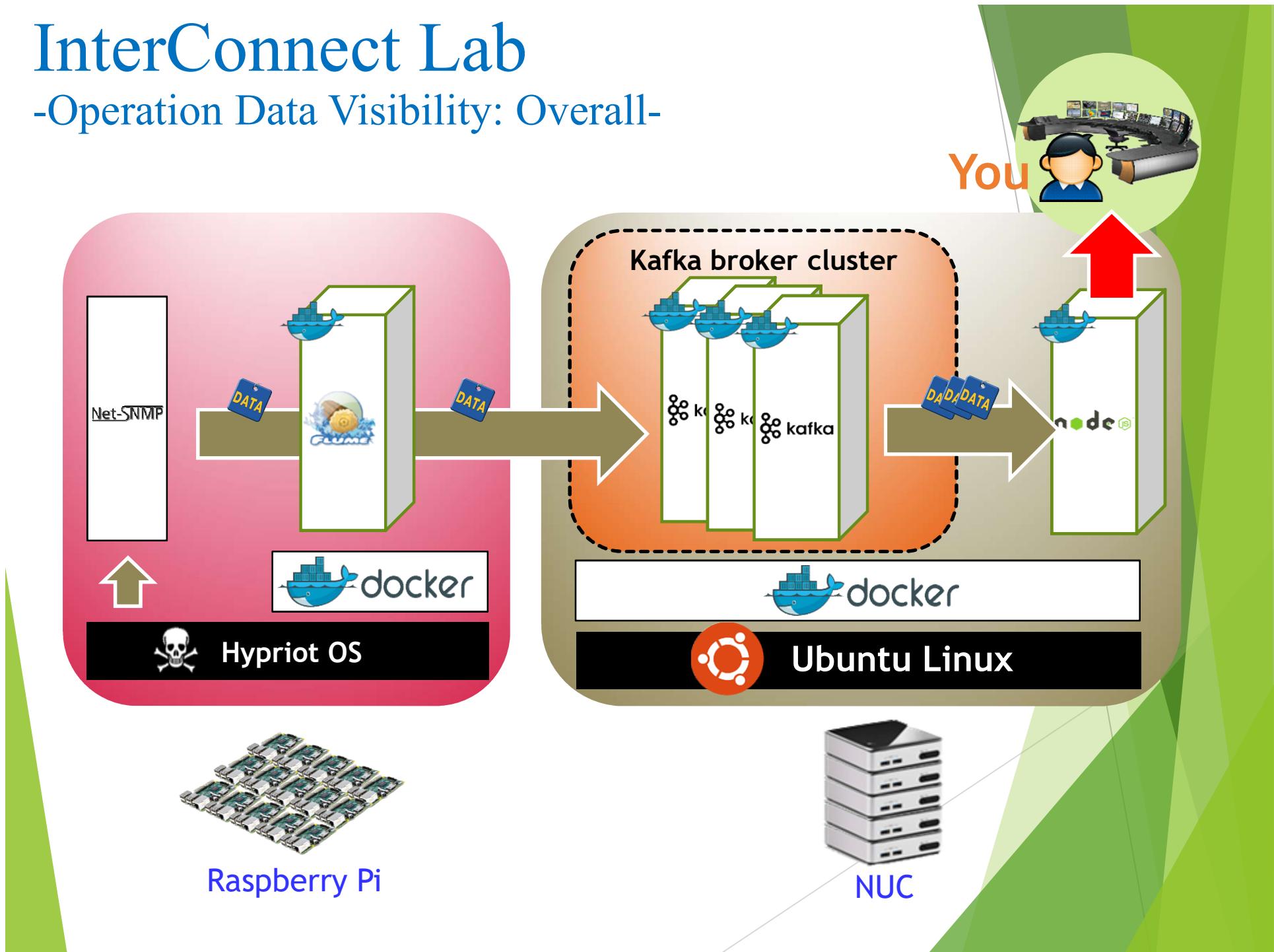
- ▶ Understanding Concepts
  - ▶ Net-SNMP, Flume, Kafka, Hypriot OS

