지능형 IoE 에지 컴퓨팅 기술
(EdgeX 기반)

2019. 04.05

유태완 (ETRI)
twyou@etri.re.kr

(참고) EdgeX에 관한 자료는 EdgeX Foundry (https://edgexfoundry.org) 자료를 인용 하였습니다.

목차

- Edge Platforms 소개
- EdgeX 실습
  - 사용자 – Work through
  - Hybrid – Device Service
  - Advanced 개발자
    - Tensorflow 연동

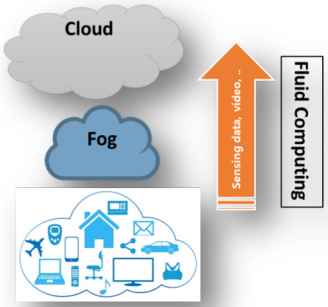
Edge computing

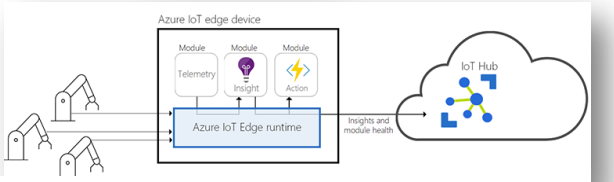
- Cloud Platform
  - *Interworking with central-Cloud*
  - *Distributed computing*

- IIoT Platform
  - *Convergence point – OT/CT/IT*
  - *On premise by factory, operator, etc.*

- Network Platform
  - *Distributed managements for 5G*

- AI Platform





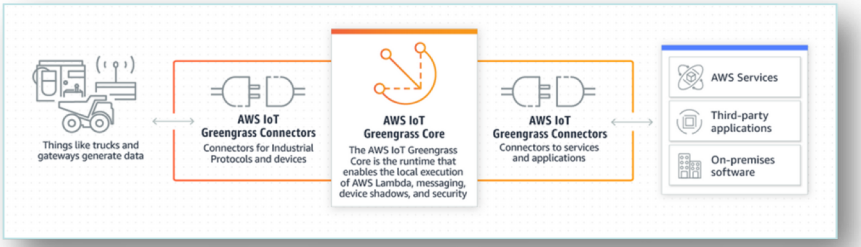
Running pre-trained models on mobile

Mobile Library	Platform	GPU	DNN Architecture Supported	Trained Models Supported
CoreML	iOS	Yes	CNN, RNN, SciKit	Keras, Tensorflow, MXNet
Tensorflow	iOS/Android	Yes	CNN,RNN,LSTM, etc	Tensorflow
Caffe2	iOS/Android	Yes	CNN	Caffe2, CNTK, PyTorch
Snapdragon NPE	Android	Yes	CNN, RNN, LSTM	Caffe, Caffe2, Tensorflow
CNNDroid	Android	Yes	CNN	Caffe, Torch, Theano
DeepLearningKit	iOS	Yes	CNN	Caffe
MXNet	iOS/Android	No	CNN,RNN,LSTM, etc	MXNet
Torch	iOS/Android	No	CNN,RNN,LSTM, etc	Torch

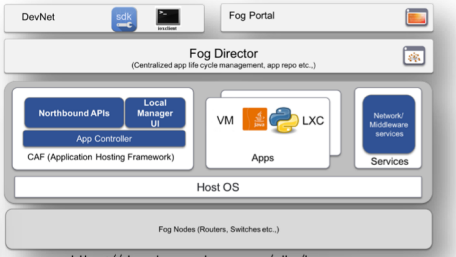
국외 기술 동향

- Cloud platform 기반 Edge computing 기술
  - *AWS Greengrass*
  - Microsoft Azure IoT Edge
  - Google Cloud Platform IoT Edge
- IIoT platform 을 위한 Edge computing 기술
  - GE Predix
  - *Cisco IOx*
- S/W platform 기술 (Open source)
  - *EdgeX Foundry*, etc.
- H/W platform
  - Nvidia
  - Intel
  - *Google Edge TPU*

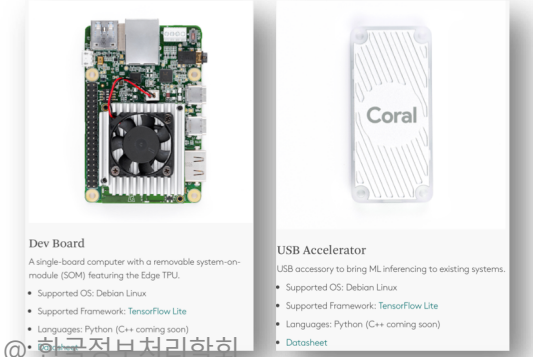
Azure aws Google Cloud Platform



https://docs.aws.amazon.com/greengrass/latest/developerguide/what-is-gg.html



https://developer.cisco.com/site/iox



https://cloud.google.com/iot-edge/

Edge computing + Intelligent (AI)

EdgeX 기반 개발

EdgeX 활용 방법

- Contributors (개발자)
  - Get the raw code, build it, and deploy the services to the target platform(s)
- Users (사용자)
  - Get EdgeX Docker container images and deploy/run to a platform where Docker is installed
- Hybrid
  - Get, build and deploy some of the services on your own
  - Get and use Docker container images for the other services

- Docker Compose is a tool to help get and run multiple containers
  - Docker Compose can be used with either the User or Hybrid approaches

[1] EdgeX Foundry Code, (https://github.com/edgexfoundry/)

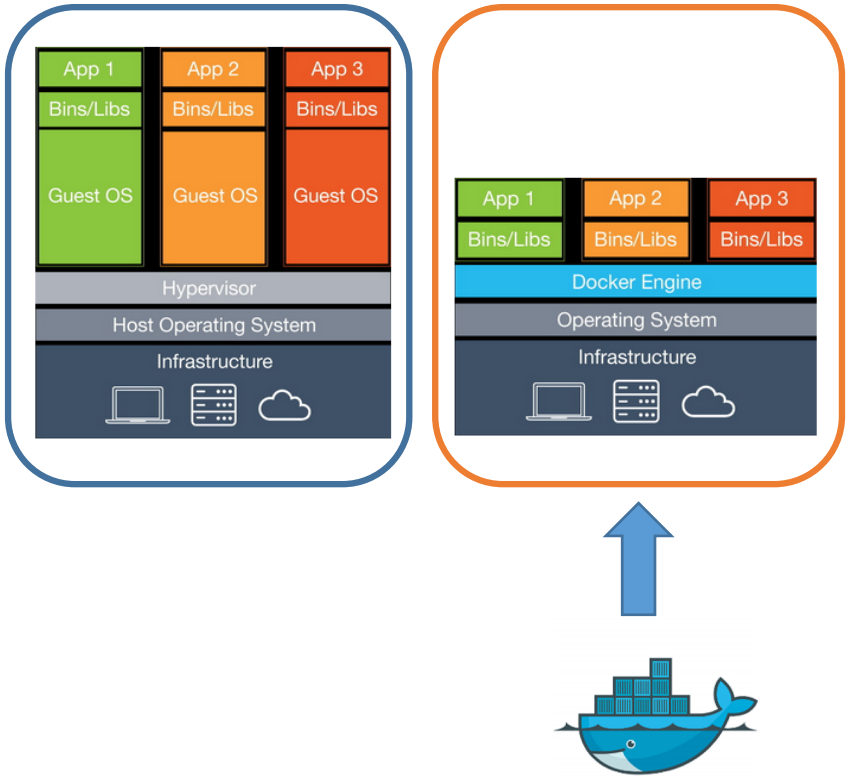
1. 사용자 접근 방법

- The EdgeX community provides *a Docker container image* as micro service
  - Get pre-built EdgeX micro services quickly
  - The container images can be run on any platform that runs Docker
  - There are different container images for hardware platforms (Intel or Arm)

