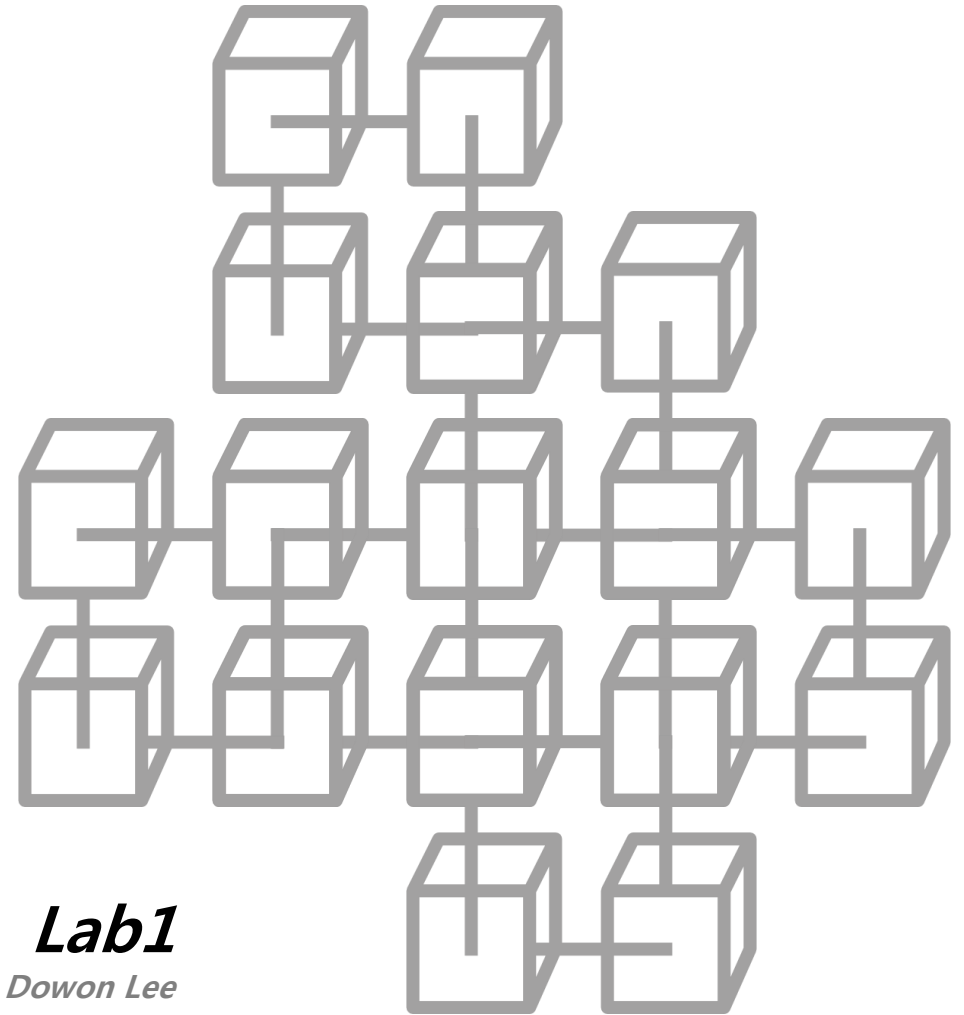




# HYPERLEDGER FABRIC



## *Lab1*

*written by Dowon Lee*

- 가상화 환경 SW
  - VirtualBox (<https://www.virtualbox.org>)
  - Vagrant (<https://www.vagrantup.com>)
- Linux Guest OS (Hyperledger Fabric을 구축하는 OS 환경)
  - Ubuntu 16.04.3 LTS x86/64bit (<https://www.ubuntu.com>)

- 필요한 Tool 또는 SW (Guest OS에 설치)
  - Docker Community Edition(CE) 17.03.0-ce 이상 (<https://www.docker.com>)
  - Docker Compose 1.8.0 이상
  - Python 2.7 (<https://www.python.org>)
  - Go 언어 1.7.x (<https://golang.org>)
  - Node.js 6.x (<https://nodejs.org>)
  - npm 3.10.10
  - GNU make, gcc/g++, libtool
- 설치 순서
  - 1) VirtualBox와 Vagrant의 인스톨
  - 2) Ubuntu Linux 게스트 OS의 인스톨과 초기설정
  - 3) 필요한 툴 또는 소프트웨어의 인스톨
  - 4) HLF의 인스톨과 동작 확인

