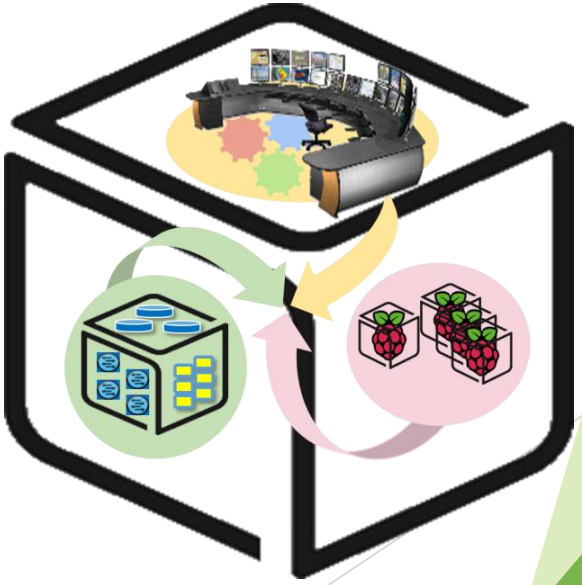
SmartX Labs
for Computer Systems

InterConnect Lab v0.6

(2016, Spring)

NetCS Lab



History and Contributor of InterConnect Lab (2016. 07. 03)

Version	Updated Date	Updated Contents	Contributor
-	2016/04	(구) InterConnect Lab 최종본 작성	김 승 룡
v0.1	2016/04	단어 교정 및 그림 추가 및 수정, RPi(SNMP, SmartX-mini) 설치 부분 추가	김 철 원
V0.2	2016/05	History 작성	김 철 원
V0.3	2016/05	슬라이드 제목 및 순서 수정, 이미지 수정 및 추가	김 철 원
V0.4	2016/05	오타 및 문구 수정	김 철 원
V0.5	2016/05	강의 도중 발생한 문제 피드백, 제목 수정, 슬라이드 순서 변경	김 철 원
V0.6	2016/07	Raspberry Pi 및 Hypriot에 대한 간단한 설명 추가, 편의를 위한 내용 추가, 목차 추가	김 철 원

InterConnect LAB
-Table of contents-

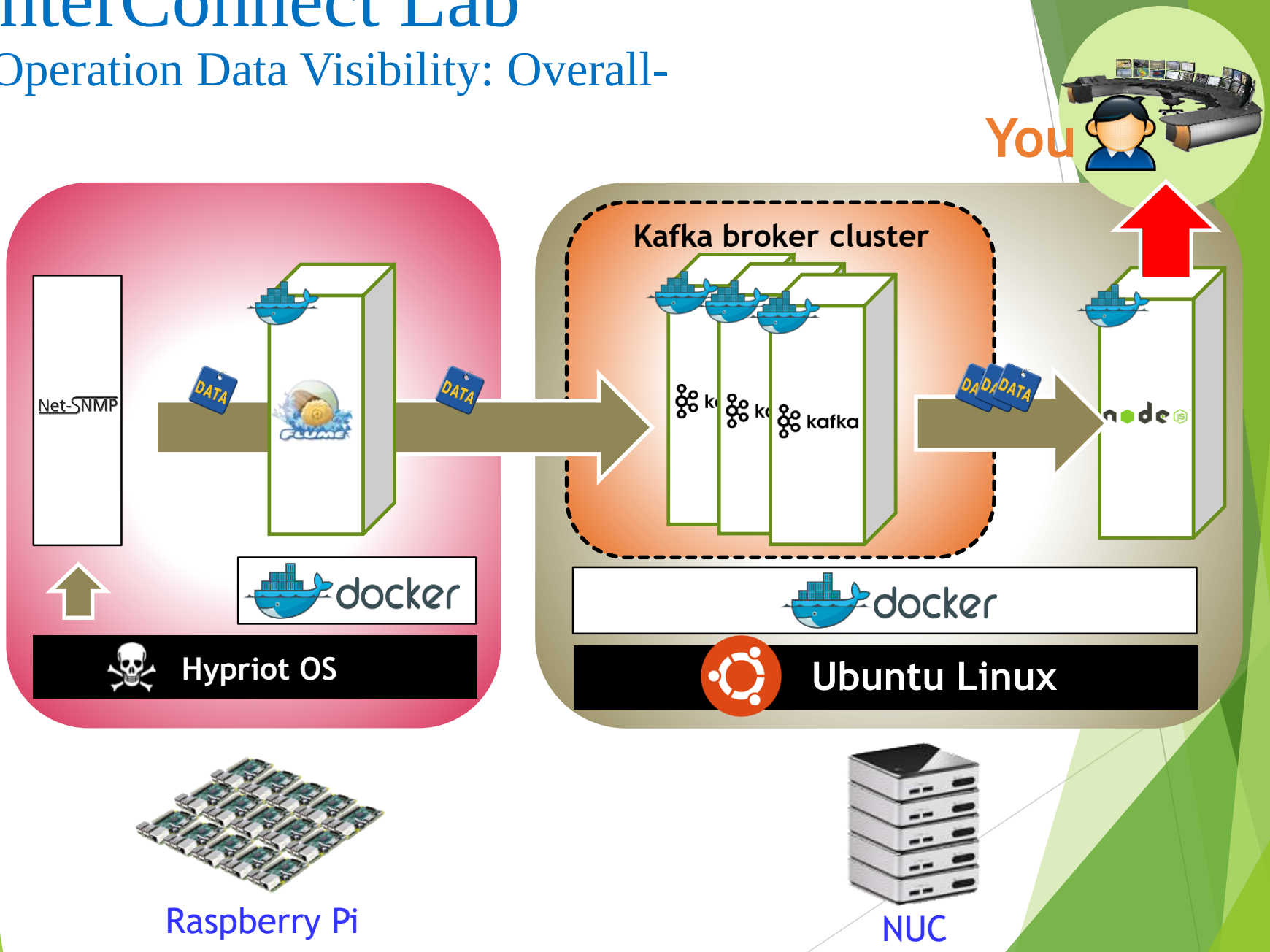
- Concepts
  - Net-SNMP, Flume, Kafka, Hypriot OS
- Configuration
  - Raspberry Pi and NUC
- Result
- Appendix
  - Default message format
  - Raspberry Pi2 OS Setting
  - NUC & Pi2 IP address Setting

