

This document instructs you to set up reThink testbed on Ubuntu. For more information about reThink project please refer to: https://rethink-project.eu/.

TODO: more introduction about reThink, hyperties and the testbed.

0. Prerequisites

- Ubuntu: 64-bit, 14.04 Trusty or later.
- A pre-installation of Java & Node.JS environment is recommended.

1. Install Docker

• Ubuntu:

1. Install Docker-Engine on Ubuntu following the link: https://docs.docker.com/engine/installation/linux/ubuntulinux/

Be careful that you should notice the version of your Ubuntu, and follow the instructions dedicated for the version that you are using.

Until now, you should have Docker-Engine installed, which can be verified by running the hello-world image with the command sudo docker run hello-world.

2. Install Docker-compose via the link: https://docs.docker.com/compose/install/

2. Get Testbed Code from GitHub

• Ubuntu:

1. Go to the folder where you want to put the testbed:

\$ cd ~

Go to reTHINK H2020 Project GitHub page: https://github.com/reTHINK-project/, and you will see a list of project repositories.

2. Find the "testbeds" repository, git clone the repository into your local folder as below:

\$ git clone git@github.com:reTHINK-project/testbeds.git

3. Find the "dev-hyperty-toolkit" and "dev-hyperty" repositories, and git clone them into the PT-node directory:

```
$ cd testbeds/nodes/PT-node/
$ git clone -b develop git@github.com:reTHINK-project/dev-hyperty-toolkit.git
$ git clone -b develop git@github.com:reTHINK-project/dev-hyperty.git
```

NOTE

- We have two folders natively under "PT-node", of which one is for production and the other is for development
- The **development** folder contains all catalogue_objects
- The production folder only contains the specific catalogue_objects for our testbed

3. Testbed Startup

- Ubuntu:
 - 1. Go to development folder and we can start all services on docker-compose.yml without the localhost support:

```
$ sudo docker-compose up
```

Then you should see something like:

With the default configuration, the toolkit would be in production mode, and the catalogue is configured to use the remote one with a default domain name of hybroker.rethink.ptinovacao.pt.

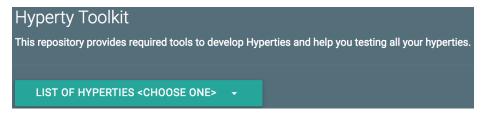
2. Open the Google Chrome browser, check the reTHINK Catalogue Test Page with the link: https://catalogue.hvbroker.rethink.ptinovacao.pt/:

reTHINK Catalogue Test Page

Conntected Databases



Check the Hyperty Toolkit Page via: https://hybroker.rethink.ptinovacao.pt/:



NOTE

- Use web console to see ongoing events.
- To run the toolkit in development mode, and catalogue local, you need to make some changes to the compose file.
- To stop the testbed, press ctrl+c, then execute command sudo docker-compose down.

4. Testbed Configuration

• Ubuntu:

1. Go to the "development" folder, and open the "docker-compose.yml" file with an editor:

```
$ cd development
$ gedit docker-compose.yml
```

The compose file provides a way to document and configure all of the application's service dependencies (msg-node, domain-registry, catalogue components, toolkit, etc);

All this services are compiled in the docker-compose.yml file;

This yml file is split by services, for example:

```
version: '2'
services:
  'msg-node-vertx':
    build: './dev-msg-node-vertx'
    container_name: 'msg-node-vertx'
    networks:
    rethink:
        ipv4_address: 172.18.0.2
    expose:
        - '443'
        - '9090'
```

For a better understanding about this file, please refer to: https://docs.docker.com/compose/compose-file/.

2. Use a customized local domain to replace the default one. First you need to register the url you want to use, for example hjiang-rethink.com, to the google apps.