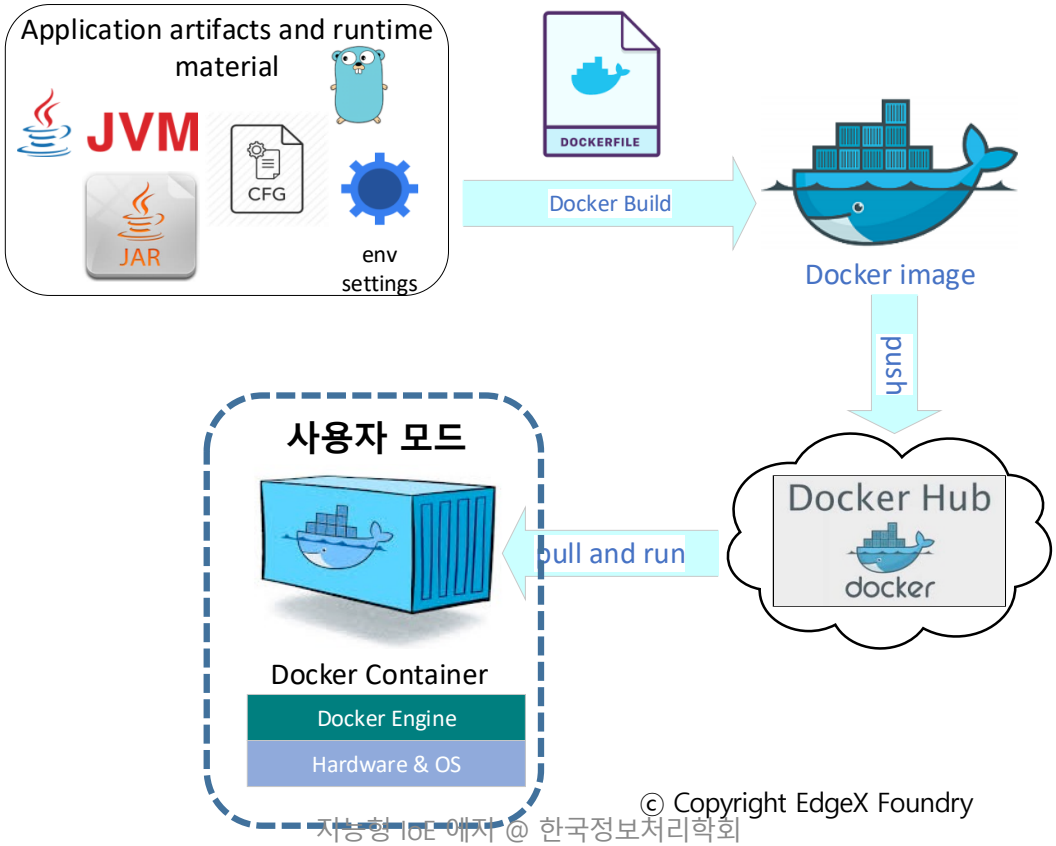
- The EdgeX Docker container images are available in Docker Hub (hub.docker.com)
  - The most recent code is always built to “developer” container images
  - These are made available from a Linux Foundation Nexus repository

추가자료 - Docker

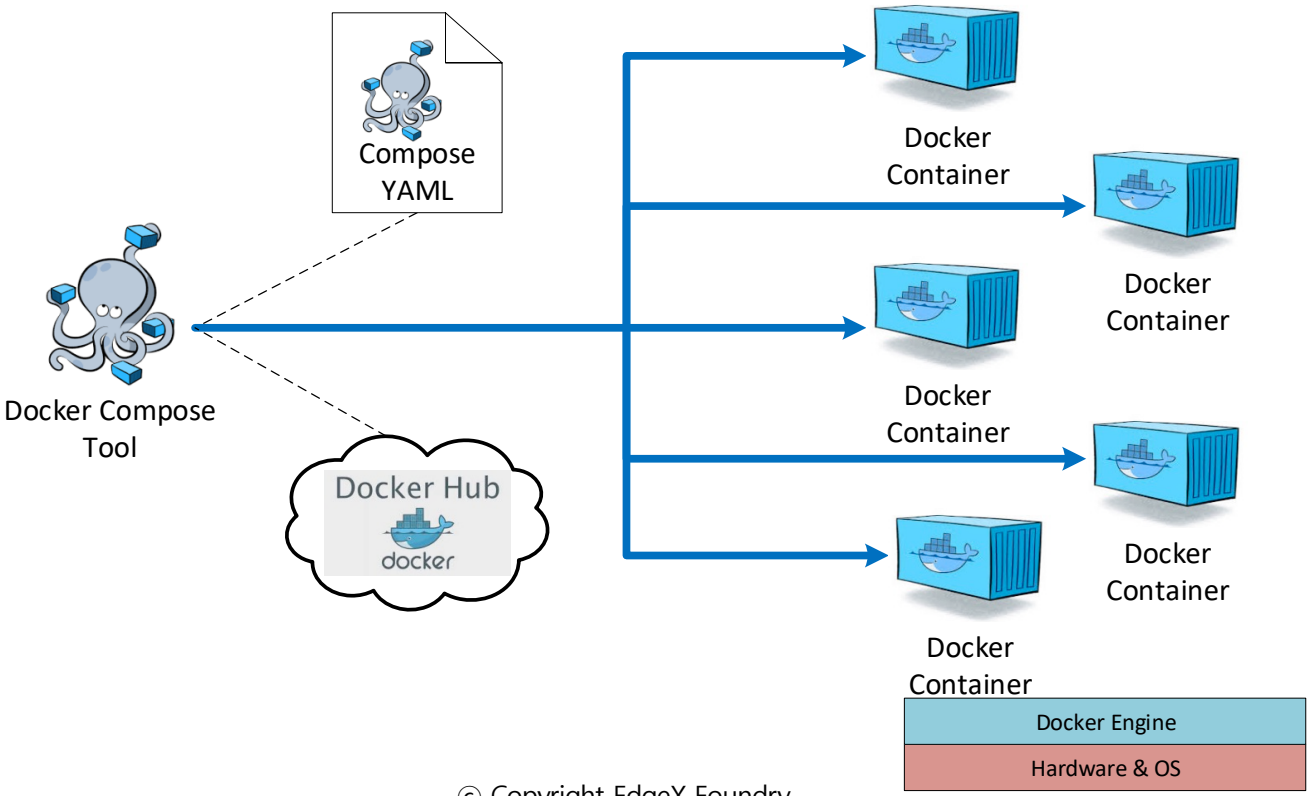
- Virtualization
  - Hypervisor vs. Container
- Container
  - operating-system-level virtualization by abstracting the “user space”
  - The one big difference between containers and VMs is that containers *share* the host system’s kernel with other containers.
    - Linux Container (LXC)
    - Docker



추가자료 - Docker



사용자 - Docker 활용



Docker-Compose 파일

- The Docker Compose YAML is a manifest file
  - It specifies to Docker …
  - What containers to pull down and start
  - What infrastructure (like a network) is needed for your containers
  - The order in which to start/stop containers

- Commands
  - *docker-compose pull -f <compose file name>*
    - pull the images but don't start them
  - *docker-compose -f <compose file name> up -d*
    - create and start all containers – the default compose file name is docker-compose.yml
  - *docker-compose -f <compose file name> stop <docker image>*
    - stop an existing container
  - *docker-compose -f <compose file name> logs -f --tail=100 <docker image>*
    - look at the last 100 lines of a micro services logs

EdgeX 의 ‘docker-compose.yaml’

- Github repository
  - https://github.com/edgexfoundry/developer-scripts/tree/master/compose-files

last updated: 12/11/2018

Docker Compose Files

This folder contains Docker Compose files for the following releases and usages:

- docker-compose-redis-delhi-0.7.1.yml: This file uses the EdgeX **Delhi release 0.7.1** container images. See Note 7.
- docker-compose-delhi-0.7.1.yml: This file uses the EdgeX **Delhi release 0.7.1** container images. See Note 6.
- docker-compose-delhi-0.7.0.yml: This file uses the EdgeX **Delhi release 0.7.0** container images.
- docker-compose-california-0.6.1.yml: This file uses the EdgeX **California release 0.6.1** container images. See Note 5.
- docker-compose-california-0.6.0.yml: This file uses the EdgeX **California release 0.6.0** container images.
- docker-compose-california-0.5.2.yml: This file includes mid-way versions of the Go core, support and export services with the latest Nexus container images. It is meant as a develoment/testing copy of EdgeX.
- docker-compose-barcelona-0.2.0.yml: This file uses the EdgeX **Barcelona release 0.2.0** container images. See Note 1
- docker-compose-barcelona-0.2.1.yml: This file uses the EdgeX **Barcelona release 0.2.1** container images. Release 0.2.1 was a bug fix release to version 0.2.0. See Note 2
- docker-compose.yml: This file uses the latest EdgeX container images from Docker Hub. This should be considered the EdgeX **developer's latest usable container images**.
- docker-compose-nexus.yml: This file uses the latest EdgeX container images from the EdgeX Nexus repository (**developer working images**) managed by the Linux Foundation. See Note 3

EdgeX 의 ‘docker-compose.yaml’

```
version: ‘3'
services:
  volume:
    image: edgexfoundry/docker-edgex-volume
    container_name: edgex-files
    networks:
      - edgex-network
    volumes:
      - /data/db
      - /edgex/logs
      - /consul/config
      - /consul/data
  …
logging:
  image: edgexfoundry/docker-support-logging
  ports:
    - "48061:48061"
  container_name: edgex-support-logging
  hostname: edgex-support-logging
  networks:
    - edgex-network
  volumes_from:
    - volume
  depends_on:
    - volume
    - config-seed
    - mongo
…
```

Containers

Container	Purpose
mongo	Mongo Database instance, and data initialization for the default NoSQL database for all of EdgeX
consul	Hashicorp's Consul configuration and registry service
data	Core Data, centralized persistence facility for data readings collected by devices and sensors
metadata	Core Metadata, knowledge about the devices and sensors and how to communicate with them
command	Core Command, enables the issuance of commands or actions to devices and sensors on behalf of other micro services, other applications, ex ternal systems
scheduler	Support Scheduling, provides facilities to kick off various events/actions on a timed schedule such as old data scrubbing
logging	Support Logging, central logging service for all micro services
notifications	Support Notifications, central alert and notification service for all micro services
rulesengine	Support Rules Engine, micro service "wrapped" Drools Rules Engine that monitors incoming sensor or device data for readings within target r anges and triggers immediate device actuation
export-client	Export Client, enables clients, whether they are on-gateway or off-gateway, to register as recipients of data coming through Core Data
export-distro	Export Distribution, receives data from Core Data, through a message queue, then filters, transforms, and formats the data per client request, and distributes to the appropriate endpoint by pre-registered protocol