- ▶ Connecting with each functions
  - ▶ With Raspberry Pi and NUC

- ▶ Service Realization
  - ▶ Operation Data Visibility

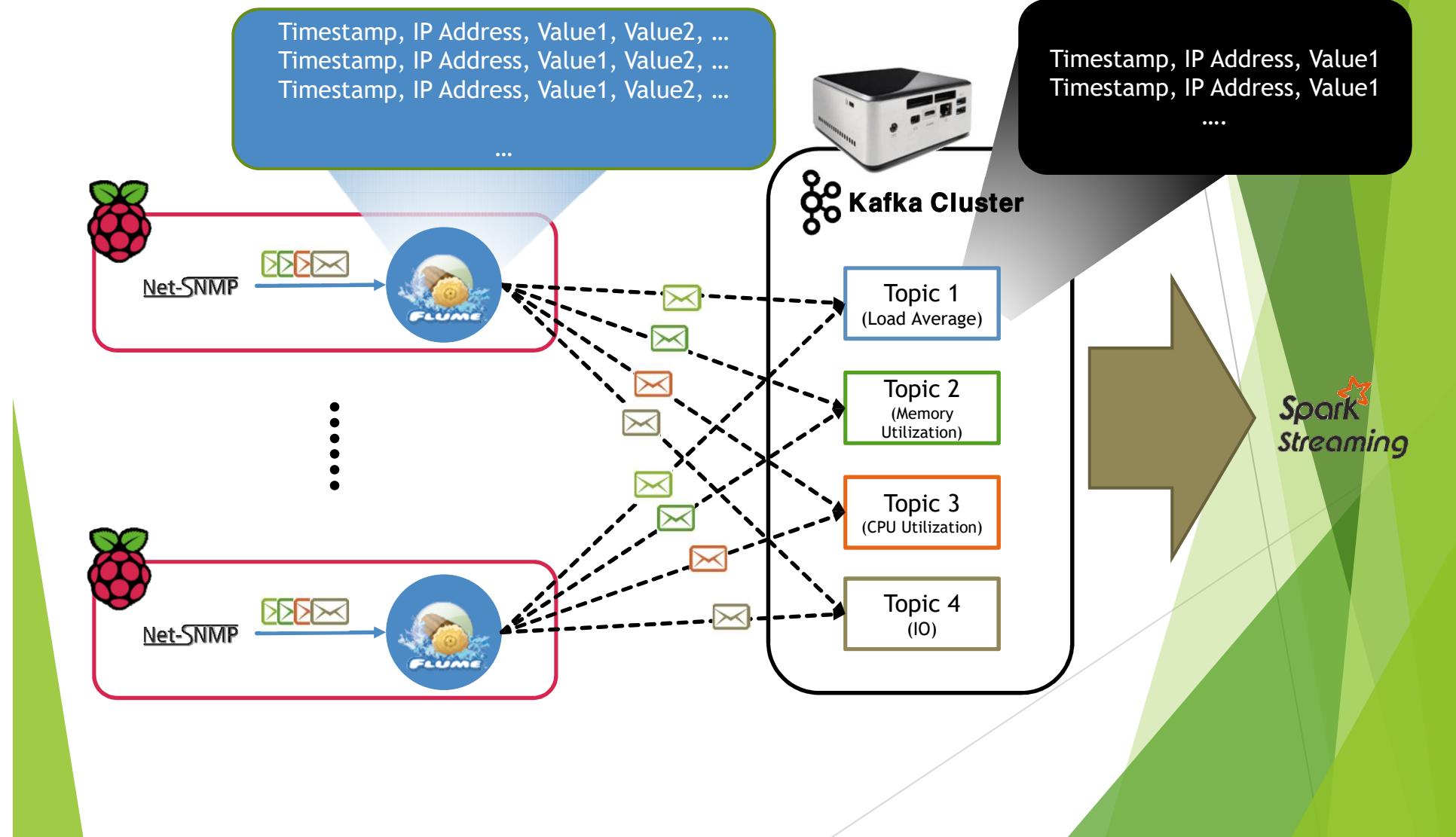
# InterConnect Lab

## -Operation Data Visibility: Overall-



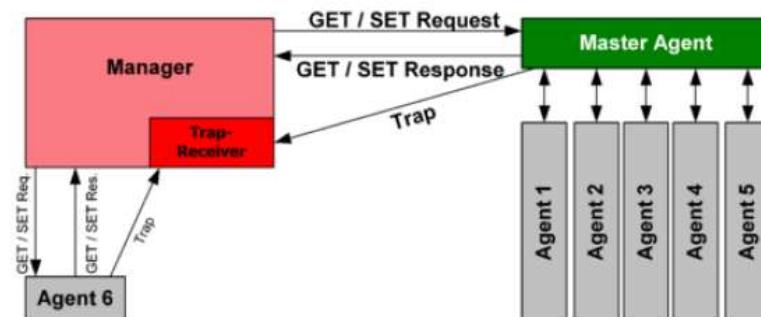
# InterConnect Lab

## -Operation Data Visibility: Data flow-



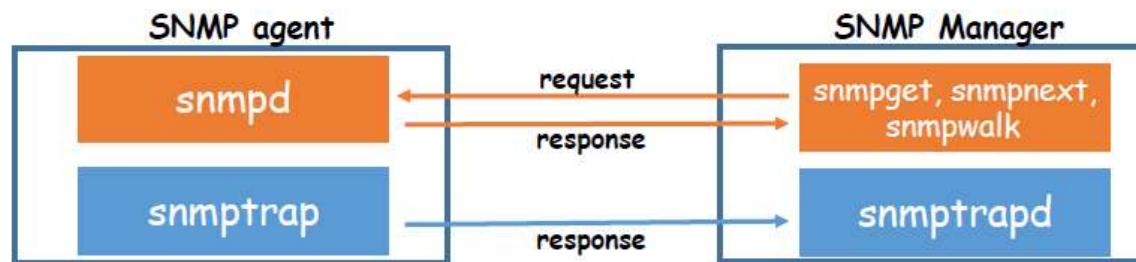
# SNMP

- SNMP: Simple Network Management Protocol
- Used in network management systems  
to monitor network-attached devices
- Include routers, switches, servers,  
workstations, printers, modem racks and more.



