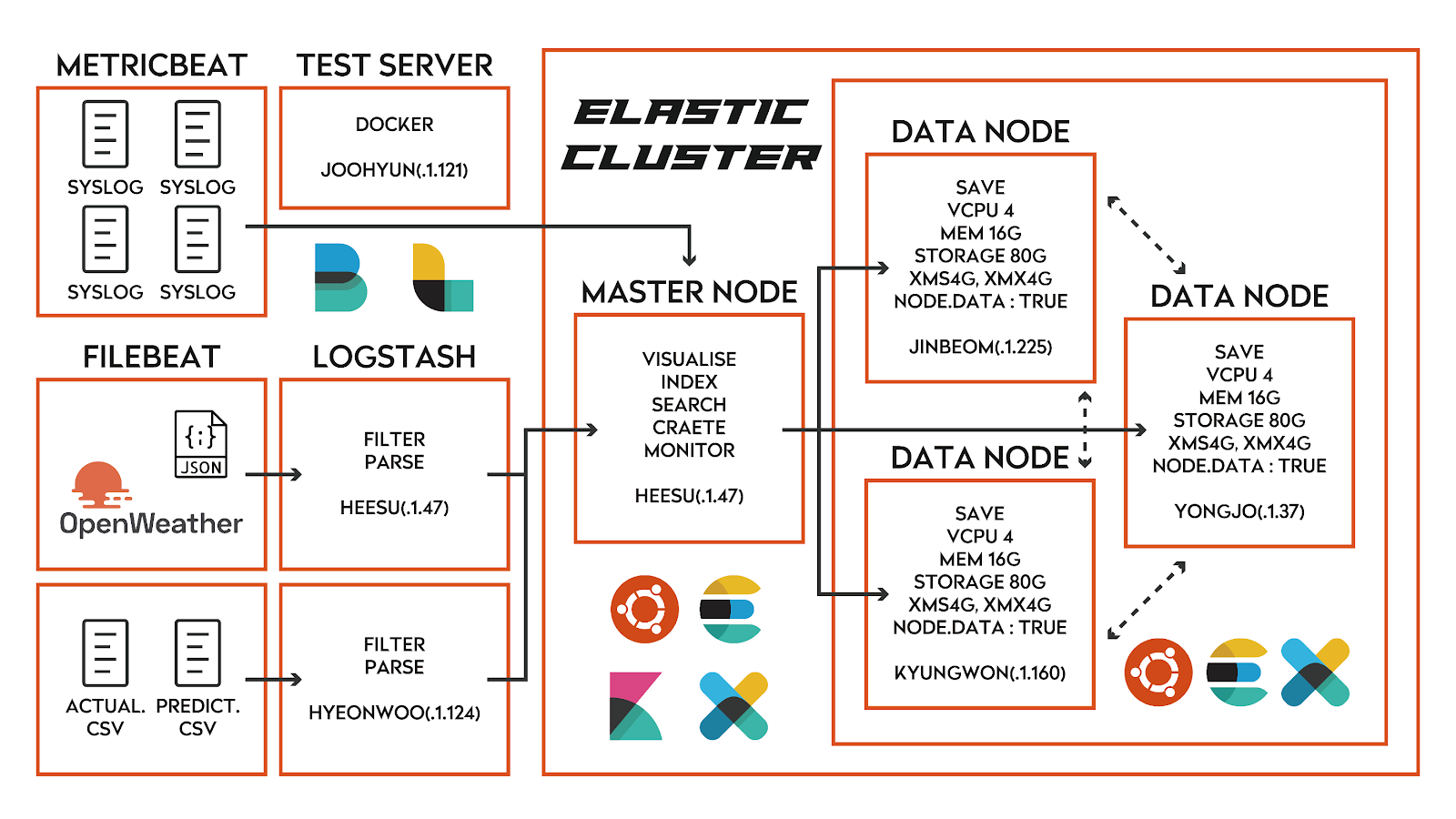
**구축 목적**

* 실시간으로 들어오는 로그데이터에 대해서 분석하여 결과를 보여주는 환경 구축

**참고 사항**

* 실시간으로 끌어와서 분석할 데이터의 부족으로 저장되어 있는 데이터를 4초 단위로 보내서 스트리밍 데이터와 유사한 환경 구축

**전체 구조**



**Elasticsearch**

* 관련 설정파일은 /etc/elasticsearch/elasticsearch.yml
* 로그경로 /var/log/elasticsearch/[cluster.name].log
* 데이터경로 /var/lib/elasticsearch
* 자바 힙메모리 설정 /etc/elasticsearch/jvmoption.yml
* 기본적으로 cluster.name이 같을 경우 자동으로 같은 클러스터로 UUID 할당
* xpack으로 모든 노드를 모니터링 가능