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

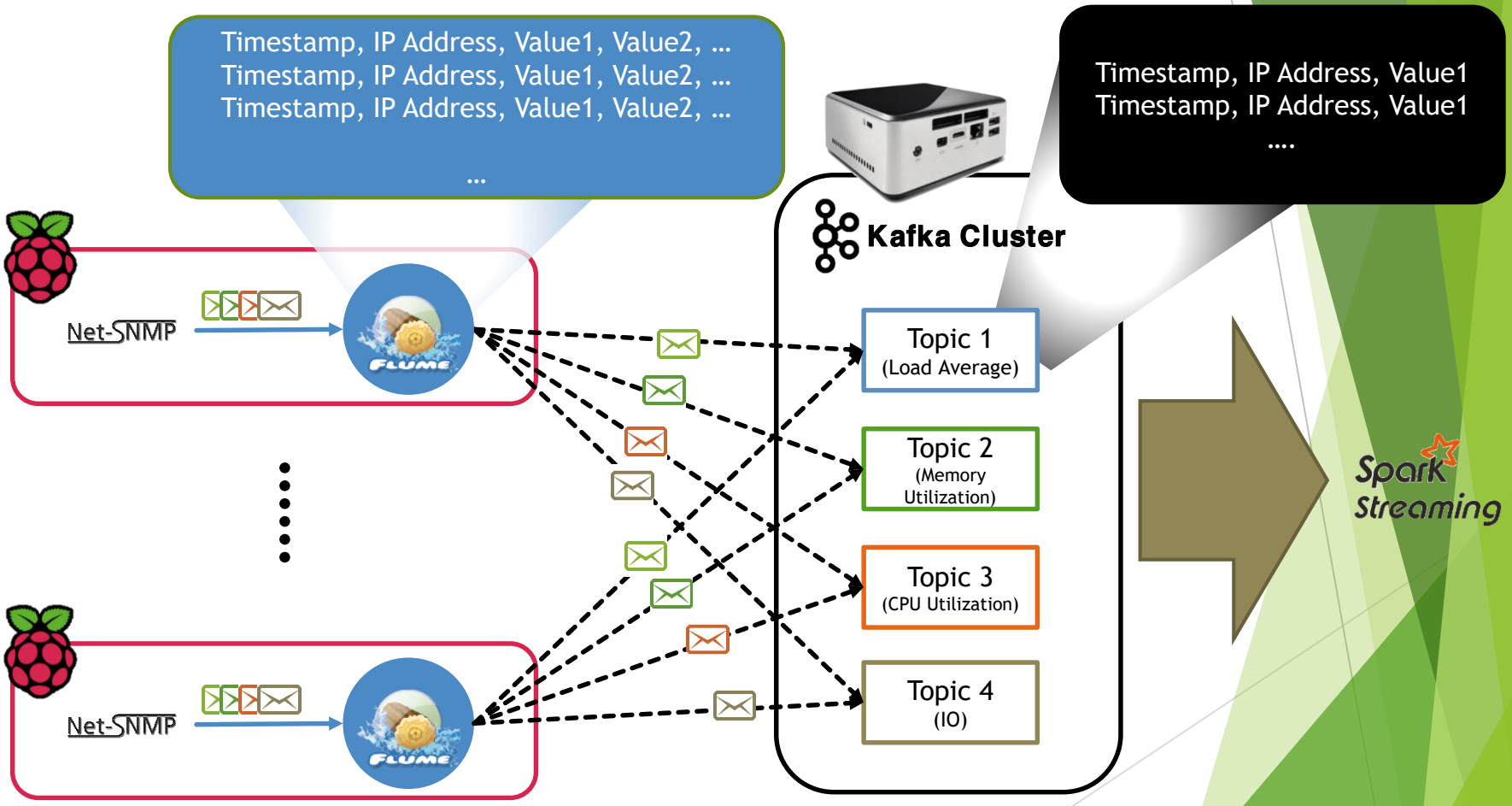
InterConnect Lab

-Operation Data Visibility: Overall-



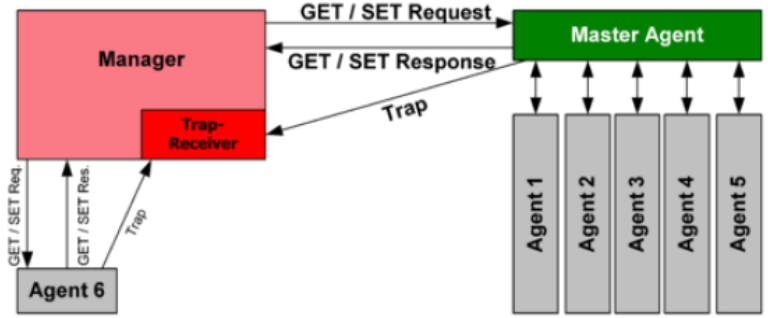
InterConnect Lab

-Operation Data Visibility: Data flow-



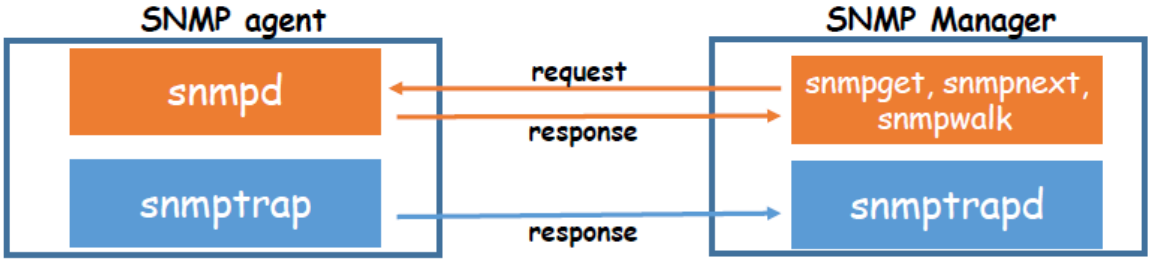
SNMP

- 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.