Replace the default domain name with your local domain in the docker-compose.yml file:



Set DEVELOPMENT mode of the "hyperty-toolkit" service to "true":

```
'hyperty-toolkit':
build: '../dev-hyperty-toolkit'
container_name: 'hyperty-toolkit'
environment:

- DEVELOMENT=true

- RUNITME_URL=hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime

- DOMAIN=hjiang-rethink.com

- HYPERTY_REPO=/tmp/dev-hyperty
volumes:

- ../dev-hyperty:/tmp/dev-hyperty
networks:

rethink:
    ipv4_address: 172.18.6.32
expose:
    - '443'
```

Open /etc/hosts file and add the following lines at the end:

```
172.18.0.2 msg-node.hjiang-rethink.com
172.18.0.4 registry.hjiang-rethink.com
172.18.0.8 catalogue.hjiang-rethink.com
172.18.0.32 hjiang-rethink.com
```

Open ./dev-hyperty-toolkit/system.config.json, replace "localhost" with your local domain:

```
{
  "develop" : {
    "development": true,
    "runtimeURL": "hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime",
    "domain": "hjiang-rethink.com"
},
  "production" : {
    "development": false,
    "runtimeURL": "hyperty-catalogue://catalogue.hysmart.rethink.ptinovacao.pt/.well-known/runtime/Runtime",
    "domain": "hysmart.rethink.ptinovacao.pt"
}
```

Open ./dev-hyperty-toolkit/resources/descriptors/ProtoStubs.json, make default configuration url to your local domain with port 9090:

Open ./development/cataloguedatabase/catalogue_objects/protocolstub/VertxProtostub/desc-ription.json, and again replace the default configuration url to your local domain with port 9090:

```
"cguid": "40001",
"version": "1.0",
"objectName": "VertxProtoStub",
"description": "This is a protostub",
"language": "Javascript ECMA6",
"dataObjects": [],
"configuration": {
"url": "wss://msg-node.hjiang-rethink.com:9090/ws"
},
"constraints": [],
"messageSchemas": []
```

3. Rebuild catalogue-databases:

Use sudo docker images to list all the docker images, and use sudo docker rmi followed with image ID to remove an image, in this case we look for image ID of development catalogue-database-ptinovacao:

```
Haos-MacBook-Pro:~ hjiang$ sudo docker images
REPOSITORY
                                             TAG
                                                                 IMAGE ID
                                                                                     CREATED
                                                                                                           SIZE
                                                                 8f2acc30dff9
                                                                                                          775.1 MB
                                             latest
                                                                                     About a minute ago
production_catalogue-database-ptinovacao
                                                                                     6 days ago
production_hyperty-toolkit
                                             latest
                                                                 af809dd0d62e
                                                                                                           1.072 GB
rethink/registry-domain-server
                                             latest
                                                                 4faed9e8c4f7
                                                                                     7 days ago
                                                                                                          842.7 MB
                                                                 36dc1bb7a52b
                                                                                                          655.5 MB
                                             latest
                                                                                     2 weeks ago
node
rethink/catalogue-broker
                                             latest
                                                                 93d8bb3ab543
                                                                                     5 weeks ago
                                                                                                           765.9 MB
rethink/catalogue-database-rethinkdefault
                                                                 4198229b0e3b
                                             latest
                                                                                     5 weeks ago
rethink/msg-node-vertx
                                                                 8084f7036cfe
                                                                                     6 weeks ago
                                                                                                          1.292 GB
                                             latest
                                                                                                          1.848 kB
                                                                 c54a2cc56cbb
                                                                                     5 months ago
hello-world
                                             latest
[Haos-MacBook-Pro:~ hjiang$ sudo docker rmi 8f2acc30dff9
Untagged: production_catalogue-database-ptinovacao:latest
Deleted: sha256:8f2acc30dff9ec3e1cf7e21ff8a09075d4a001a35b7515b42123594bb0678e0f
Deleted: sha256:06057a65d4b11a642abfb5d077f0b467ba5adfa2bf6b458c3bfd8a0c64c90ed5
Deleted: sha256:ed481fb7527a17ce6a339e7466f32031d62ca0731a531682e7e4d58eb61b76ca
Deleted: sha256:7f69706b6c0468143c33882f9f7514f2039d787caa73ee6971a2ec585f1aafae
```

4. Restart the testbed:

```
$ sudo docker-compose up
```

5. Change Default Message Node (optional)

The default message node is Vertx, and this section instructs you to set Nodejs message node as the default.