EdgeX 기본 서비스들

- EdgeX relies on a shared file space among services (called a Docker volume)
  - Allows the database files to be shared across services
  - Allows log file space to be shared across services
- EdgeX use MongoDB as its default persistence storage
  - Mongo has been containerized for EdgeX use
- EdgeX uses Consul as its registry and configuration service
  - Consul has been containerized for EdgeX use
- EdgeX config-seed is a service that initializes Consul with EdgeX configuration data
  - config-seed exits quickly after populating Consul (i.e. it is not long running)
- EdgeX needs all micro services to be connected to a virtual network
  - Docker provides a virtual network facility
  - The Docker Compose file specifies the network and includes all the services and infrastructure on that network

EdgeX - 참고 Links

- EdgeX help, details and information
  - Documentation: https://docs.edgexfoundry.org/
  - The Getting started guides can be a great place to start: https://docs.edgexfoundry.org/Ch-GettingStarted.html
  - Wiki pages: https://wiki.edgexfoundry.org/
  - Github: https://github.com/edgexfoundry/
  - Rocket Chat: https://chat.edgexfoundry.org/channel/general
    - Question/answer forum with channels dedicates to particular EdgeX topics
  - Mailing Lists: https://lists.edgexfoundry.org/mailman/listinfo
    - Again, several lists for emailing the community

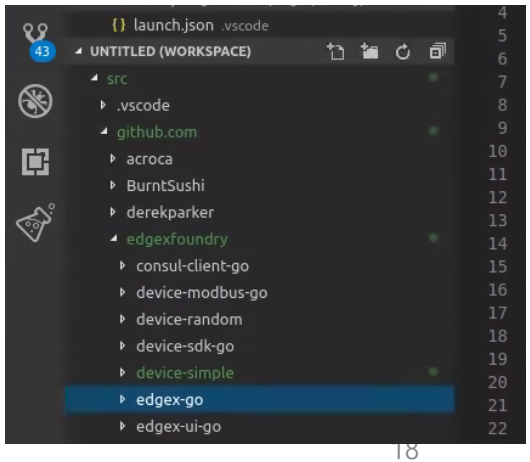
2. 개발자 접근 방법

- Go development environment
  - go, glide
  - Directory
  - IDE

- Source code download
  - git clone / go get
    - *go get github.com/edgexfoundry/edgex-go*
      - Mono repo.

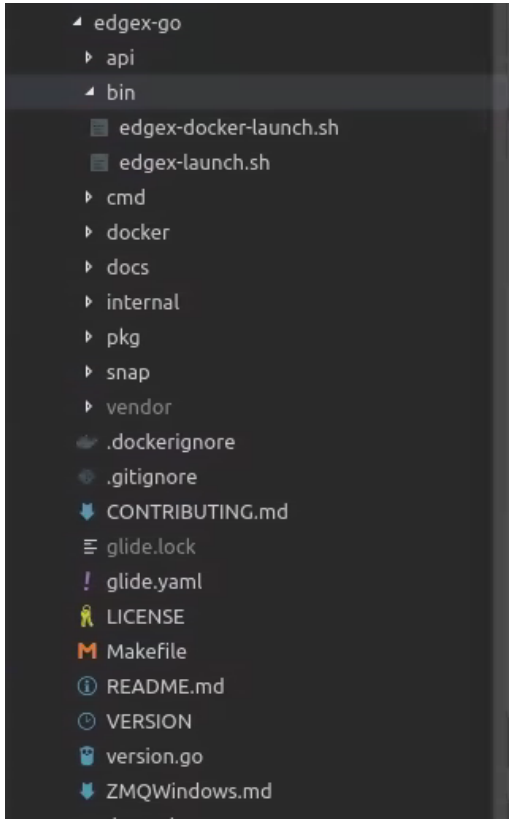
    - *make prepare*
    - *make build*
    - *make run*

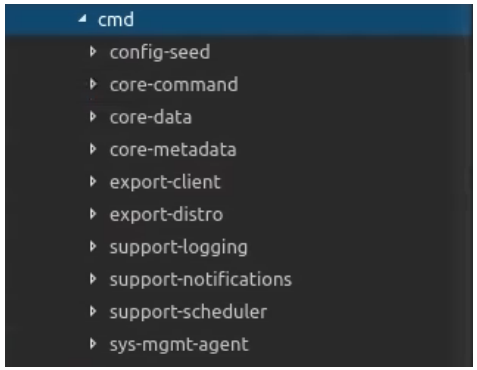
```
bin/
    hello
    outyet
src/
    github.com/golang/example/
        .git/
        hello/
            hello.go
        outyet/
            main.go
            main_test.go
        stringutil/
            reverse.go
            reverse_test.go
    golang.org/x/image/
        .git/
        bmp/
            reader.go
            writer.go
pkg/
```



EdgeX Core

- edgex-go
  - api
    - ‘raml’ 파일을 통해 functions 정의
  - bin
    - ‘docker-launch’, ‘edgex-launch’ 스크립트
  - cmd
    - go 기반의 components 정의/main.go 위치
  - docker
    - Dockerfiles
  - Internal
    - 관련 동작 및 설정 ‘.go’ 파일들
  - pkg
    - ‘go’ 를 만든 관련 package 파일들
  - snap
    - snap packaging