## Net-SNMP Net-SNMP

- A Suite of **software** for using and deploying the SNMP protocol

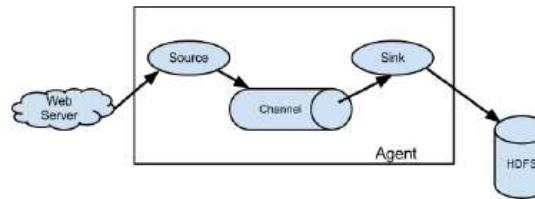


# Flume



- **Log aggregator**

- Many customizable data sources
- Flume can be used to handle them.
- Run asynchronously



- Flume Agent

- Source

- Consuming events having a specific format.
- Delivering it to the channel

- Channel

- Holding the event until that consumed

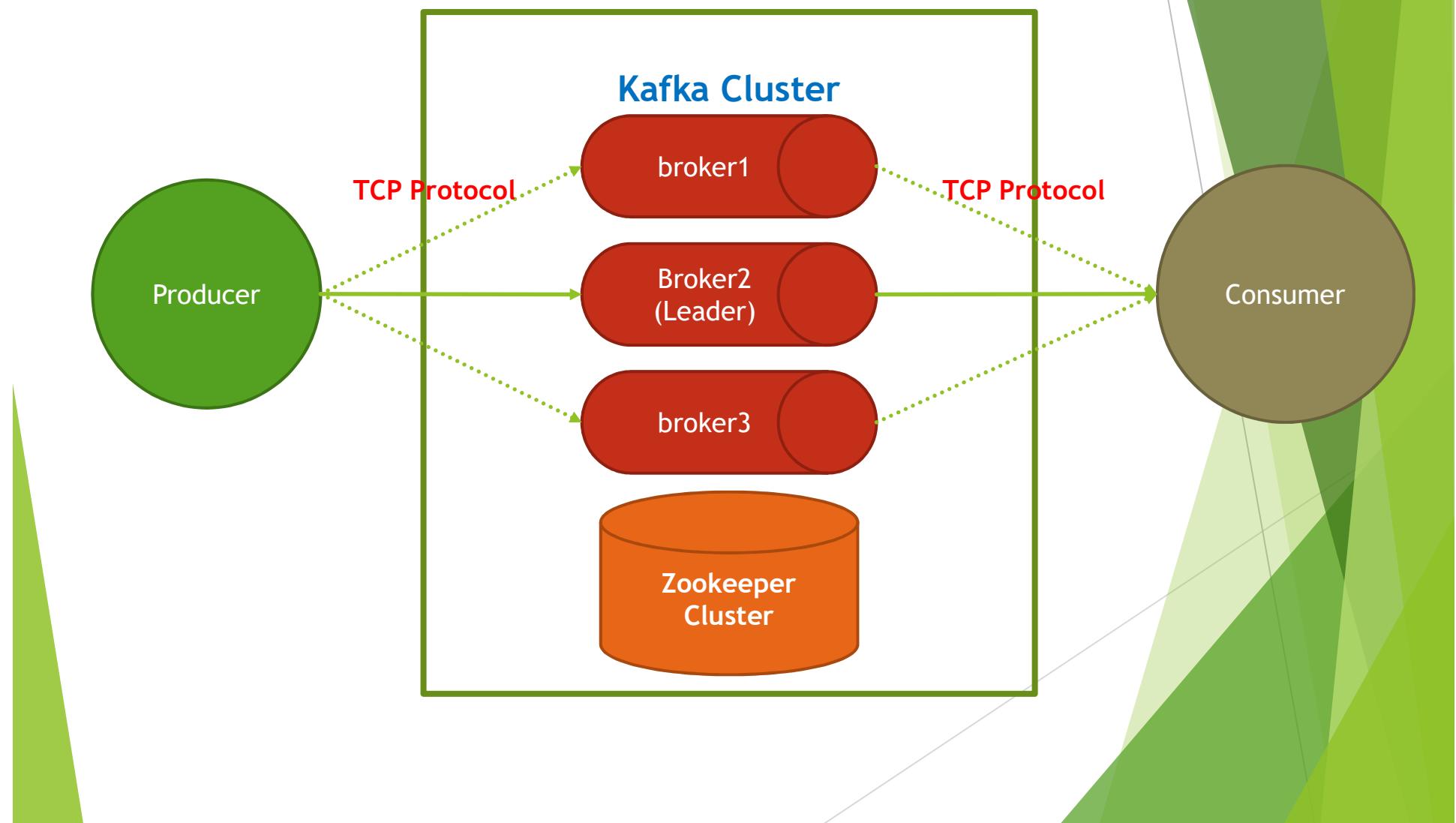
- Sink

- Removing an event from the channel.
- Putting it into an external repository or another source.