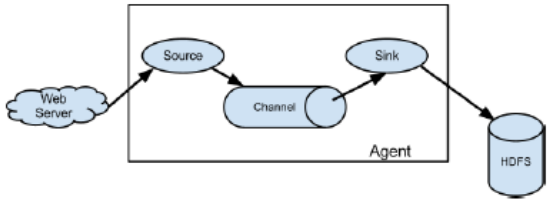
Net-SNMP Net-SNMP

- 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

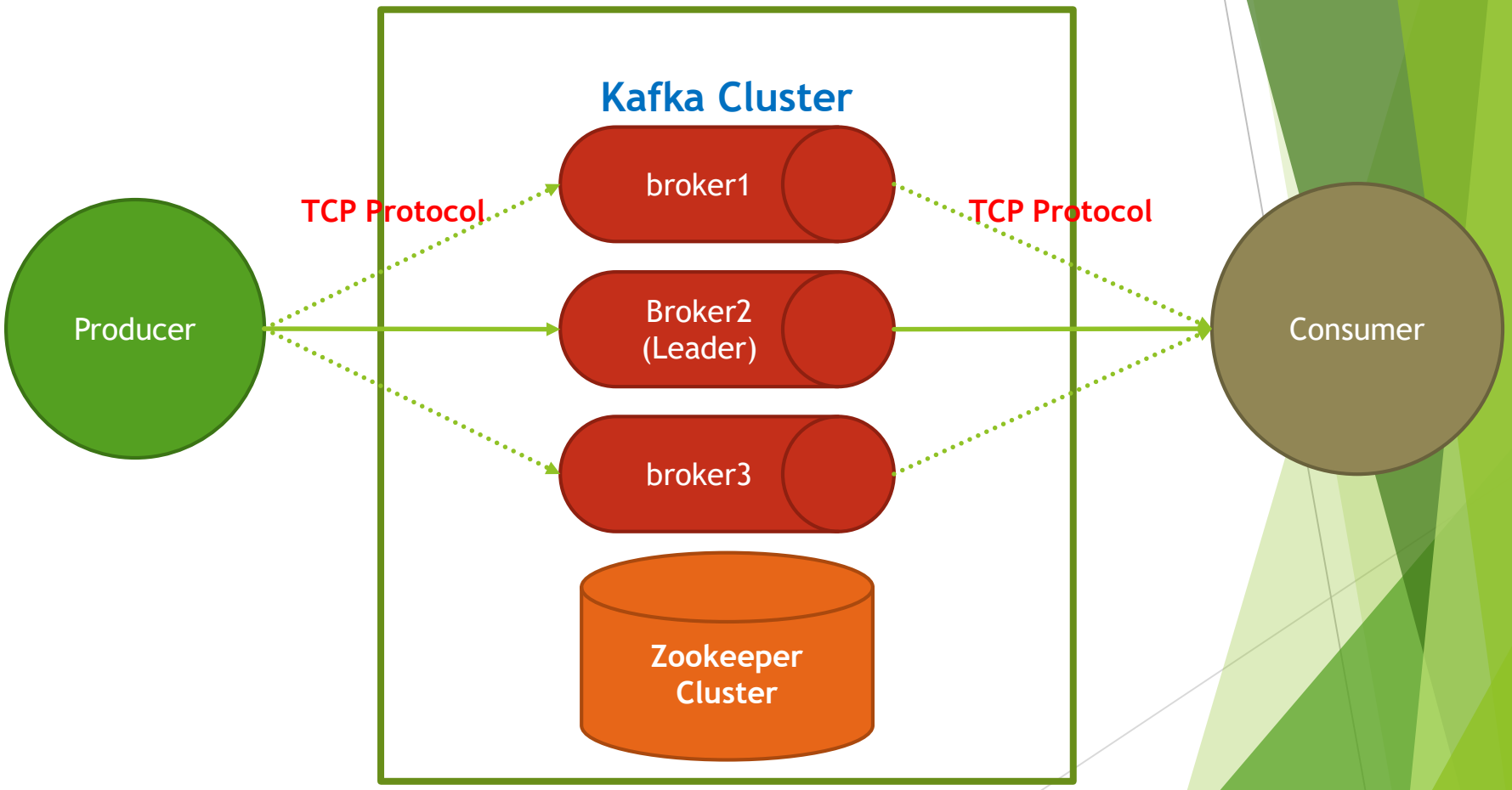


- Flume Agent
  - **Source**
    - Consuming events having a specific format.
    - Delivering it to the channel
  - **Channel**
    - Holding the event until that consumed
  - **Sink**
    - Removing an event from the channel.
    - Putting it into an external repository or another source.