# Step1. VirtualBox와 Vagrant의 인스톨

- VirtualBox 인스톨
  - <https://www.virtualbox.org>



The screenshot shows the VirtualBox.org homepage. On the left is a sidebar with navigation links: About, Screenshots, Downloads, Documentation (with sub-links for End-user docs and Technical docs), Contribute, and Community. The main content area features the VirtualBox logo, a 'Welcome to VirtualBox.org!' heading, and a paragraph describing the product as a powerful x86 and AMD64/Intel64 virtualization product. Below this is a large green button that says 'Download VirtualBox 6.0'. To the right of the main text is a 'News Flash' section with three entries: 'New January 28th, 2019 VirtualBox 6.0.4 released!', 'New January 28th, 2019 VirtualBox 5.2.26 released!', and 'New December 18th, 2018 VirtualBox 6.0 released!'. Each entry includes a brief description and a link to the Changelog. At the top right of the page are search, login, and preferences links.

**VirtualBox**

search...  
Login Preferences

## Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 [virtualization](#) product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "[About VirtualBox](#)" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of [guest operating systems](#) including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

**Download VirtualBox 6.0**

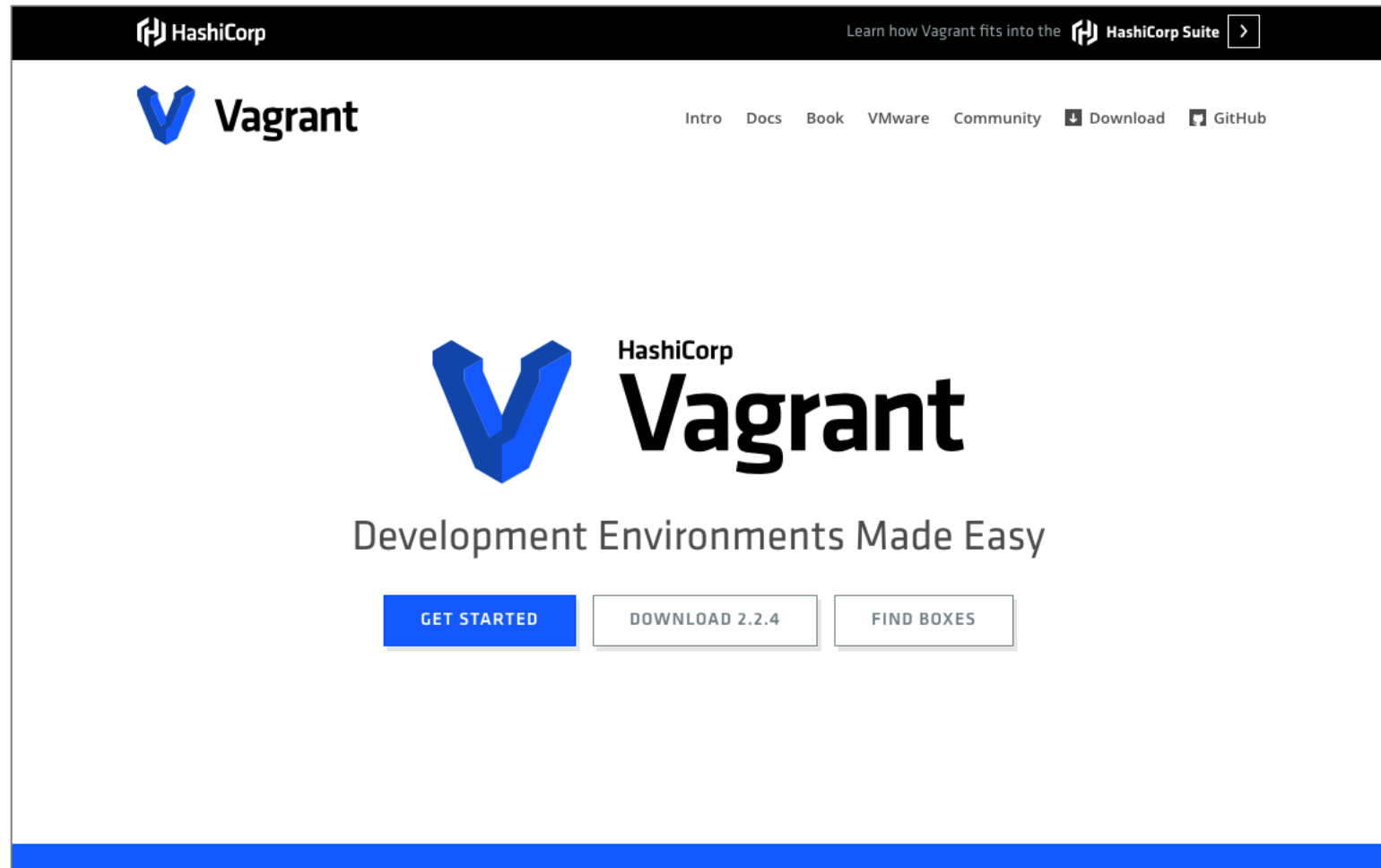
### News Flash

- **New January 28th, 2019 VirtualBox 6.0.4 released!**  
Oracle today released a 6.0 maintenance release which improves stability and fixes regressions. See the [Changelog](#) for details.
- **New January 28th, 2019 VirtualBox 5.2.26 released!**  
Oracle today released a 5.2 maintenance release which improves stability and fixes regressions. See the [Changelog](#) for details.
- **New December 18th, 2018 VirtualBox 6.0 released!**  
Oracle today shipped a new major release, VirtualBox 6.0. See the [Changelog](#) for details.

[More information...](#)

# Step1. VirtualBox와 Vagrant의 인스톨

- Vagrant 인스톨
  - <https://www.virtualbox.org>



# Step2. Ubuntu Linux 게스트 OS 설치와 초기설정

- Vagrant box
  - Guest OS 이미지 (\*.box)
- Ubuntu vagrant box 파일 다운로드와 인스톨
  - > **vagrant box add** ubuntu1604 <https://cloud-images.ubuntu.com/xenial/current/xenial-server-cloudimg-amd64-vagrant.box>