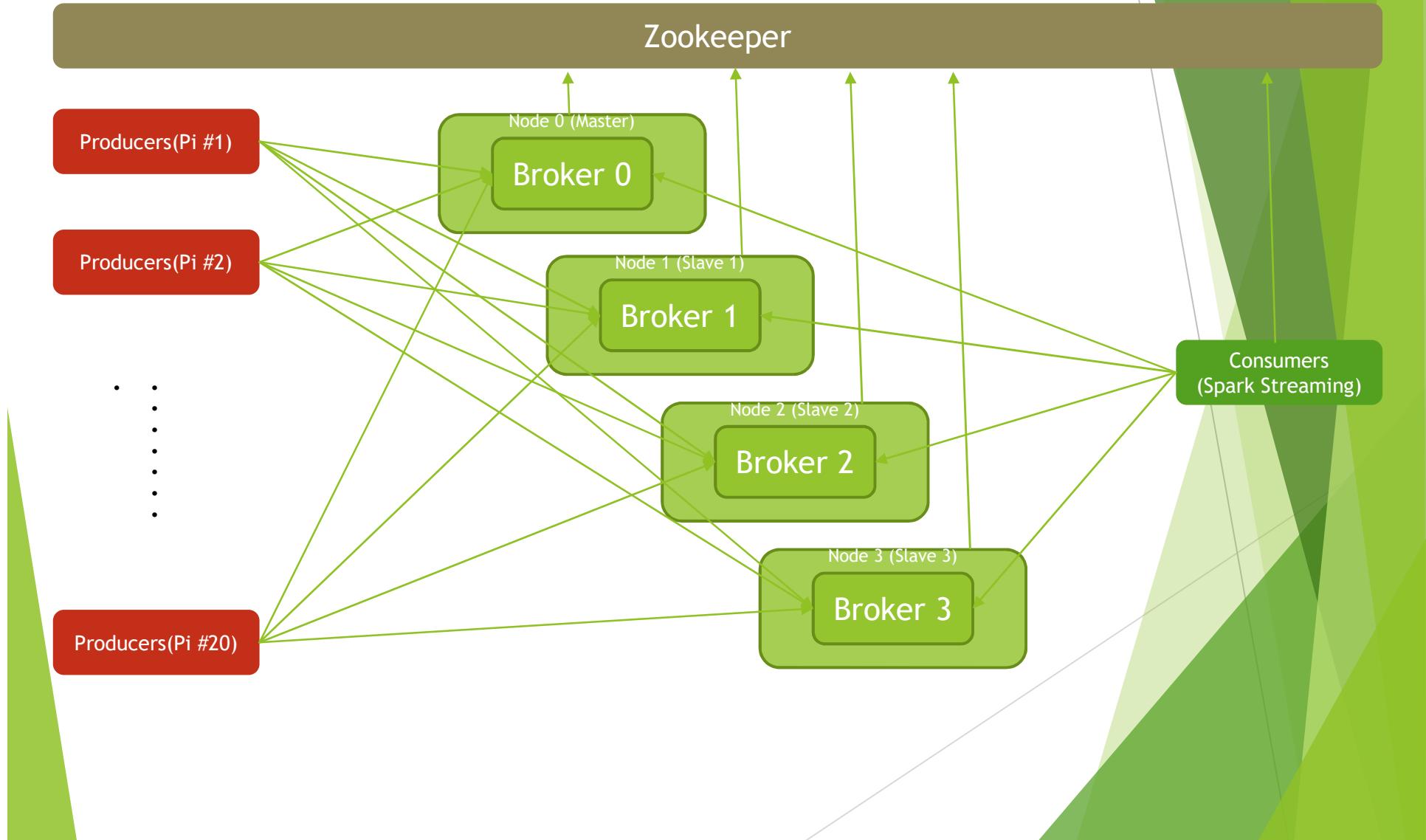


- **Kafka**
  - is a **distributed, partitioned, replicated** commit log service.
  - It provides **the functionality of a messaging system**, but with a unique design
- **Basic messaging terminology**
  - **Topics:**
    - maintains feeds of messages in categories
  - **Producers:**
    - processes that publish messages to a Kafka topic
  - **Consumers:**
    - processes that subscribe to topics and process **the feed of published messages**
  - **Broker:**
    - run as a cluster comprised of one or more servers