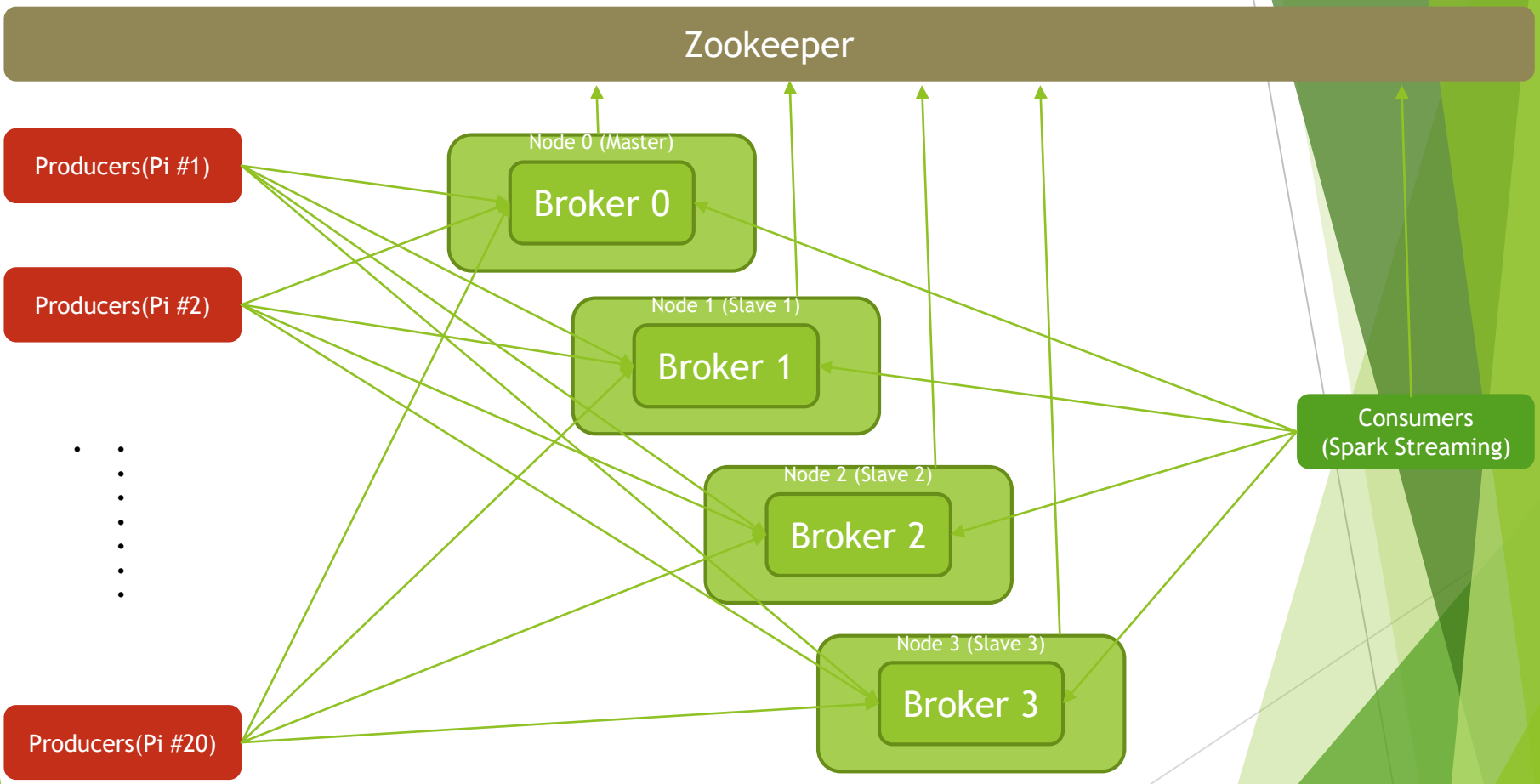
Kafka

- 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



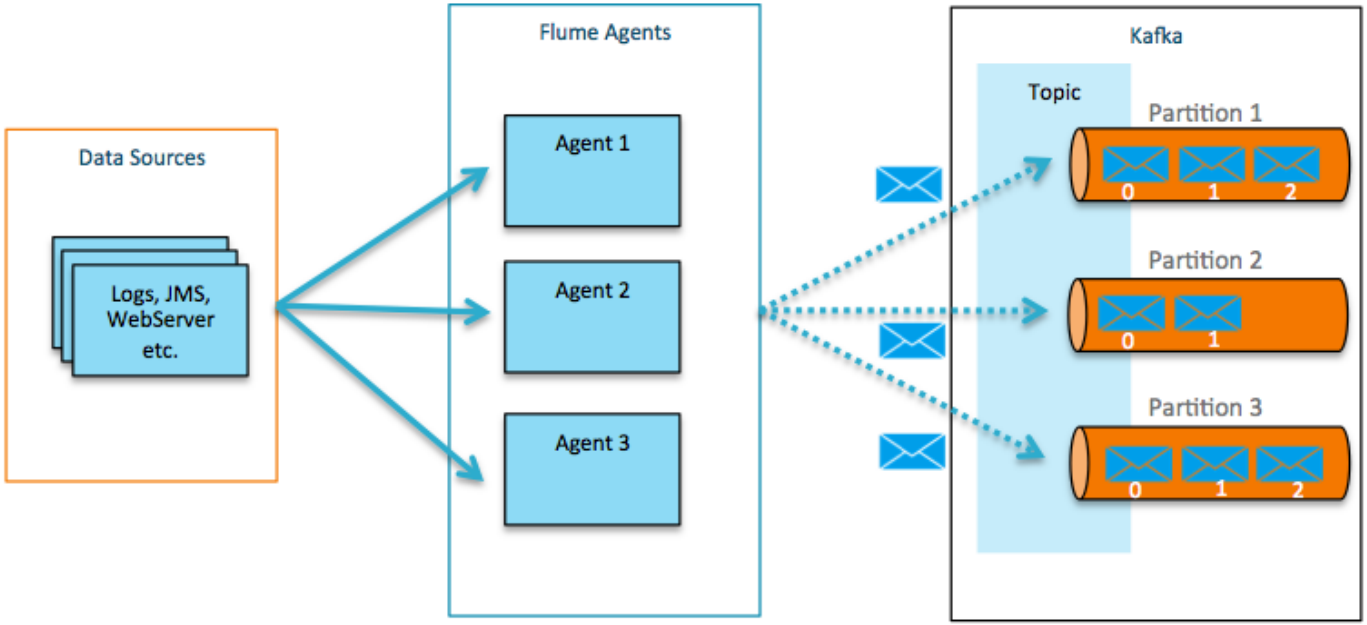
Cont'd


Zookeeper
Producers(Pi #1)
Producers(Pi #2)
Producers(Pi #20)
Node 0 (Master)
Broker 0
Node 1 (Slave 1)
Broker 1
Node 2 (Slave 2)
Broker 2
Node 3 (Slave 3)
Broker 3
Consumers
(Spark Streaming)