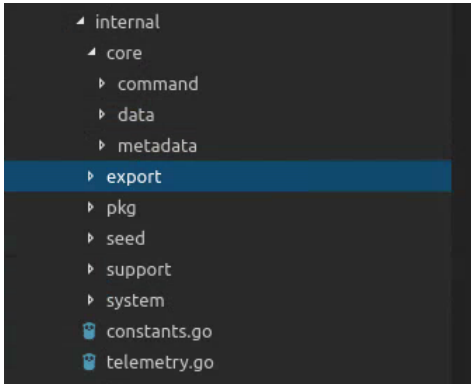


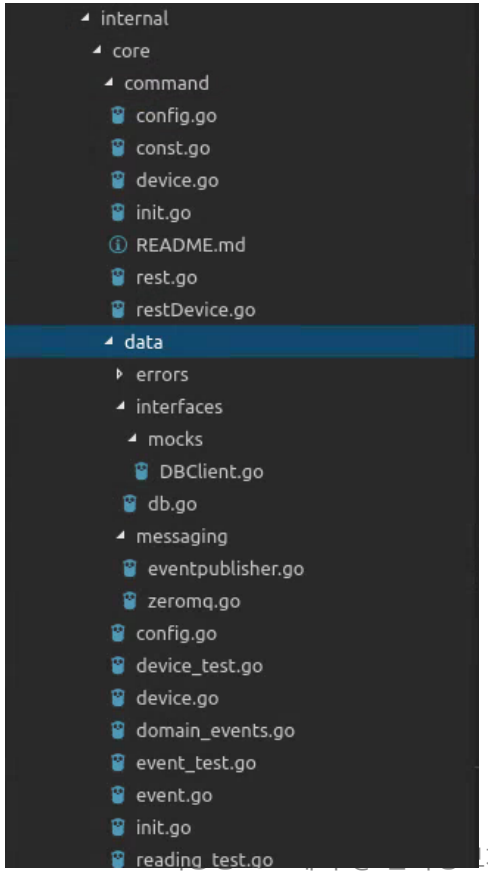


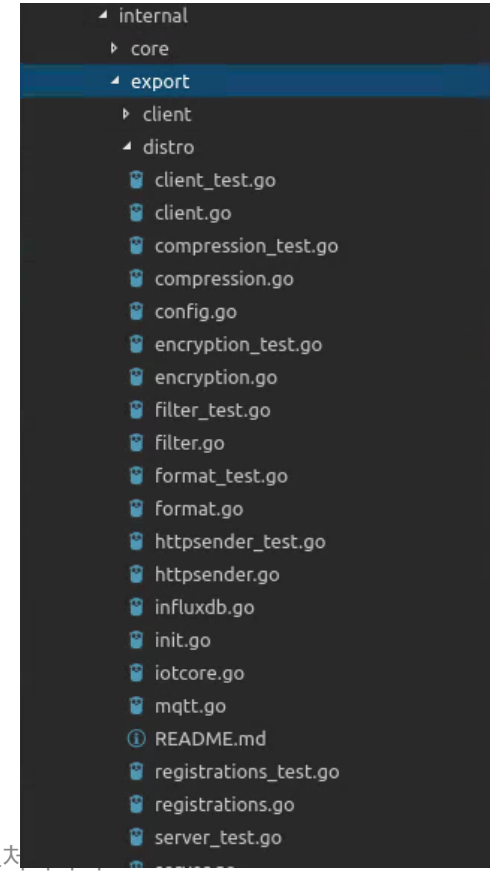
‘/api/raml’

```
title: core-data
title: core-metadata
#%RAML 0.8
title: core-comand
version: "1.0.0"
baseUri: "http://localhost:48082/api/v1"
schemas:
    -
        device: '{"type":"object","$schema":"http://json-schema.org/draft-03/schema#","description":"device or sensor su
    -
        addressable: '{"type":"object","$schema":"http://json-schema.org/draft-03/schema#","title":"addressable","proper
    -
        commandresponse: '{"type":"object","$schema":"http://json-schema.org/draft-03/schema#","title":"commandresponse"
/ping:
    displayName: Ping Resource
    description: Example - http://localhost:48082/api/v1/ping
    get:
        description: Test service providing an indication that the service is available.
        responses:
            "200":
                description: pong as a string
            "500":
                description: for unanticipated or unknown issues encountered.
/config:
    displayName: Config Resource
    description: Example - http://localhost:48082/api/v1/config
    get:
        description: Fetch the current state of the service's configuration.
        responses:
            "200":
                description: The service's configuration as JSON document
/metrics:
    displayName: Metrics Resource
    description: Example - http://localhost:48082/api/v1/metrics
    get:
        description: Fetch the current state of the service's metrics.
        responses:
            "200":
                description: The service's metrics as JSON document
/device/{id}/command/{commandid}:
    displayName: Issue command
    description: Example - http://localhost:48082/api/v1/device/57bd0f2d32d258ad3fcd2d4b/command/57bd0f1432d258ad3fcd2d4
    uriParameters:
```

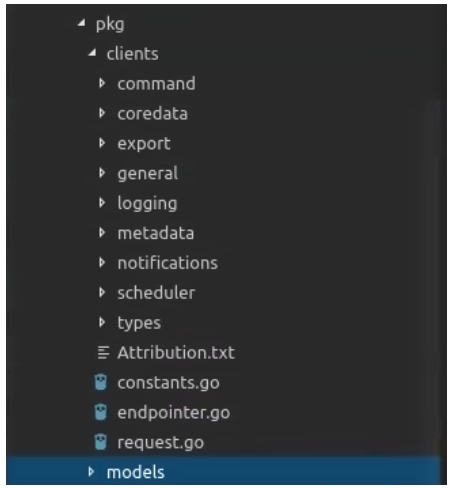
‘/edgex-go/internal’

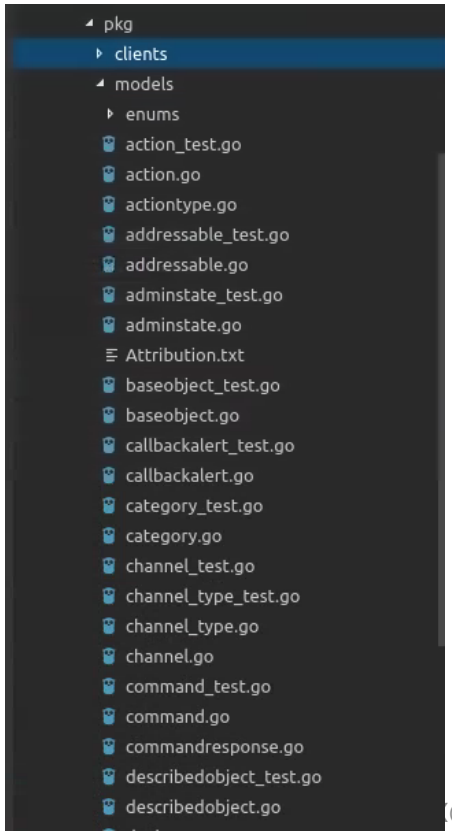


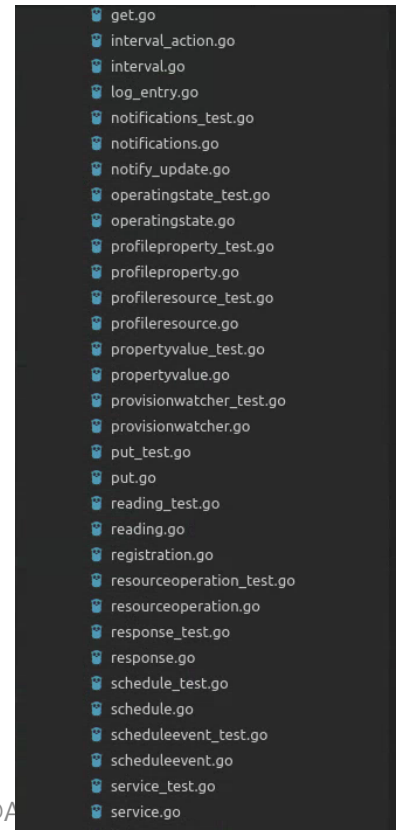


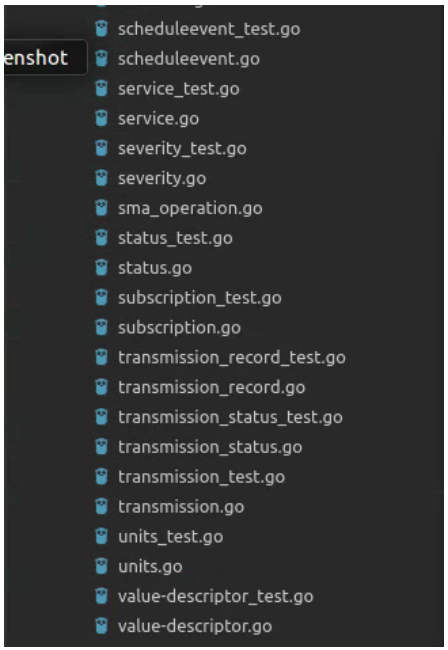


‘/edgex-go/pkg’


pkg
clients
command
coredata
export
general
logging
metadata
notifications
scheduler
types
Attribution.txt
constants.go
endpointer.go
request.go
models


pkg
clients
models
enums
action_test.go
action.go
actiontype.go
addressable_test.go
addressable.go
adminstate_test.go
adminstate.go
Attribution.txt
baseobject_test.go
baseobject.go
callbackalert_test.go
callbackalert.go
category_test.go
category.go
channel_test.go
channel_type_test.go
channel_type.go
channel.go
command_test.go
command.go
commandresponse.go
describedobject_test.go
describedobject.go


get.go
interval_action.go
interval.go
log_entry.go
notifications_test.go
notifications.go
notify_update.go
operatingstate_test.go
operatingstate.go
profileproperty_test.go
profileproperty.go
profileresource_test.go
profileresource.go
propertyvalue_test.go
propertyvalue.go
provisionwatcher_test.go
provisionwatcher.go
put_test.go
put.go
reading_test.go
reading.go
registration.go
resourceoperation_test.go
resourceoperation.go
response_test.go
response.go
schedule_test.go
schedule.go
scheduleevent_test.go
scheduleevent.go
service_test.go
service.go


scheduleevent_test.go
scheduleevent.go
service_test.go
service.go
severity_test.go
severity.go
sma_operation.go
status_test.go
status.go
subscription_test.go
subscription.go
transmission_record_test.go
transmission_record.go
transmission_status_test.go
transmission_status.go
transmission_test.go
transmission.go
units_test.go
units.go
value-descriptor_test.go
value-descriptor.go

APIs - Core Data

- Introduction
  - APIs to *expose the database to other services* as well as north-bound integration.
  - The database is secure. Direct access to the database is *restricted* to the Core Data service APIs.
  - Core Data also provides the REST API to create and register a new device.

- Codes
  - /api/raml/core-data.raml
  - https://docs.edgexfoundry.org/core-data.html

APIs - Metadata

- Introduction
  - includes the device/sensor *metadata database* and APIs to *expose the database* to other services.
  - In particular, the *device provisioning service* deposits and manages device metadata through this service.
  - This service may also hold and manage other configuration metadata used by other services on the gateway such as clean up schedules, hardware configuration (Wi-Fi connection info, MQTT queues, and so forth).
  - *Non-device metadata* may need to be held in a different database and/or managed by another service–depending upon implementation.

- Codes
  - /api/raml/core-metadata.raml
  - https://docs.edgexfoundry.org/core-metadata.html

APIs - Core Command

- Introduction
  - a conduit for other services to trigger action on devices and sensors through their managing Device Services.
  - The service provides an API to get the list of commands that can be issued for all devices or a single device. Commands are divided into two groups for each device:
    - GET commands are issued to a device or sensor to get a current value for a particular attribute on the device, such as the current temperature provided by a thermostat sensor, or the on/off status of a light.
    - PUT commands are issued to a device or sensor to change the current state or status of a device or one of its attributes, such as setting the speed in RPMs of a motor, or setting the brightness of a dimmer light.