```
1. vagrant box add ubuntu1604 (curl)
dowon@DOWON-MacBook ~/Desktop/강의/hlf$ vagrant box add ubuntu1604 https://cloud-images.ubuntu.com/xenial/current/xenial-server-cloudimg-amd64-vagrant.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'ubuntu1604' (v0) for provider:
box: Downloading: https://cloud-images.ubuntu.com/xenial/current/xenial-server-cloudimg-amd64-vagrant.box
box: Progress: 1% (Rate: 445k/s, Estimated time remaining: 0:16:55)
```

```
1. dowon@DOWON-MacBook: ~/Desktop/강의/hlf (zsh)
dowon@DOWON-MacBook ~/Desktop/강의/hlf$ vagrant box list
ubuntu1604 (virtualbox, 0)
dowon@DOWON-MacBook ~/Desktop/강의/hlf$
```

# Step2. Ubuntu Linux 게스트 OS 설치와 초기설정

- Ubuntu Guest OS 설치
  - > ***vagrant init*** ubuntu1604

```
1. downon@DOWON-MacBook: ~/Desktop/강의/hlf (zsh)
downon@DOWON-MacBook > ~/Desktop/강의/hlf > vagrant init ubuntu1604
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
downon@DOWON-MacBook > ~/Desktop/강의/hlf > ll
total 8
-rw-r--r--  1 downon  staff   2.9K  4  9 17:32 Vagrantfile
downon@DOWON-MacBook > ~/Desktop/강의/hlf > █
```

## Step2. Ubuntu Linux 게스트 OS 설치와 초기설정

- Guest OS 메모리 설정 → Vagrantfile 수정

***vb.memory = "4096"***

```
48 # Provider-specific configuration so you can fine-tune various
49 # backing providers for Vagrant. These expose provider-specific options.
50 # Example for VirtualBox:
51 #
52 config.vm.provider "virtualbox" do |vb|
53 #   # Display the VirtualBox GUI when booting the machine
54 #   vb.gui = true
55 #
56 #   # Customize the amount of memory on the VM:
57   vb.memory = "4096"
58 end
59 #
```



# Step2. Ubuntu Linux 게스트 OS 설치와 초기설정

- 가상머신과 Guest OS 기동

> *vagrant up*

```
1. downon@DOWON-MacBook: ~/Desktop/강의/hlf (zsh)
downon@DOWON-MacBook ~/Desktop/강의/hlf$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu1604'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: hlf_default_1554799220452_79111
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
```

```
    default:
    default: Guest Additions Version: 5.1.38
    default: VirtualBox Version: 5.2
==> default: Mounting shared folders...
    default: /vagrant => /Users/down/Desktop
downon@DOWON-MacBook ~/Desktop/강의/hlf$
```