Kafka: with Flume

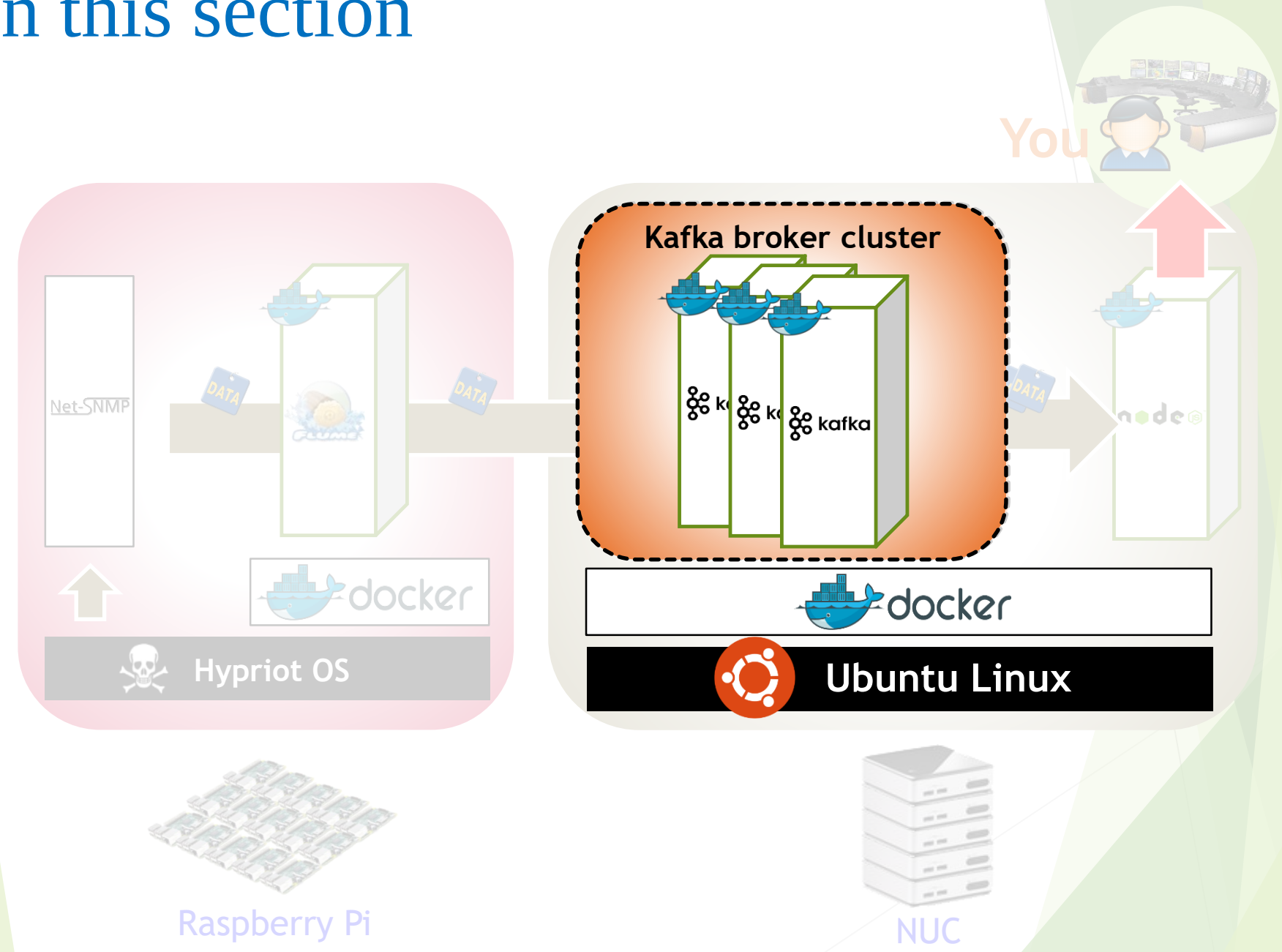
- **Flafka**

  : Apache Flume Meets Apache Kafka for Event Processing



Connecting Configuration on NUC

In this section



1. Download Source from Github

- Download all files from Github
  (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)

Host Name	IP address	Broker id	Listening port
zookeeper	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

3-1. Build Docker

- Build Docker Part

1. $cd ~/SmartX-mini/ubuntu-kafka
2. Build Dockerfile ※ It takes long time.

   $ docker build --tag ubuntu-kafka .

- $ If you want to check docker instruction word

  $ docker --help

  ex) docker ps : List containers

  docker start : Start one or more stopped containers

  docker rm : Remove one or more containers

3-2. Run Docker
(recommend making new terminal window)

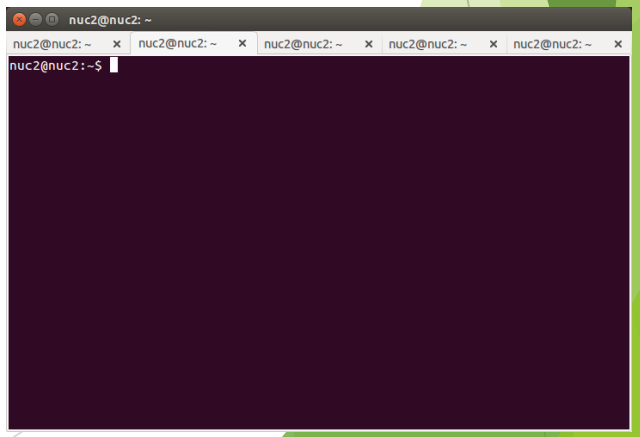
- Run Docker Part

  Run Docker Container

  $ 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/

  - When you make new terminal, you just press 'ctrl + shift + t', then you can make multiple terminals like taps.
    Also, you want to change tap name,
    press right mouse button and select set titles.



4-1. 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]

   // if you have a problem about setting ipaddress, use this command.