- Codes
  - /api/raml/core-command
  - https://docs.edgexfoundry.org/core-command.html

APIs - rulesengine

- Introduction
  - Rules Engine Microservice receives data from the Export Service through 0MQ,
  - triggers actuation based on event data it receives and analyzes.
    - Built on Drools technology

- Codes
  - /support-rulesengine/raml/support-rulesengine.raml
  - https://docs.edgexfoundry.org/support-rulesengine.html

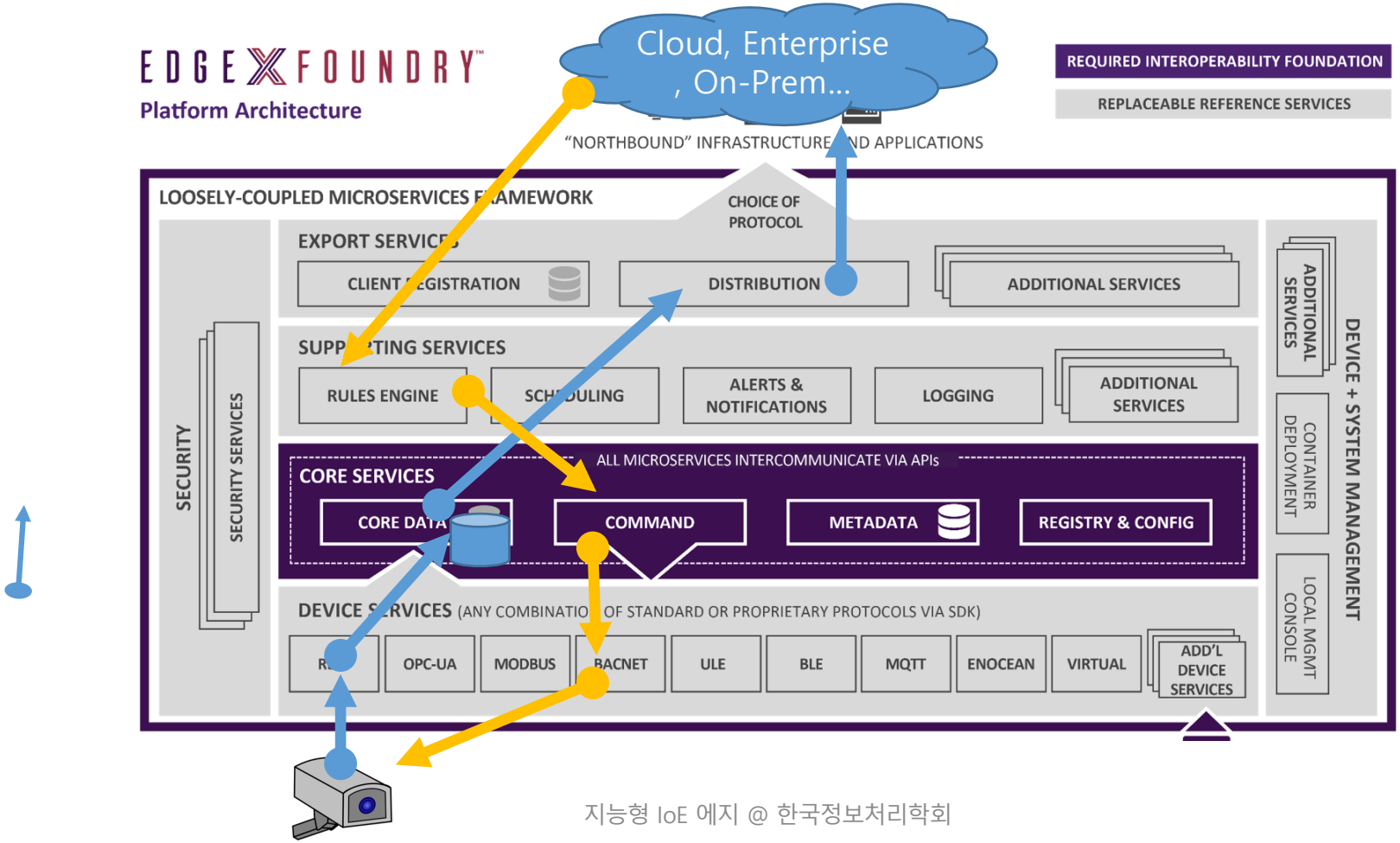
Work through by 사용자

사전 준비

- EdgeX
  - docker-compose.yaml
  - docker-compose pull
  - docker-compose up –d

- Postman
  - https://www.getpostman.com/downloads/

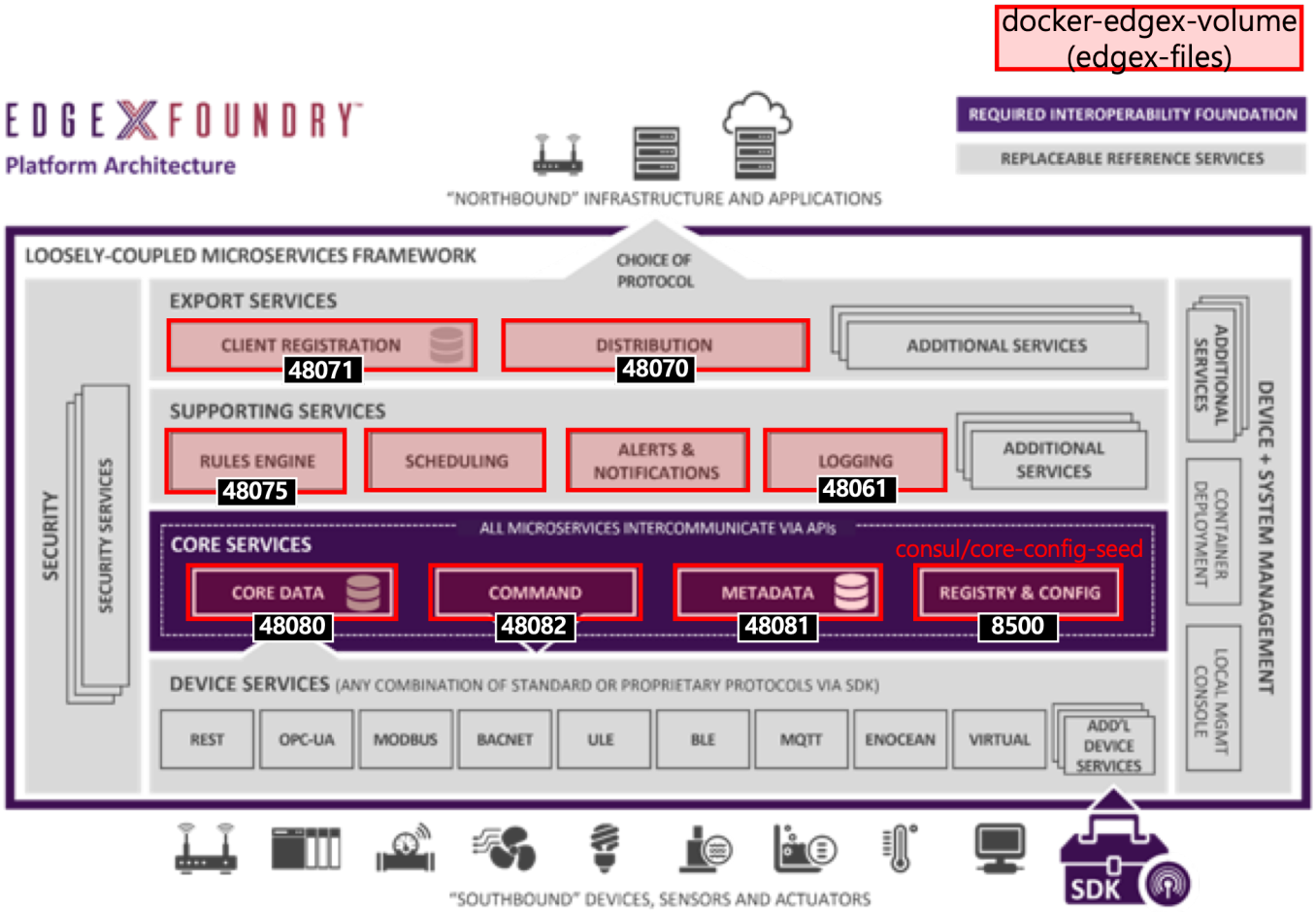
- Document
  - https://docs.edgexfoundry.org/Ch-Walkthrough.html


EDGEXFOUNDRY™
Platform Architecture
Cloud, Enterprise , On-Prem…
REQUIRED INTEROPERABILITY FOUNDATION
REPLACEABLE REFERENCE SERVICES
"NORTHBOUND" INFRASTRUCTURE AND APPLICATIONS
LOOSELY-COUPLED MICROSERVICES FRAMEWORK
CHOICE OF PROTOCOL
EXPORT SERVICES
CLIENT REGISTRATION
DISTRIBUTION
ADDITIONAL SERVICES
SUPPORTING SERVICES
RULES ENGINE
SCHEDULING
ALERTS & NOTIFICATIONS
LOGGING
ADDITIONAL SERVICES
ALL MICROSERVICES INTERCOMMUNICATE VIA APIs
CORE SERVICES
CORE DATA
COMMAND
METADATA
REGISTRY & CONFIG
DEVICE SERVICES (ANY COMBINATION OF STANDARD OR PROPRIETARY PROTOCOLS VIA SDK)
REST
OPC-UA
MODBUS
BACNET
ULE
BLE
MQTT
ENOCEAN
VIRTUAL
ADD'L DEVICE SERVICES
SECURITY
SECURITY SERVICES
DEVICE + SYSTEM MANAGEMENT
ADDITIONAL SERVICES
CONTAINER DEPLOYMENT
LOCAL MGMT CONSOLE