# Kafka: Architecture



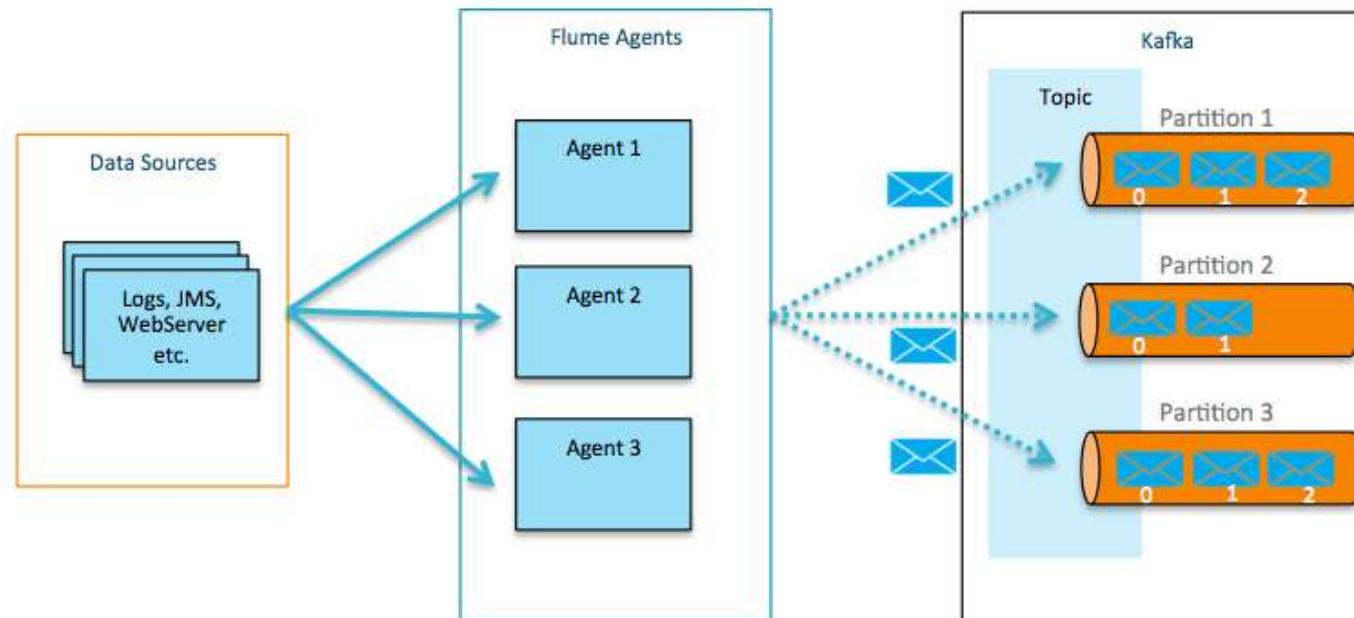
## Cont'd



# Kafka: with Flume

## ► Flafka

: Apache Flume Meets Apache Kafka for Event Processing

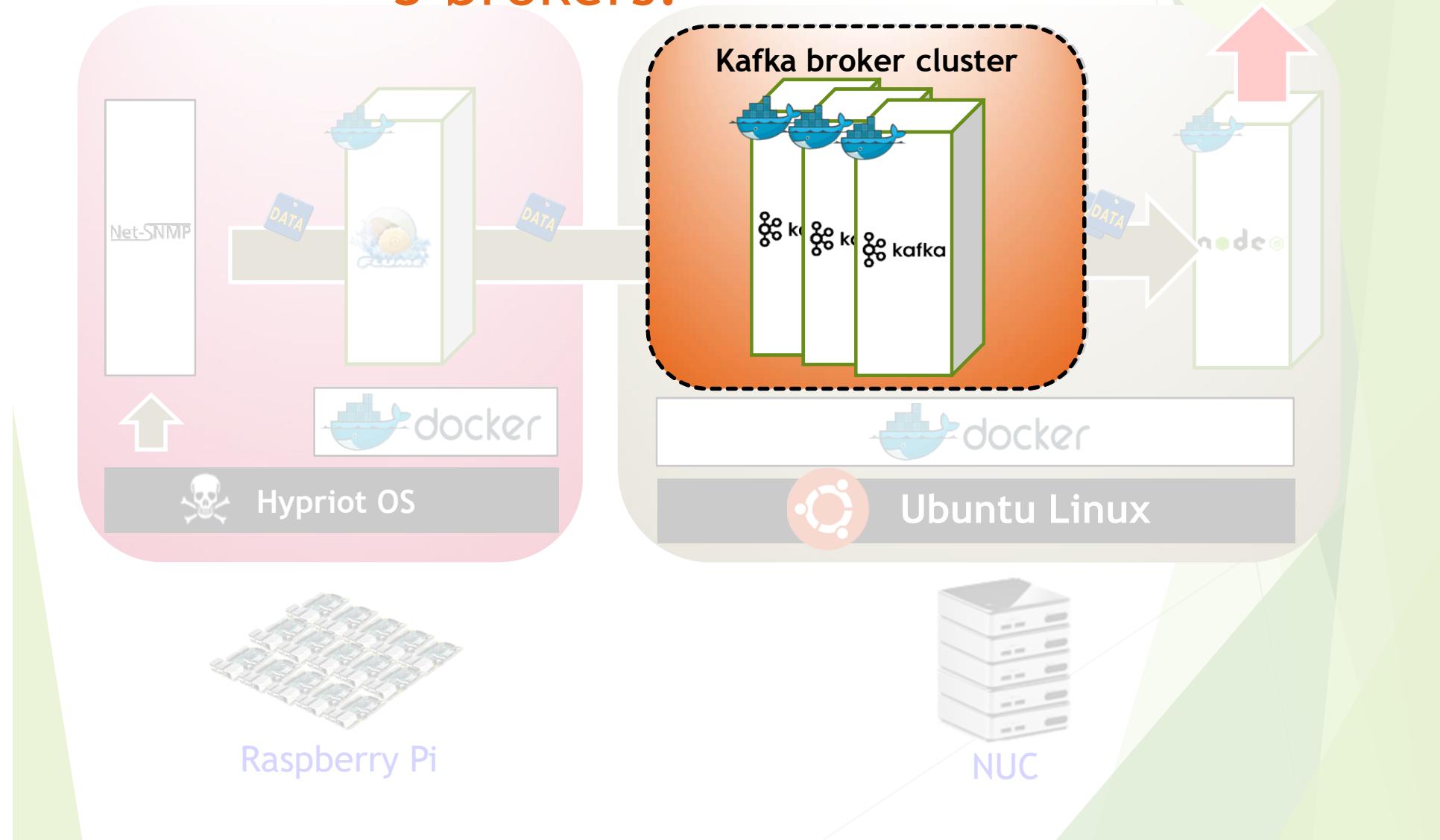


# Connecting Configuration on NUC



# In this section

3 brokers!