   $ sudo ovs-docker del-ports br0 [container name]

4-2. 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
```

[host name] (O)
[container name] (X)

5-1. 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
```

5-2. 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
broker.id=0
############################# Socket Server S

# The port the socket server listens on
port=9092
```

broker id

```
############################# Zookeeper ###

# Zookeeper connection string (see zookeepe
# This is a comma separated host:port pairs
# server. e.g. "127.0.0.1:3000,127.0.0.1:30
# You can also append an optional chroot st
# root directory for all kafka znodes.
zookeeper.connect=localhost:2181
```

zookeeper address

- Follow the same procedures (3-2 ~ 5) for every container (zookeeper 1, broker 3)

6. 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 (
[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
```

7. 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)
```

8. 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'

Host 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

9. Making topic on Consumer

- Create topic
  - $ bin/kafka-topics.sh --create --zookeeper [zookeeper host name]:2181 --replication-factor 1 --partitions 3 --topic <topic_name>

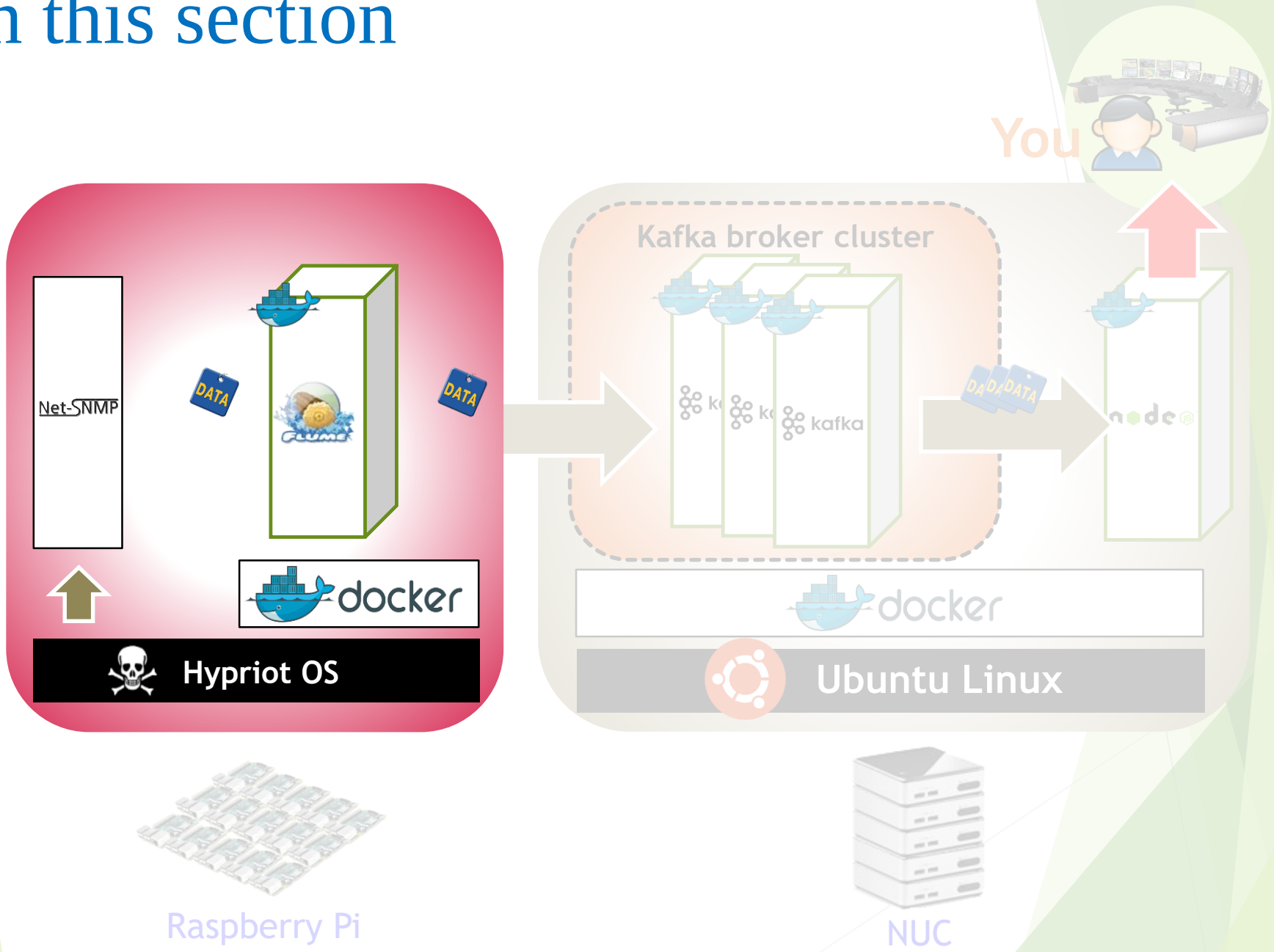
- We can check topics'.

  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>

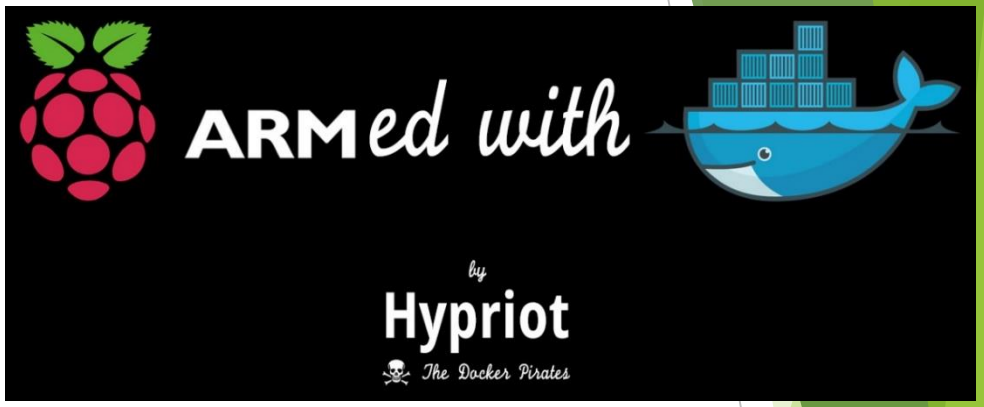
Connecting Configuration on Raspberry Pi

In this section



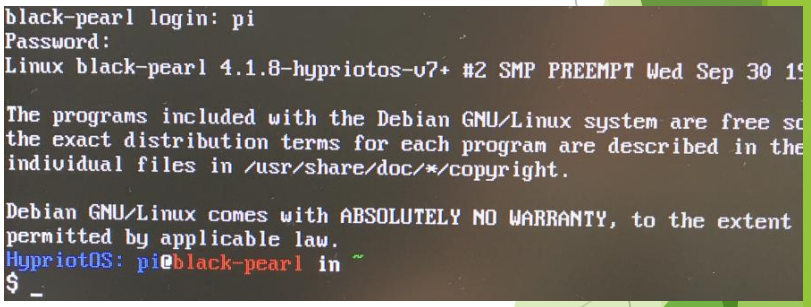
0. Before Setting

- Hypriot
  - Raspberry Pi OS
  - To run docker



- Cf) Raspbian Jessie

  32bit OS -> It can't run docker



- Therefore we must use Hypriot OS.

- Detailed Information about Raspberry Pi & hypriot, you can check Appendix.

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