Device Service by Hybrid

사전 준비

- Go Lang 설치
  - Go
  - glide
  - Go IDE 설치
- EdgeX docker 실행
  - docker-compose up –d
- Documentation
  - https://docs.edgexfoundry.org/Ch-GettingStartedSDK-Go.html


docker-edgex-volume
(edgex-files)
EDGEXFOUNDRY
Platform Architecture
REQUIRED INTEROPERABILITY FOUNDATION
REPLACEABLE REFERENCE SERVICES
"NORTHBOUND" INFRASTRUCTURE AND APPLICATIONS
LOOSELY-COUPLED MICROSERVICES FRAMEWORK
CHOICE OF PROTOCOL
EXPORT SERVICES
CLIENT REGISTRATION
48071
DISTRIBUTION
48070
ADDITIONAL SERVICES
SUPPORTING SERVICES
RULES ENGINE
48075
SCHEDULING
ALERTS & NOTIFICATIONS
LOGGING
48061
ADDITIONAL SERVICES
ALL MICROSERVICES INTERCOMMUNICATE VIA APIs
CORE SERVICES
consul/core-config-seed
CORE DATA
48080
COMMAND
48082
METADATA
48081
REGISTRY & CONFIG
8500
DEVICE SERVICES (ANY COMBINATION OF STANDARD OR PROPRIETARY PROTOCOLS VIA SDK)
REST
OPC-UA
MODBUS
BACNET
ULE
BLE
MQTT
ENOCEAN
VIRTUAL
ADD'L DEVICE SERVICES
SECURITY
SECURITY SERVICES
ADDITIONAL SERVICES
CONTAINER DEPLOYMENT
LOCAL MGMT CONSOLE
DEVICE + SYSTEM MANAGEMENT
SDK
"SOUTHBOUND" DEVICES, SENSORS AND ACTUATORS

환경 설정

- Device SDK 다운

```
$ git clone https://github.com/edgexfoundry/device-sdk-go.git
```

- ‘device-sdk-go’ 를 ‘device-simple’ 로 수정

```
$ find . -type f | xargs sed -i 's/device-sdk-go/device-simple/g'
```

- 파일 수정
  - /cmd/device-simple
- 디바이스 서비스 빌드

```
$ make prepare
$ make build
```

실행 결과

- Device Profile 등록
  - 다운로드, https://docs.edgexfoundry.org/_downloads/random-generator-device.yaml (난수 발생 함수)
  - '/cmd/res' 디렉토리에 위치

- 실행 및 확인

```
$ ./device-simple
```

  - http://localhost:48080/api/v1/event/device/RandNum-Device-01/100

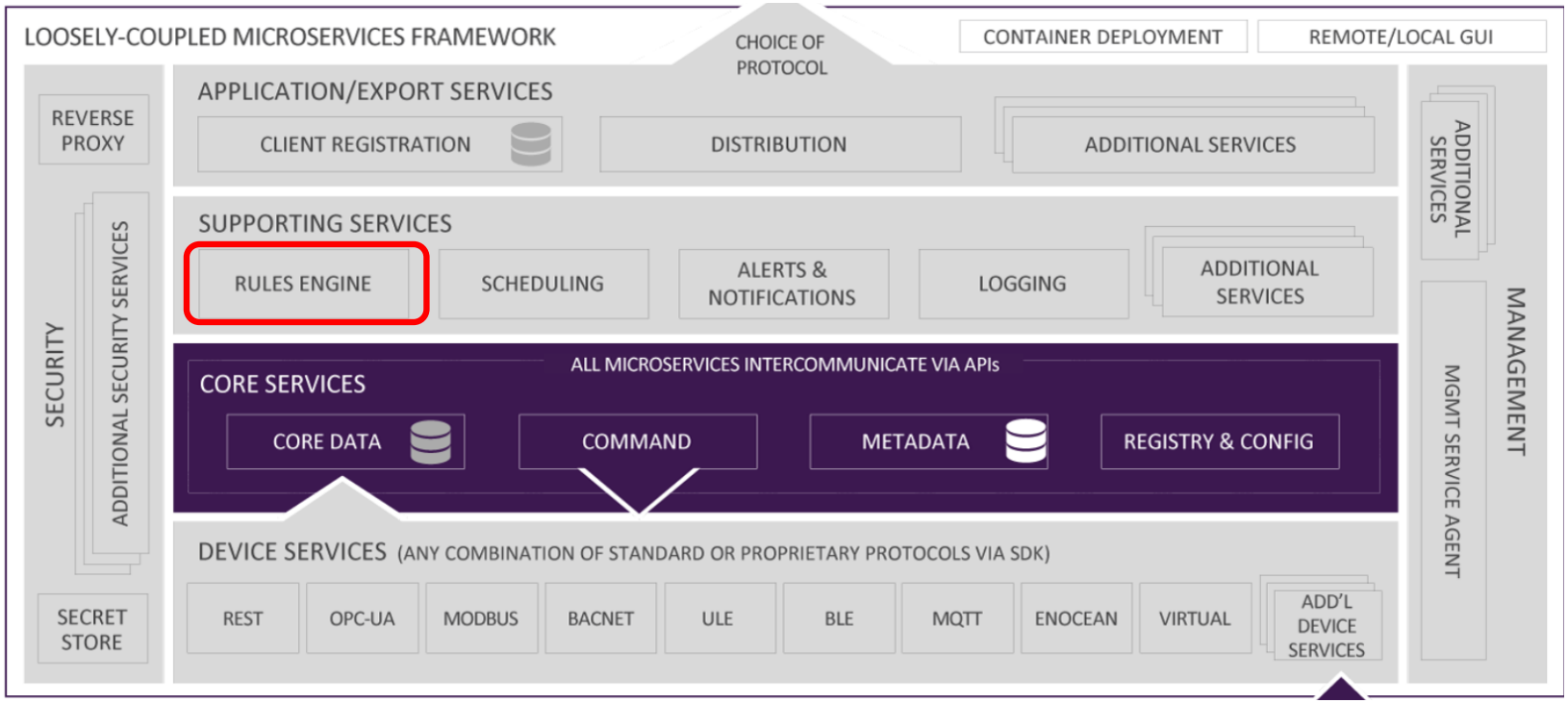
EdgeX + Tensorflow

Intelligent

- EdgeX support Intelligent
  - Rulesengine
    - Local analysis

- Interworking with other services
  - Cloud services

Local Analysis



Rules Engine

2 results for repositories matching **rules**

support-rulesengine

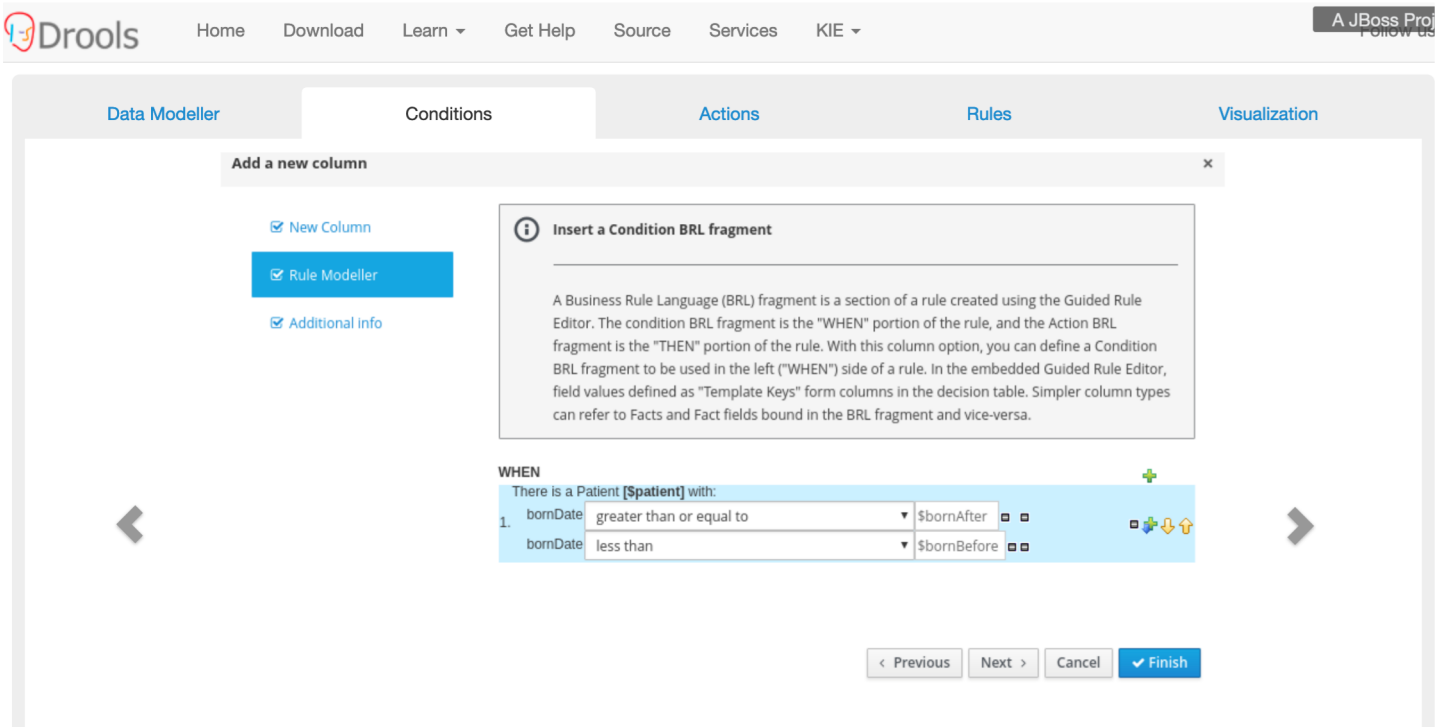
Java ★ 7 17 Apache-2.0 2 issues need help Updated Nov 17, 2018