# 1. Download Source from Github

- ▶ Download all files from Github  
([http://github.com/SmartXBox/SmartX-mini\\_](http://github.com/SmartXBox/SmartX-mini_)
  - `$ git clone https://github.com/SmartXBox/SmartX-mini.git`
- ▶ Folder List



We'll use it



## 2. Define a address table

1. We'll use 1 **zookeeper** and 3 **broker containers** which have own public IP address
2. Let's define your own address table
3. We'll type these on each container

(For Example)

Container Name	IP address	Broker id	Listening port
zookeeper1	210.125.88.10	-	2181
broker0	210.125.88.20	0	9092
broker1	210.125.88.21	1	9092
broker2	210.125.88.22	2	9092

### 3. Build Docker image

1. `$cd ~/SmartX-mini/ubuntu-kafka`
  2. Build Dockerfile   ※ It takes long time.  
`$ docker build --tag ubuntu-kafka .`
  3. Run Docker Container (recommend making new terminal window)  
`$ docker run -it --net=none -h [host name] --name [container name]  
ubuntu-kafka`
- ▶ If you want to look for more details about Docker command, see  
<https://docs.docker.com/reference/commandline/>

# 4. Allocate IP address on Container



1. Let's check the present status

\$ifconfig

```
root@zookeeper:/kafka# ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
root@zookeeper:/kafka#
```

2. Go out from container

(Ctrl+P)(Ctrl+Q)

3. Adding bridge port for container's interface

\$ sudo ovs-docker add-port br0 eth0 [container name] --ipaddress=[container ip address]/24 --gateway=[gateway address]

## 5. Edit /etc/hosts

- ▶ Every machine which Kafka runs on must know all of their host name with IP address.

1. Go into container

```
$ docker attach [container name]
```

2. Edit /etc/hosts

```
$ sudo vi /etc/hosts
```

(For Example)

```
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

210.125.88.10 zookeeper1
210.125.88.20 broker0
210.125.88.21 broker1
210.125.88.22 broker2
```

# 6-1. Configure Zookeeper properties



- ▶ Actually we use default configurations

1. Open zookeeper properties file

`$vi config/zookeeper.properties`

2. Check the client port

```
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# the directory where the snapshot is stored.
dataDir=/tmp/zookeeper
# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of connections since this is a non-production config
maxClientCnxns=0
```

## 6-2. Configure Kafka properties



1. Open server properties file

\$vi config/server.properties

2. Editing proper broker id (it must be unique) and zookeeper address
3. Check the port number

```
##### Server Basics #####
# The id of the broker. This must be set to a unique integer
# across all brokers in the cluster.
broker.id=0
# broker id

##### Socket Server Settings #####
# The port the socket server listens on
port=9092

##### Zookeeper #####
# Zookeeper connection string (see zookeeper.properties)
# This is a comma separated host:port pairs
# server. e.g. "127.0.0.1:3000,127.0.0.1:3001"
# You can also append an optional chroot string
# root directory for all kafka znodes.
zookeeper.connect=localhost:2181
# zookeeper address
```

- ▶ Follow the same procedures (3~6)  
for every container (zookeeper 1, broker 3)

# 7. Launching Zookeeper node



- ✓ zookeeper must launch first

`$bin/zookeeper-server-start.sh config/zookeeper.properties`

```
[2015-11-20 04:13:18,607] INFO Server environment:java.library.path=/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib (o
[2015-11-20 04:13:18,607] INFO Server environment:java.io.tmpdir=/tmp (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,607] INFO Server environment:java.compiler=<NA> (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,607] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,607] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,607] INFO Server environment:os.version=3.19.0-25-generic (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,607] INFO Server environment:user.name=root (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,607] INFO Server environment:user.home=/root (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,608] INFO Server environment:user.dir=/kafka (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,614] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,614] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,614] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-11-20 04:13:18,625] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-11-20 04:13:19,034] INFO Accepted socket connection from Zookeeper address :48648 (org.apache.zookeeper.server.NIOServerCnxnFacto
[2015-11-20 04:13:19,135] INFO Client attempting to renew session 0x15122d708dd000c at Zookeeper address :48648 (org.apache.zookeeper.s
[2015-11-20 04:13:19,142] INFO Established session 0x15122d708dd000c with negotiated timeout 6000 for client Zookeeper address :48648 (o
[2015-11-20 04:13:19,632] INFO Accepted socket connection from Zookeeper address :48649 (org.apache.zookeeper.server.NIOServerCnxnFacto
[2015-11-20 04:13:19,632] INFO Client attempting to renew session 0x15122d708dd000b at Zookeeper address :48649 (org.apache.zookeeper.s
[2015-11-20 04:13:19,633] INFO Established session 0x15122d708dd000b with negotiated timeout 30000 for client Zookeeper address :48649
```

# 8. Launching Kafka brokers

- ✓ Attach into each kafka broker container and run scripts to launch  
`$bin/kafka-server-start.sh config/server.properties`

```
INFO Logs loading complete. (kafka.log.LogManager)
INFO Starting log cleanup with a period of 300000 ms. (kafka.log.LogManager)
INFO Starting log flusher with a default period of 9223372036854775807 ms. (kafka.log.LogManager)
INFO Awaiting socket connections on 0.0.0.0:9092. (kafka.network.Acceptor)
INFO [Socket Server on Broker 0], Started (kafka.network.SocketServer)
INFO Will not load MX4J, mx4j-tools.jar is not in the classpath (kafka.utils.Mx4jLoader$)
INFO 0 successfully elected as leader (kafka.server.ZookeeperLeaderElector)
INFO New leader is 0 (kafka.server.ZookeeperLeaderElector$LeaderChangeListener)
INFO Registered broker 0 at path /brokers/ids/0 with address broker1:9092. (kafka.utils.ZkUtils$)
INFO [Kafka Server 0], started (kafka.server.KafkaServer)
```

# 9. Making Consumer container



1. Making new kafka container for consumer

```
$docker run -it --net=host --name [container name] ubuntu-kafka
```

2. Set /etc/hosts as other kafka brokers'

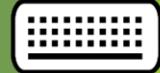
Container Name	IP address	Broker id	Listening port
zookeeper1	210.125.88.10	-	2181
broker0	210.125.88.20	0	9092
broker1	210.125.88.21	1	9092
broker2	210.125.88.22	2	9092