```
#  createUser authOnlyUser  MD5 "remember to change this password"
#  createUser authPrivUser  SHA "remember to change this one too"  DES
#  createUser internalUser  MD5 "this is only ever used internally, but still change the password"

#  If you also change the usernames (which might be sensible),
#  then remember to update the other occurances in this example config file to match.



###############################################################################
#
#  ACCESS CONTROL
#

                                                #  system + hrSystem groups only
view   systemonly  included   .1.3.6.1.2.1.1
view   systemonly  included   .1.3.6.1.2.1.25.1

                                                #  Full access from the local host
 rocommunity public  localhost
                                                #  Default access to basic system info
 rocommunity public  default    -V systemonly
                                                #  rocommunity6 is for IPv6
 rocommunity6 public  default   -V systemonly


                                                #  Full access from an example network
                                                #     Adjust this network address to match your local
                                                #     settings, change the community string,
                                                #     and check the 'agentAddress' setting above
#rocommunity secret  10.0.0.0/16
```

2. Download Source from Github

- Git package is already installed in Hypriot OS
- Download all files from Github

  (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

  -> we can see host name $ hostname

- Also add zookeeper, broker host 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.propert
ies --name agent -Dflume.root.logger=INFO,console
```

Result

Consuming message from brokers

1. Launch consumer script and Result
   - $bin/kafka-console-consumer.sh --zookeeper [zookeeper host name]:2181 --topic [topic name] --from-beginning

```
root@netcs10: /kafka
7574,3385904
[2016-06-28]18:40:31,203.237.53.127,0.13,0.10,0.11,47744,0,119340,592488,98,23657,0,0,41
7574,3385904
[2016-06-28]18:40:32,203.237.53.127,0.12,0.10,0.11,47776,0,119348,592492,98,23657,0,0,41
7574,3385976
[2016-06-28]18:40:33,203.237.53.127,0.12,0.10,0.11,47776,0,119348,592492,98,23657,0,0,41
7574,3385976
[2016-06-28]18:40:34,203.237.53.127,0.12,0.10,0.11,47776,0,119348,592492,98,23657,0,0,41
7574,3385976
[2016-06-28]18:40:35,203.237.53.127,0.12,0.10,0.11,47776,0,119348,592492,98,23657,0,0,41
7574,3385976
[2016-06-28]18:40:36,203.237.53.127,0.12,0.10,0.11,47776,0,119348,592492,98,23657,0,0,41
7574,3385976
[2016-06-28]18:40:37,203.237.53.127,0.11,0.10,0.11,47776,0,119352,592496,98,23657,0,0,41
7574,3386008
[2016-06-28]18:40:38,203.237.53.127,0.11,0.10,0.11,47776,0,119352,592496,98,23657,0,0,41
7574,3386008
[2016-06-28]18:40:39,203.237.53.127,0.11,0.10,0.11,47776,0,119352,592496,98,23657,0,0,41
7574,3386008
[2016-06-28]18:40:40,203.237.53.127,0.11,0.10,0.11,47776,0,119352,592496,98,23657,0,0,41
7574,3386008
[2016-06-28]18:40:41,203.237.53.127,0.11,0.10,0.11,47776,0,119352,592496,98,23657,0,0,41
7574,3386008
[2016-06-28]18:40:42,203.237.53.127,0.10,0.10,0.11,47744,0,119356,592500,98,23657,0,0,41
7574,3386040
```

Default Message Format in SmartX-mini

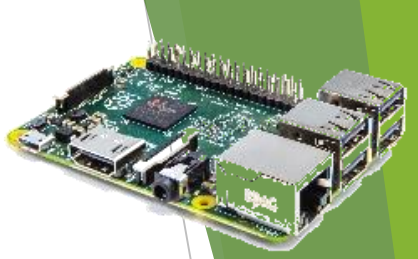
- Kafka message value format

Timestamp	IP address	Monitoring item number	Monitoring item value

ex) [2016-06-28]18:40:33,203.237.x.x,0.12,0.10,0.11,47776,0, ...

- 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)

Raspberry Pi2 OS Setting



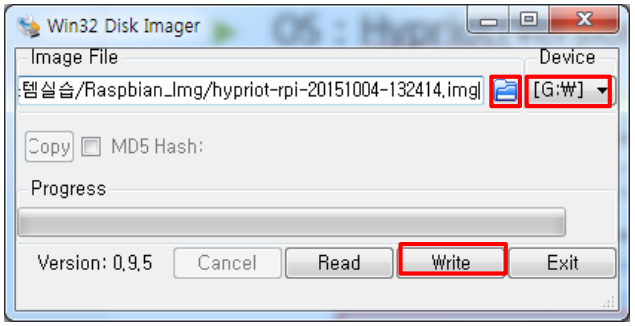
- OS : Hypriot(Version : 0.5 Will, 07.10.2015 published)
  - Download Site : http://blog.hypriot.com/downloads/

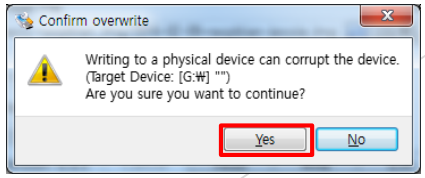
Hypriot Docker Image for Raspberry Pi

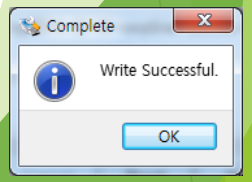
Download and flash this image to your SD card.
Start your Pi with the flashed SD card and enjoy instant Docker awesomeness.