• Ubuntu:

1. Build Nodejs message node image. Download Nodejs message node source code from reThink Github:

```
$ git clone git@github.com:reTHINK-project/dev-msg-node-nodejs.git
```

Make sure there is a folder called ssl/ under dev-msg-node-nodejs/src/ containing three files:

myCert.crt:

```
----BEGIN CERTIFICATE----
MIIDBjCCAe4CCQDTAKAOrXDViDANBgkqhkiG9w0BAQsFADBFMQswCQYDVQQGEwJB
VTETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50ZXJuZXQgV21kZ210
cyBQdHkgTHRkMB4XDTE2MTIyMzA5MjExNVoXDTE3MTIyMzA5MjExNVowRTELMAkG
A1UEBhMCQVUxEzARBgNVBAgMClNvbWUtU3RhdGUxITAfBgNVBAoMGEludGVybmV0
IFdpZGdpdHMgUHR5IEx0ZDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AKILljk9A5Ck8O+KwuhHLODog1eoZbCAp72UOO1x51uliuwY5CcDe8CZizxMQUrV
+KIK3qBi2x3P7CM+HFCA6iGCAUqykW27E5rCGsBqGSI9fxfzESwTR7dYPTgfSdMs
GyfE4jA8GGyXVW/44F/RTw0DlbssmykO+5JNAYk5j3cQRhr14JEKmfz2uSsETR/x
bCyI//2POWCKX5Fa6M0TAOAiokxSXbphKGnLnsWTb0doshtcf3OMaE/cKHkIWFAW
BA9ayhHOfvIUIyVOB7z10UzBJBc26H9o7lEcO6qlH+q7t6j2AJpNlNa6zBWIaivG
yji+Bw/5QeTNC6qhdDaNvzcCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAkYvJNAtx
c5eupdwQJiCtiQ7NkuPfiWEm/+NSTqJrLYZNkLKXsl1yG+Cxow32vYLe9hjagtBU
nm2ReBQyck9K5fbzxgR4gOnGwJv3/Rf6HV6743KVcwqujtqO9Va/cqSIPWPMpnQX
78bU6EfV0GZCGp5pU3058DMQK09WsRWEF9ewTh/Tan1JMWjbJhdPKvvXisKRxmzX
UjokSoXShYHzM4YXImEEW0hkFqkPvwqyxC9oww5iWf4pboVonNoR9SqXIPakUZYt
kjNcCSiUSBqT2CRQiqYnfq7wBU7WyxnjutTsxCIckXmVFFH580x2qbUrz2CCXT1n
8g+h62g990g3Kw==
----END CERTIFICATE----
```

myCert.csr:

----BEGIN CERTIFICATE REOUEST----

MIICijCCAXICAQAwRTELMAkGA1UEBhMCQVUxEzARBgNVBAgMC1NvbWUtU3RhdGUx ITAfBgNVBAoMGEludGVybmV0IFdpZGdpdHMgUHR5IEx0ZDCCASIwDQYJKoZIhvcN AQEBBQADggEPADCCAQoCggEBAKILljk9A5Ck8O+KwuhHLODoq1eoZbCAp72UOO1x 51uliuwY5CcDe8CZizxMQUrV+KIK3qBi2x3P7CM+HFCA6iGCAUqykW27E5rCGsBq GSI9fxfzESwTR7dYPTgfSdMsGyfE4jA8GGyXVW/44F/RTw0Dlbssmyk0+5JNAYk5 j3cQRhr14JEKmfz2uSsETR/xbCyI//2POWCKX5Fa6M0TAOAiokxSXbphKGnLnsWT b0doshtcf30MaE/cKHkIWFAWBA9ayhHOfvIUIyVOB7z1OUzBJBc26H9o71EcO6q1 H+q7t6j2AJpNlNa6zBWIaivGyji+Bw/5QeTNC6qhdDaNvzcCAwEAAaAAMA0GCSqG SIb3DQEBCwUAA4IBAQB9vw1Matn4XLGUsD2ZHZS4fdvtrjcpdAaBhwtbJIsYRvmG OihvLW0pxNHIQuKvxYTLA4yaSUCt6MKQfUyYCwZyh5705EodVsk7bKCDiGir7NQG 5SFqwMaWHiKTexATHCzLi0tZDTfgMqmJJZJhNigFOES82wUifn/Du+UOUxoQuQtE 23PmdTesTP04T8C9SeBCRJxWginzKT5IHVGX9fk4+HNwAZ5L0qBqZMB5SqTvrQd2 uNvXreWCnJvVTtRzculJxM/OQu/P9wcRsfbReQvaaiRrNlgrrLprU5jACk6kYx4E oBi+EIK9ptKqPyzZK641zMzYhoyGla40WRefz7bY

----END CERTIFICATE REQUEST----

myCert.key:

----BEGIN RSA PRIVATE KEY----

MIIEowIBAAKCAQEAoguWOT0DkKTw74rC6Ecs40irV6hlsICnvZQ47XHnW6WK7Bjk JwN7wJmLPExBStX4ogreoGLbHc/sIz4cUIDqIYIBSrKRbbsTmsIawGoZIj1/F/MR LBNHt1g90B9J0ywbJ8TiMDwYbJdVb/jgX9FPDQOVuyybKQ77kk0BiTmPdxBGGvXg kQqZ/Pa5KwRNH/FsLIj//Y85YIpfkVrozRMA4CKiTFJdumEoacuexZNvR2iyG1x/ c4xoT9woeQhYUBYED1rKEc5+8hQjJU4HvOU5TMEkFzbof2juURw7qqUf6ru3qPYA mk2U1rrMFYhqK8bKOL4HD/1B5M0LqqF0No2/NwIDAQABAoIBABQoDM+h30emvLGP OR9beyGez3Ff6QZDepoCTfJ7MKQdu2IS1o5jyIj0kHU3pweABiVh5L9ejJLwwhbA KxV4c6cnXLHF9IEJvC6L5nf4upefvqdj/MbWvxz3R0YuPiieW+JegyQs1Xzsdhwv gPlqN8MOpxlZOxOoYXKZUY+YMVhF40bbQSiRMc6YhQf4D5nBH1vkGNiwPaU1vmCT ph0zaMpYOE4SOCIdRWdR9/01KWDCqAi5yIChHjiY5EUGrnlC9zCDqQ/MXYw0cT1r ogoamLN322R24v2dXT7EnxFJNaR0b0ik1oMttzoKT2kQRqb5Z0qa76Zvpw4eu0vL E5UhrcECgYEAzIGkcNb+7PLAG810YOSDhKo2TepKyfcQp+n618G0hieqCPTH02KD z93bAbufF4Gk4nZqHDqkMZ7NOdp6ITi9E+pVt6ypVUPOQNw+WNMqfE2qCtL1qycf lWCFIiw0/ce0zhXp8wPo0d20GSvL9zbsBFQXX1ltRBx+3WGTYRtVeUsCgYEAytjs sa3pr48KCK055CLJUN3piL1LRUk2RqeQF4yz9aQXCjJHxT7SzzuRK0JR16Y8H63v FUQpUUSOZwW3jsVm/hqhSQ3MkCnQY9wgiEuP8Q+nVq5zwNim9xU18V0EI7fWMkVc +X0jXfy/DcBLJ5ysGWz/RG94ucUb9zAO2REBakUCgYAHy4E7E9bPrUkJRhp7fvpM WjiLq7yCCjrGS7LV+nopCuX+RWZAdmbTy38Hh+E+giCIyT6uKMYNZiIobF1vWrxK QApYX+psZ5M7AaoWtKXP6x3qH7Rhe7JGfVFVxh2W9r5IuKkcLC/5n9u3geXaXGuO lq31YnE5wWRMh+WYM/SEFQKBgQCyeeDwiKOnK0TZyyzhFNZqx4YWHWqkAMG3svEu N0eT5xe6ifjqDNexLKkTFdvygXo/KWvGG1370kXNUZNrFDcFdurX+0M0yQWpB2K8 aTeyE2LZ+s6wLC/kEI6Q2pWs07CsNKmUfs1igeKcN8goh0pJaMTFMwtW8J4lh0xb gwHqvQKBgDF70etJoAWYB+YDk3TLogrIrh7hX2GxdFzqgkMy7Nmt4PjwNyfYevgW A29n/OHwvT3yKNGOZmFkIoIL2pL8z9r6ipo91zxT97zJTBRDSMYBX3nMEk9kDmb3 KLF8bV1qyQ6Pz2x6TzrLYpxEL4zgIgpCQj0VawkCUeD8EwhP7tLg ----END RSA PRIVATE KEY----