# 10. Making topic on Consumer



- ▶ Create topic
  - ▶ `$ bin/kafka-topics.sh --create --zookeeper [zookeeper host name]:2181 --replication-factor 1 --partitions 3 --topic <topic_name>`
- ▶ We can check topics'.
  - topic List
    - ▶ `$ bin/kafka-topics.sh --list --zookeeper <zookeeper host name>:2181`
  - topic specification
    - ▶ `$ bin/kafka-topics.sh --describe --zookeeper <zookeeper host name>:2181 --topic <topic_name>`

# 11. Consuming message from brokers



## 1. Launch consumer script

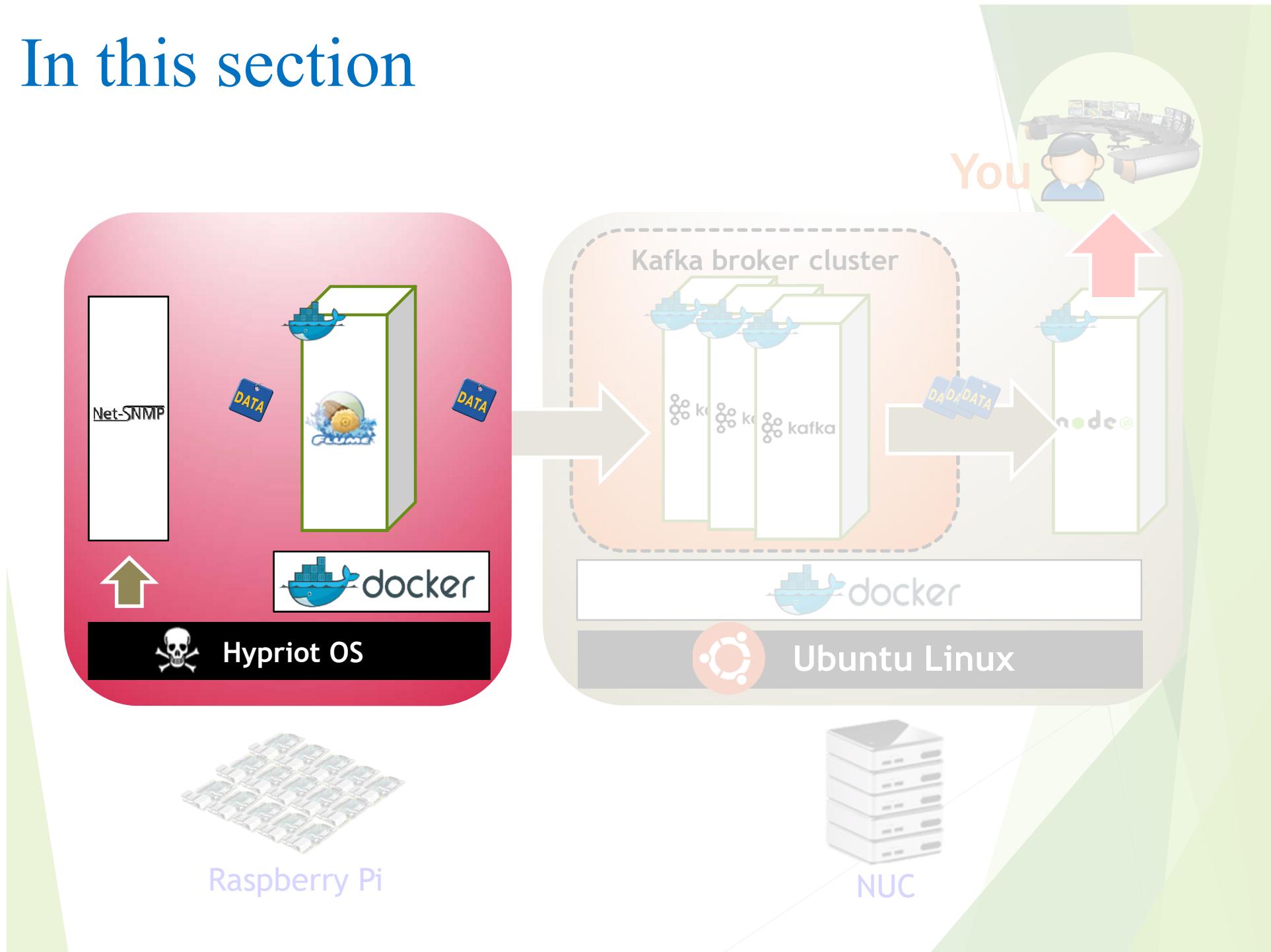
▶ `$bin/kafka-console-consumer.sh --zookeeper [zookeeper host name]:2181 --topic [topic name] --from-beginning`

```
! 1 zookeeper x ! 2 broker1 x ● 3 broker2 x ● 4 broker3 x ● 5 consumer x ! 6 pi01 x
1447989025957,172.17.42.1,0,0.06,12.00,82120,0,163164,506512,98,76170,0,0,673468,8
1447989026969,172.17.42.1,0,0.06,12.00,82120,0,163164,506512,98,76170,0,0,673468,8
1447989027986,172.17.42.1,0,0.06,12.00,82152,0,163168,506516,98,76170,0,0,673468,8
1447989029004,172.17.42.1,0,0.06,12.00,82152,0,163168,506516,98,76170,0,0,673468,8
1447989030019,172.17.42.1,0,0.06,12.00,82152,0,163168,506516,98,76170,0,0,673468,8
1447989031031,172.17.42.1,0,0.06,12.00,82152,0,163168,506516,98,76170,0,0,673468,8
1447989032042,172.17.42.1,0,0.06,12.00,82152,0,163168,506516,98,76170,0,0,673468,8
1447989033054,172.17.42.1,0,0.06,12.00,82152,0,163172,506516,98,76170,0,0,673468,8
1447989034067,172.17.42.1,0,0.06,12.00,82152,0,163172,506516,98,76170,0,0,673468,8
1447989035081,172.17.42.1,0,0.06,12.00,82152,0,163172,506516,98,76170,0,0,673468,8
1447989036094,172.17.42.1,0,0.06,12.00,82152,0,163172,506516,98,76170,0,0,673468,8
1447989037106,172.17.42.1,0,0.06,12.00,82152,0,163172,506516,98,76170,0,0,673468,8
1447989038119,172.17.42.1,0,0.06,12.00,82120,0,163180,506520,98,76171,0,0,673468,8
1447989039131,172.17.42.1,0,0.06,12.00,82120,0,163180,506520,98,76171,0,0,673468,8
1447989040142,172.17.42.1,0,0.06,12.00,82120,0,163180,506520,98,76171,0,0,673468,8
1447989041156,172.17.42.1,0,0.06,12.00,82120,0,163180,506520,98,76171,0,0,673468,8
1447989042160,172.17.42.1,0,0.06,12.00,82120,0,163180,506520,98,76171,0,0,673468,8
```