Description	Download Link	SHA256 Checksum	Published
Version 0.6.1 Hector	hypriot-rpi-20151115-132854.img.zip	Checksum	15.11.2015
Version 0.6 Hector	hypriot-rpi-20151103-224349.img.zip	Checksum	03.11.2015
Version 0.5 Will	hypriot-rpi-20151004-132414.img.zip	Checksum	07.10.2015
Version 0.5 Will (beta)	hypriot-rpi-20150727-151455.img.zip	Checksum	27.07.2015
Version 0.4 Elizabeth	hypriot-rpi-20150416-201537.img.zip	Checksum	16.04.2015

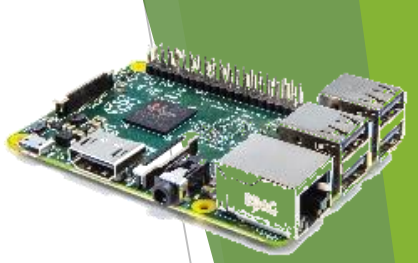
  - 압축을 푼 후 파일(hypriot-rpi-20151004-132414.img, 1.39Gb)은 SD Writer 를 이용하여 Write.
  - SD Writer Download : https://sourceforge.net/projects/win32diskimager/files/latest/download?source=navbar



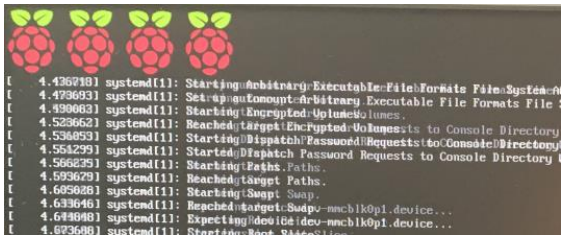




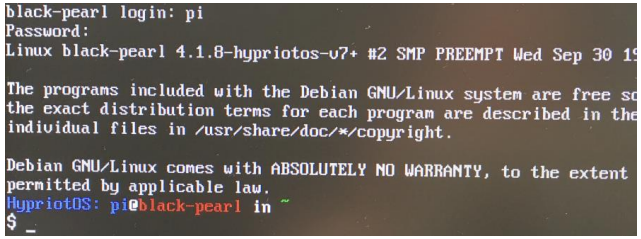
Raspberry Pi2 Environment Setting



- Micro SD 카드에 Hypriot 설치가 완료되면 Pi2에 삽입 후 부팅



loading 화면



CUI 환경으로 부팅되면 성공
Login ID : root
Password : hypriot

- Root password 변경
  - package 설치, RPM upgrade, 시스템 관리를 위해 필요
    $sudo passwd root

```
HypriotOS: root@black-pearl in ~
$ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
HypriotOS: root@black-pearl in ~
$
```

  - password 입력시 * 표시가 나오지 않지만 입력되고 있으니 걱정하지 말 것
  - **root password는 꼭 기억할 것!!!**
  - login ID인 pi 계정의 password 도 변경하도록 하자. **$passwd**

NUC & Pi2 IP address Setting

- Pi2 IP address 설정을 위해 필요한 파일(편집은 root 만 가능)
  - /etc/network/interfaces

$cd /etc/network
$sudo vi interfaces

```
HypriotOS: pi@black-pearl in ~
$ cd /etc/network/
HypriotOS: pi@black-pearl in /etc/network
$ sudo vi interfaces
```

#iface eth0 inet dhcp ← # 은 주석

auto eth0

iface eth0 inet static

address 172.29.0.X ← ip address

netmask 255.255.255.0 ← subnet mask

gateway 172.29.0.254 ← Gateway

dns-nameservers 203.237.32.100 203.237.32.101

입력 예

```
auto lo
iface lo inet loopback

#allow-hotplug eth0
#iface eth0 inet dhcp

auto eth0
iface eth0 inet static
        address 172.29.1.9
        netmask 255.255.255.0
        gateway 172.29.1.254
        dns-nameservers 203.237.32.100

iface eth0 inet6 auto

allow-hotplug wlan0
iface wlan0 inet dhcp
pre-up /usr/bin/occi
wpa-conf /etc/wpa_supplicant/wpa_suppl
iface default inet dhcp
```

- 일반적으로 dns-nameservers 를 입력하면 45 page 는 필요 없으나, Hypriot OS 는 삽입되지 않으므로 resolv.conf 파일에 직접 nameserver를 입력해야 함!

NUC & Pi2 IP address Setting

- Pi2 IP address 설정을 위해 필요한 파일
  - /etc/resolv.conf

$cd /etc/
$sudo vi resolv.conf

```
# nameserver config
nameserver 203.237.32.100
nameserver 203.237.32.101
```

기존의 nameserver 는 #을 추가하여 주석처리

```
# nameserver config
#nameserver 213.133.98.98
#nameserver 213.133.99.99
#nameserver 213.133.100.100

nameserver 203.237.32.100
nameserver 203.237.32.101
```

$sudo /etc/init.d/networking restart 입력 또는 rebooting 후 network 확인
$sudo reboot (rebooting command)

```
$ sudo /etc/init.d/networking restart
[....] Restarting networking (via systemctl): networking.serviceWarni
ce changed on disk, 'systemctl daemon-reload' recommended.
. ok
HypriotOS: pi@black-pearl in /etc
```

Thank You for Your Attention Any Questions?