docker-support-rulesengine Archived

2 Apache-2.0 Updated May 25, 2017

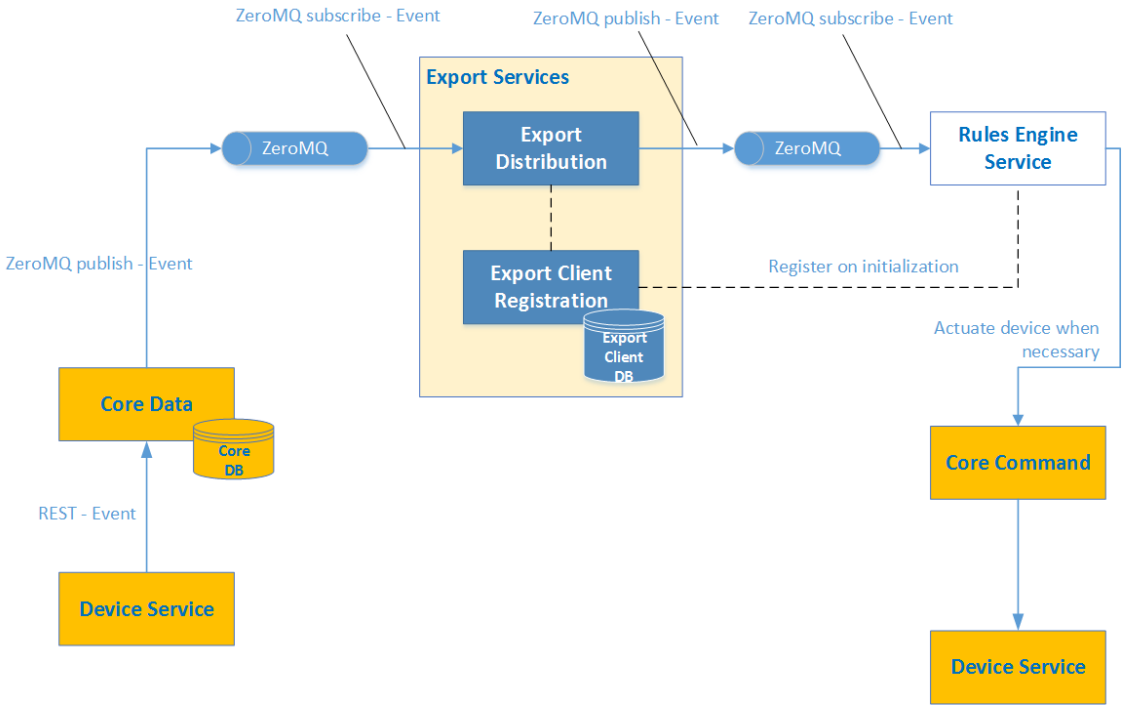
- Introduction
  - provides a reference implementation, edge-event triggering mechanism.
  - monitors incoming sensor or device data for readings within target ranges and triggers immediate device actuation.
  - The implementation **uses a Drools (https://www.drools.org/)** rules engine at its core.
    - Drools is an open source rules engine provided by the JBoss community. This microservice *is able to be replaced or augmented* by many other edge-analytic capabilities provided by 3rd parties.

Drools

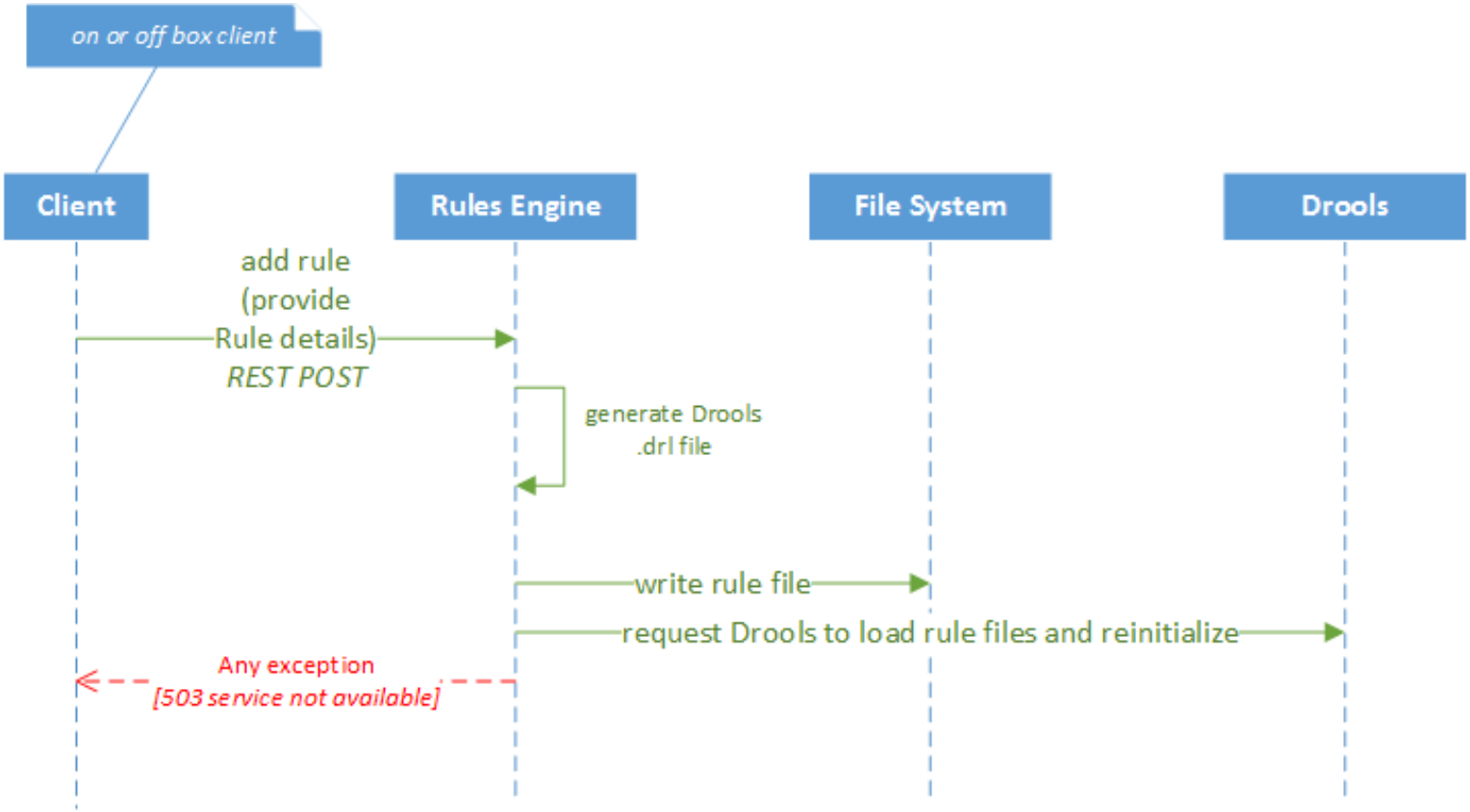

Drools
Home
Download
Learn
Get Help
Source
Services
KIE
Data Modeller
Conditions
Actions
Rules
Visualization
Add a new column
New Column
Rule Modeller
Additional info
Insert a Condition BRL fragment
A Business Rule Language (BRL) fragment is a section of a rule created using the Guided Rule Editor. The condition BRL fragment is the "WHEN" portion of the rule, and the Action BRL fragment is the "THEN" portion of the rule. With this column option, you can define a Condition BRL fragment to be used in the left ("WHEN") side of a rule. In the embedded Guided Rule Editor, field values defined as "Template Keys" form columns in the decision table. Simpler column types can refer to Facts and Fact fields bound in the BRL fragment and vice-versa.
WHEN
There is a Patient [$patient] with:
1. bornDate greater than or equal to $bornAfter
bornDate less than $bornBefore
Previous
Next
Cancel
Finish