# Connecting Configuration on Raspberry Pi



# In this section



# 1. Install Net-SNMP

- \$ sudo apt-get update
- ▶ Download Net-SNMP
  - \$ apt-get install -y snmp snmpd
- ▶ Download and apply mibs
  - \$ apt-get install -y snmp-mibs-downloader
  - \$ download-mibs
- ▶ Modify configuration file
  - \$ vi /etc/snmp/snmpd.conf
    - #rocommunity public localhost -> Delete #
  - \$ /etc/init.d/snmpd restart



## 2. Download Source from Github

- ▶ Git package is already installed in Hypriot OS
- ▶ Download all files from Github  
([http://github.com/SmartXBox/SmartX-mini\\_](http://github.com/SmartXBox/SmartX-mini_)
  - `$ git clone https://github.com/SmartXBox/SmartX-mini.git`
- ▶ Folder List



We'll use it

### 3. Edit /etc/hosts

- ▶ Adding raspberry pi and nuc's host name and ip address
- ▶ Also add zookeeper, broker container name and ip address
- ▶ **\$ sudo vi /etc/hosts**

```
127.0.0.1      localhost
::1            localhost ip6-localhost ip6-loopback
fe00::0        ip6-localnet
ff00::0        ip6-mcastprefix
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters
```

```
127.0.1.1      black-pearl
192.168.10.106 rpi06
192.168.10.10  master1
```

```
210.125.88.10  zookeeper1
210.125.88.20  broker0
210.125.88.21  broker1
210.125.88.22  broker2
```

# 4. Install Flume on RPi



## 1) Build Dockerfile

※ It takes long time

```
$ cd SmartX-mini/raspbian-flume  
$ docker build --tag raspbian-flume .  
$ docker run -it --net=host raspbian-flume
```

## 2) Modify configuration file

```
$ vi conf/flume-conf.properties
```

We modify this.

```
agent.sinks.sink1.topic=[topic_name]  
agent.sinks.sink1.brokerList=[broker_ipaddress:port]
```

# Cont'd

```
# The sink
agent.sinks.sink1.type = org.apache.flume.sink.kafka.KafkaSink
agent.sinks.sink1.topic = test
agent.sinks.sink1.brokerList = master1:9092
agent.sinks.sink1.requiredAcks = 1
agent.sinks.sink1.batchSize = 1
```



```
# The sink
agent.sinks.sink1.type = org.apache.flume.sink.kafka.KafkaSink
agent.sinks.sink1.topic = topic1
agent.sinks.sink1.brokerList = broker1:9092,broker2:9092,broker3:9092
agent.sinks.sink1.requiredAcks = 1
agent.sinks.sink1.batchSize = 1
```

# 4. Run Flume Agent

- Run Flume on RPi

```
$ bin/flume-ng agent --conf conf --conf-file conf/flume-conf.properties --name agent -Dflume.root.logger=INFO,console
```

```
root@black-pearl:/flume# bin/flume-ng agent --conf conf --conf-file conf/flume-conf.properties --name agent -Dflume.root.logger=INFO,console
```

# Default Message Format in SmartX-mini

- ▶ Kafka message value format

Timestam p	IP address	Monitoring item number	Monitoring item value
------------	------------	------------------------	-----------------------

ex) 1428151677440,192.168.56.1,4,0.00

- ▶ Monitoring Item numbers

- ▶ CPU

- : user(0), nice(1), system(2), iowait(3), steal(4), idle(5)

- ▶ Memory

- : total(6), free(7), buffer(8), cached(9)

- ▶ Storage

- : tps(10), kbReads(11), kbWrtns(12), kbRead(13), kbWrtn(14)

- ▶ Network

- : ttl(15), latencyTime(16)

Thank You for  
Your Attention  
Any Questions?