**Elasticsearch – Master Node**

* 이하 내용은 /etc/elasticsearch/elasticsearch.yml
* Cluster.name 설정 (후에 설명할 Data Node도 같은 클러스터에 묶을 거라면 같은 이름으로 설정)
* Node.name 설정 (default => ${HOSTNAME})
* 기본적으로 규모가 작은 클러스터 라면 모든 노드를 싱글노드(마스터와 데이터노드 역할 모두 수행)로 설정하는 것이 좋다. 현재는 마스터와 데이터 노드를 구분하여 클러스터를 구축하는 것이 목적이므로 아래와 같이 진행한다.
* node.master: true // node.data: false
* network.host: \_site\_ (기본으로 localhost로 지정되어 있다. \_site\_로 변경할 경우 test 모드가 아닌 production 모드로 진행된다.)
* discovery.seed\_hosts: 클러스터에 속하는 모든 노드의 ip를 작성한다.
* Cluster.initial\_master\_nodes: 마스터 노드 후보들의 ip 를 작성한다.
  + 마스터 노드의 경우 하나만 설정하는 경우는 없다.
  + 기본적으로 (모든 노드 수 / 2) + 1로 계산하며 홀수로 설정한다.
  + 현재는 구축 환경의 제한으로 마스터 노드는 한대만 설정했다.
* xpack으로 모니터링 하기 위해서는 아래 두가지 설정을 작성해야 한다.
  + xpack.monitoring.enabled: true
  + xpack.monitoring.collection.enabled: true
* bootstrap.memory\_lock 설정을 통해서 elasticsearch의 메모리를 다른 프로세스가 점유하지 못하게 설정하는 것이 좋다. 현재 환경에서는 필요하지 않아 미설정

**Elasticsearch – Data Node**

* Master Node와 다른 설정에 대해서만 작성한다.
* node.master: false // node.data: true

**Logstash**

* 관련 설정파일 경로 : /etc/logstash/logstash.yml
* 실행하기 위해서는 /etc/logstash/conf.d/ 아래에 .conf파일이 있어야 한다.
* 여기서는 temp.conf 파일을 작성하였다.

input {

    beats {

        port => "5044"

}

}

filter {

    json {

        source => "message"

    }

    date {

        match => ["date", "yyyy M d H m s"]

        target => "dateStamp"

*#timezone => "UTC"*

*#locale => "ko"*

    }

    mutate {

        convert => {

            "temp" => "float"

        }

    }

}

output {

    elasticsearch {

        hosts => "192.168.1.47:9200"

        index => "temp-%{+YYYY.MM.dd}"

    }

*#stdout {  }*

}

* .conf파일은 크게 input / filter /output으로 나뉜다.
* Input의 경우 filebeat로 받아오기 때문에 beats 모듈을 사용하여 포트를 지정
* Filter의 경우 json모듈을 사용하여 message를 source로 설정한다.
  + date모듈은 전달받은 파일에서 date type을 지정해 주기 위해서 작성한다.
  + mutate모듈은 기본으로 string으로 지정된 숫자 컬럼을 float로 변환하였다.
* Output의 경우 elasticsearch로 전송한다.
  + 바로 ES로 보내기 전에 stdout으로 충분한 디버깅 후 전송하는 것이 좋다.
* 데이터가 csv로 들어올 경우 아래와 같이 설정해 줄 수 있다.

input {

file {

path => "/etc/logstash/conf.d/data/stock.csv"

start\_position => "beginning"

sincedb\_path => "NUL"

}

}

filter {

csv {

separator => ","

columns => ["Date", "Open", "High", "Low", "Close", "Volume", "Name"]

}

date {

match => ["Date", "yyyy-MM-dd"]

target => "DateStamp"

}

* 다양한 모듈을 지원하고 있으니 관련해서는 가이드 참고
  + <https://www.elastic.co/guide/en/logstash/7.6/index.html>

**Filebeat**

* 설정파일 경로 : /etc/filebeat/filebeat.yml
* Filebeat.inputs: 아래 내용을 수정해주면 된다.
  + enabled: true (default => false)
  + paths: filebeat가 주시하고 있을 파일 설정
    - ex) /var/log/\*.log
* output은 logstash, ES로 갈 수 있다. 지금은 Logstash로 전송하여 filter처리를 한다.
* output.logstash: hosts: [ipaddress:5044]