Rules Engine as Export Service Client

- Rules engine as export service client
  - *receives all events* and readings through the Export Distribution.
  - is instructed *to monitor each event* and reading received through the Export Distribution, and the rules engine *triggers any actuation* to a device through the Core Command

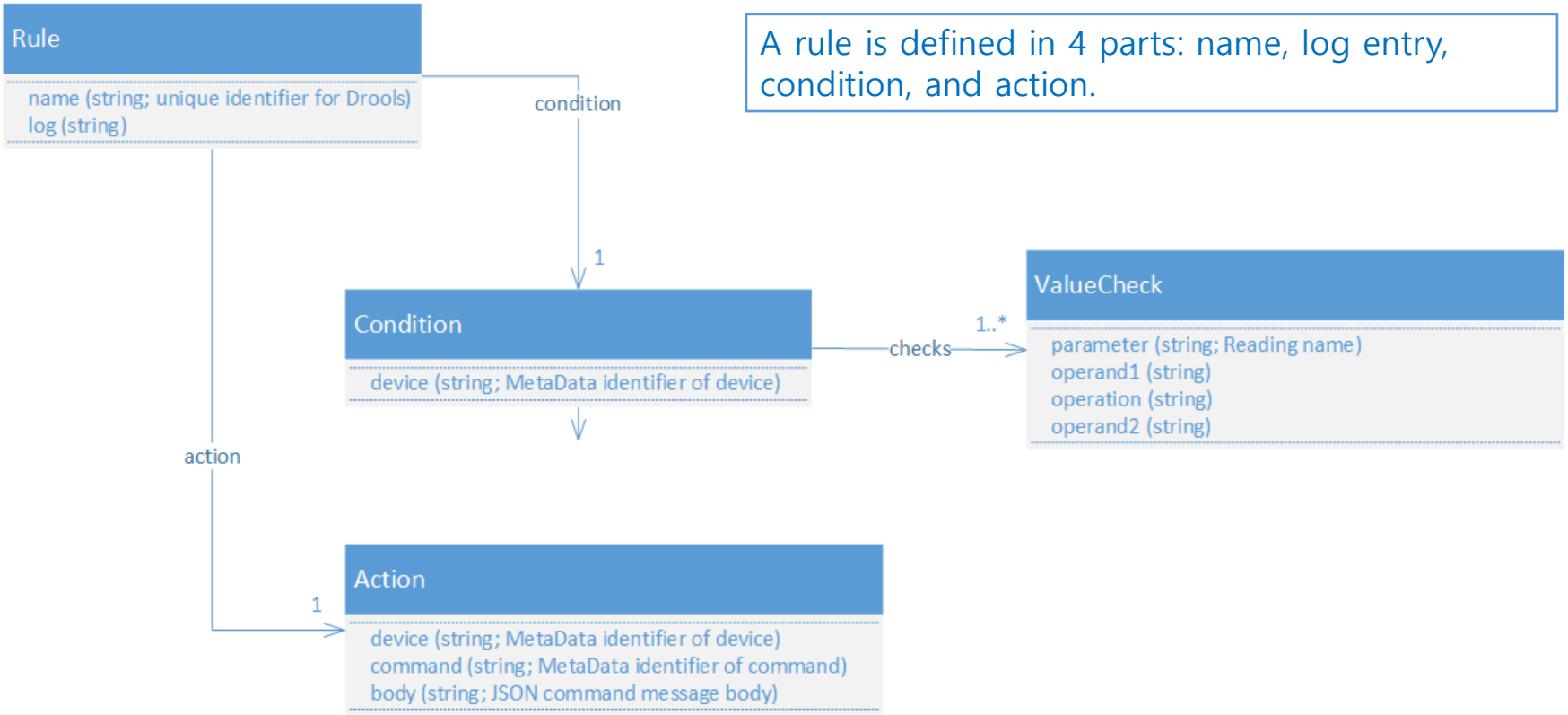


Rules Client and High Level Interaction Diagram



Rules (Defined), and Data Model

A rule is defined in 4 parts: name, log entry, condition, and action.

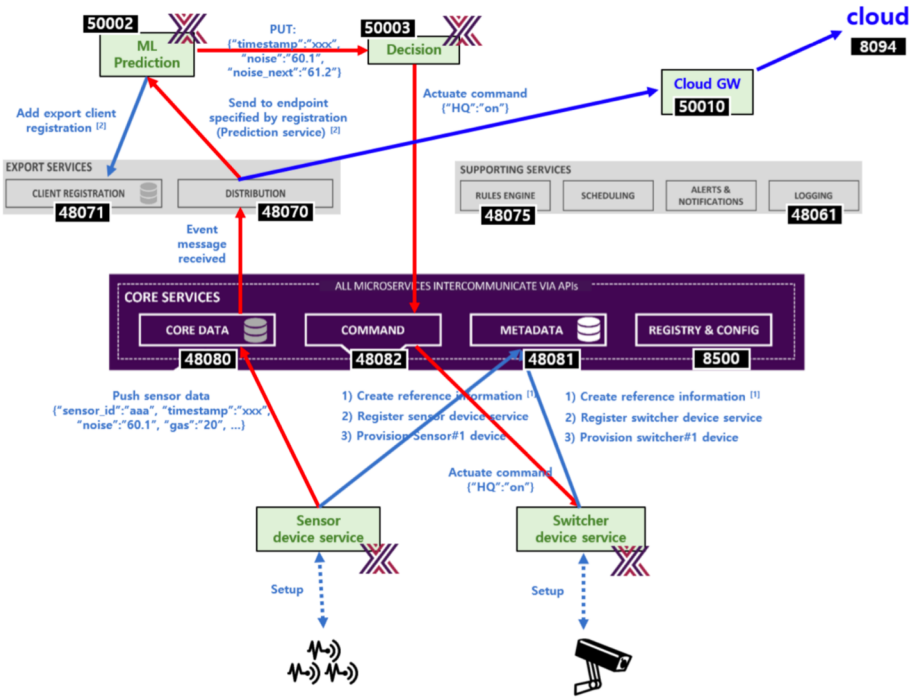


drule file

- ‘support-rulesengine/src/main/resources/rule-template.drl‘

```
package org.edgexfoundry.rules;
global org.edgexfoundry.engine.CommandExecutor executor;
...
import java.util.Map;
rule "${rulename}"
when
  $e:Event($rlist: readings && device=="${conddeviceid}")
  <#if valuechecks??>
  <#assign idx = 0>
  <#list valuechecks as valuecheck>
  $r${idx}:Reading(name=="${valuecheck.parameter}" && ${valuecheck.operand1} ${valuecheck.operation} ${valuecheck.operand2}) from $rlist
    <#assign idx = idx + 1>
  </#list>
  </#if>
then
executor.fireCommand("${actiondeviceid}", "${actioncommandid}", "${commandbody}");
logger.info("${log}");
end
```

Interaction with other services



사전 준비

- EdgeX docker images
- Tensorflow Serving 설치
  - https://www.tensorflow.org/tfx/serving/serving_advanced
- Mnist 학습 및 모델 이해하기

환경 설정

- Tensorflow MNIST serving

```
$ git clone https://github.com/tensorflow/serving.git
$ cd serving
$ tools/run_in_docker.sh python tensorflow_serving/example/mnist_saved_model.py
₩ /"SaveToYourDirectory"
```

- Docker-compose 를 통한 EdgeX 실행

```
$ wget –O docker-compose.ymal
https://github.com/mskim16/edgextfserving/blob/master/docker-compose.yml?raw=true
$ docker-compose up -d
```

실행 결과

- Docker를 이용한 테스트 및 추론 결과

```
$ tools/run_in_docker.sh python tensorflow_serving/example/mnist_client.py ₩ --
num_tests=1000 --server=127.0.0.1:8000
```

```
(tensorflow) mk-gpu@mkgpu:~/Documents/serving$ tools/run_in_docker.sh python tensorflow_serving/example/mnist_client.py --num_tests=
1000 --server=127.0.0.1:8000
== Pulling docker image: tensorflow/serving:nightly-devel
nightly-devel: Pulling from tensorflow/serving
Digest: sha256:08ea87f1fc6d585f59ba5935ad6b85898229b27159b56e10421077c670d64898
Status: Image is up to date for tensorflow/serving:nightly-devel
== Running cmd: sh -c 'cd /home/mk-gpu/Documents/serving; python tensorflow_serving/example/mnist_client.py --num_tests=1000 --serve
r=127.0.0.1:8000'
Extracting /tmp/train-images-idx3-ubyte.gz
Extracting /tmp/train-labels-idx1-ubyte.gz
Extracting /tmp/t10k-images-idx3-ubyte.gz
Extracting /tmp/t10k-labels-idx1-ubyte.gz

WARNING: The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md
  * https://github.com/tensorflow/addons
If you depend on functionality not listed there, please file an issue.

.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................
Inference error rate: 10.4%
```

감사합니다.