If there is not, create a folder named "ssl" and past the three files in it.

After that, under dev-msg-node-nodejs/, use the command below to build the nodejs message node image:

\$ sudo docker build -t msg-node-rethink .

For detailed information refer to: https://github.com/reTHINK-project/dev-msg-node-nodeis/tree/master

When it finishes, check the docker images that you just built: msg-node-rethink and node:

```
- - sudo docker images
REPOSITORY
development catalogue-database-ptinovacao latest 059cbc433be2 2 hours ago 775.6 MB
msg-node-rethink
rethink/catalogue-database-rethinkdefault
rethink/catalogue-database-rethinkdefault
dev 5e474263c15d 2 days ago 770.7 MB
rethink/registry-domain-server latest 4faedge8c4f7 8 days ago 842.7 MB
vitormtsilva/dev-hyperty-toolkit develop 9ed239017922 3 weeks ago 994.4 MB
rethink/catalogue-broker latest 93d8bb3ab543 6 weeks ago 765.9 MB
node 6.2 9121f2a78999 6 months ago 659.4 MB
```

2. Open development/docker-compose.yml, change the default protocolstub under "catalogue-broker" service to "NodejsProtoStub":

```
'catalogue-broker':
image: rethink/catalogue-broker
container_name: 'catalogue-broker'
networks:
rethink:
ipv4_address: 172.18.0.8
hostname: 'catalogue.hjiang-rethink.com'
command: [-host, 'catalogue.hjiang-rethink.com', -default, protocolstub/NodejsProtoStub]
expose:
- '443'
- '5683'
```

Comment the "msg-node-vertx" service, and add "msg-node-nodejs" service:

If you are using different local domain, please remember to change the url to your own domain.

Open ./development/cataloguedatabase/catalogue_objects/protocolstub/NodeProtostub/description.json, and again replace the default configuration url to your local domain with port 9090:

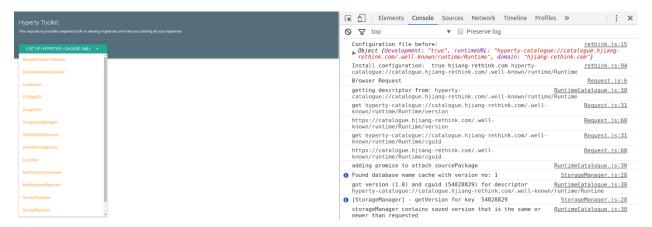
3. Start your testbed and verify your configuration:

You would see a message showing "[INFO] server – [S] HTTP & WS server listening on 9090", which can also be verified by using command:

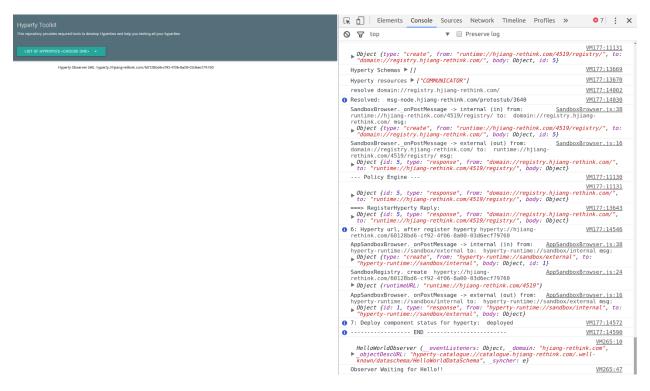
```
$ sudo docker ps
```

6. Play with Hyperties

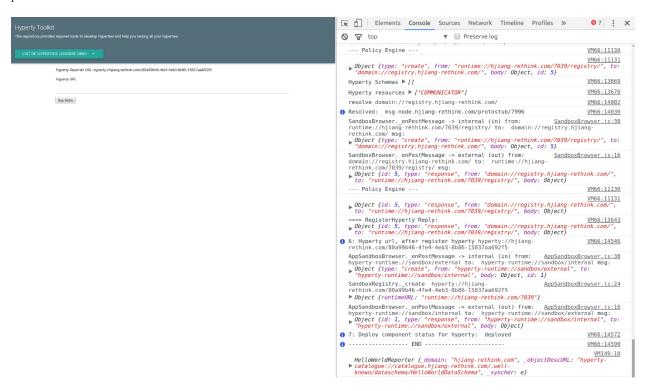
This section shows you some example hyperties. First open https://hjiang-rethink.com, and you will see a list of hyperties under the dropdown menu:



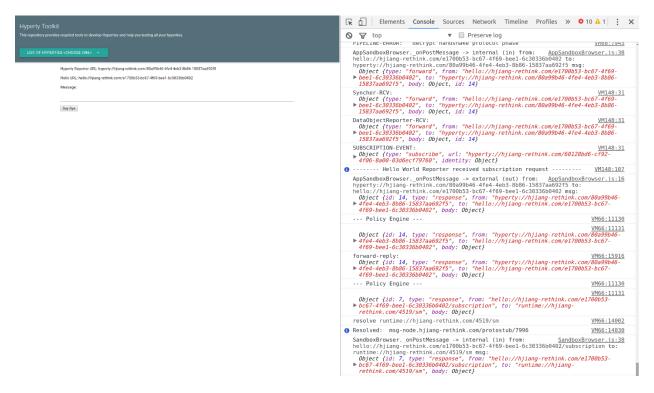
Select "HelloWorldObserver", and log in with a google account. You will get an observer URL started with "hyperty://" as below:



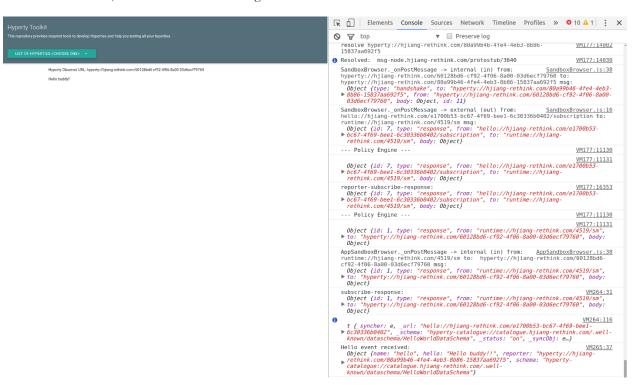
Open a new hyperty page using the same address: https://hjiang-rethink.com, and select "HelloWorldReporter". Log in with another google account other than the previous one. You will then get a reporter URL and be requested to provide the observer URL:



Copy the previous observer URL to the corresponding field, and click "Say Hello", and on the reporter side, you would see:



On the observer side, we can see the hello message received:



We can continue to send "Say Bye" message from the reporter side and see similar results. Until now, we have basically set up a functional local reThink testbed, based on which we can develop interesting hyperties, architectures, and policy engine, etc.

7.	Development	