# Step2. Ubuntu Linux 게스트 OS 설치와 초기설정

- Guest OS 로그인

> ***vagrant ssh***

```
1. vagrant@ubuntu-xenial: ~ (ssh)
down@DOWON-MacBook ~/Desktop/강의/hlf$ vagrant ssh
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@ubuntu-xenial:~$
```

- 유저 비밀번호 변경

\$ ***sudo passwd vagrant***

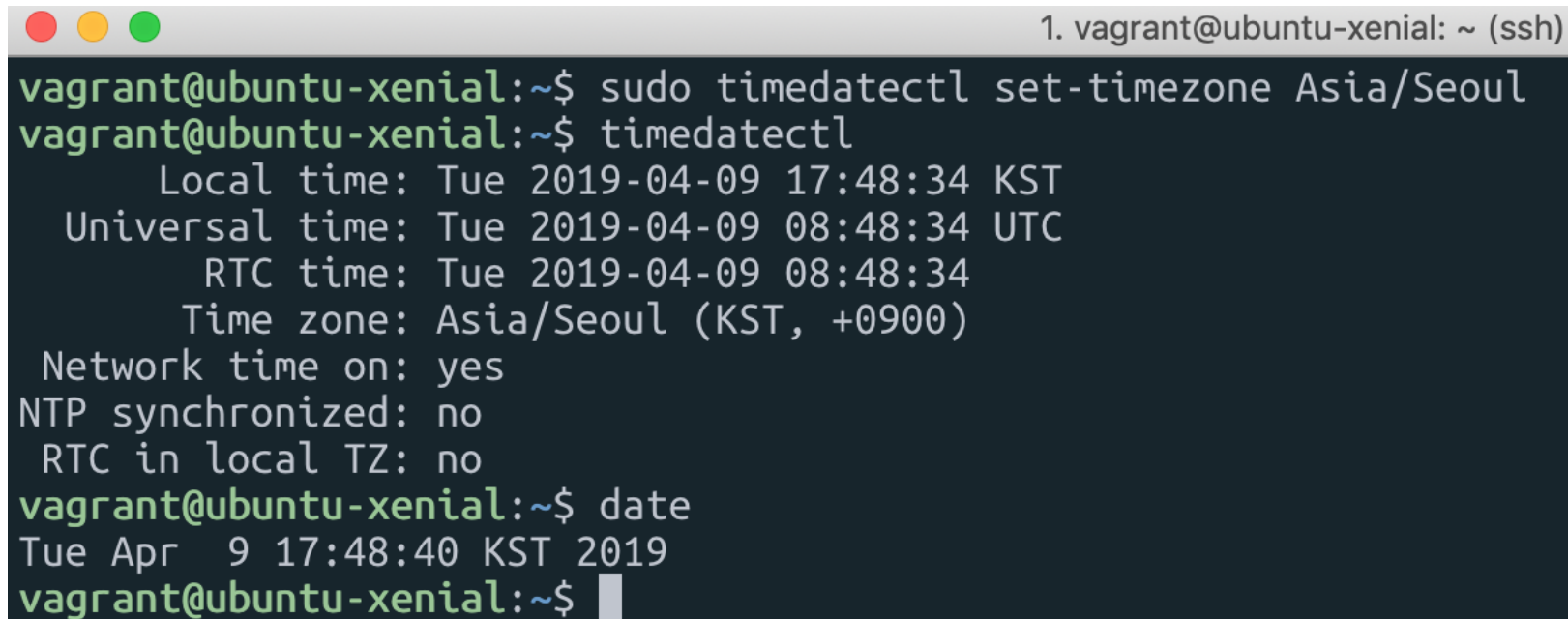
```
1. vagrant@ubuntu-xenial: ~ (ssh)
vagrant@ubuntu-xenial:~$ sudo passwd vagrant
Enter new UNIX password: vagrant
Retype new UNIX password: vagrant
passwd: password updated successfully
vagrant@ubuntu-xenial:~$
```

## Step2. Ubuntu Linux 게스트 OS 설치와 초기설정

- Timezone 설정

```
$ sudo timedatectl set-timezone Asia/Seoul
```

```
$ timedatectl
```



A terminal window titled "1. vagrant@ubuntu-xenial: ~ (ssh)" displays the following commands and output:

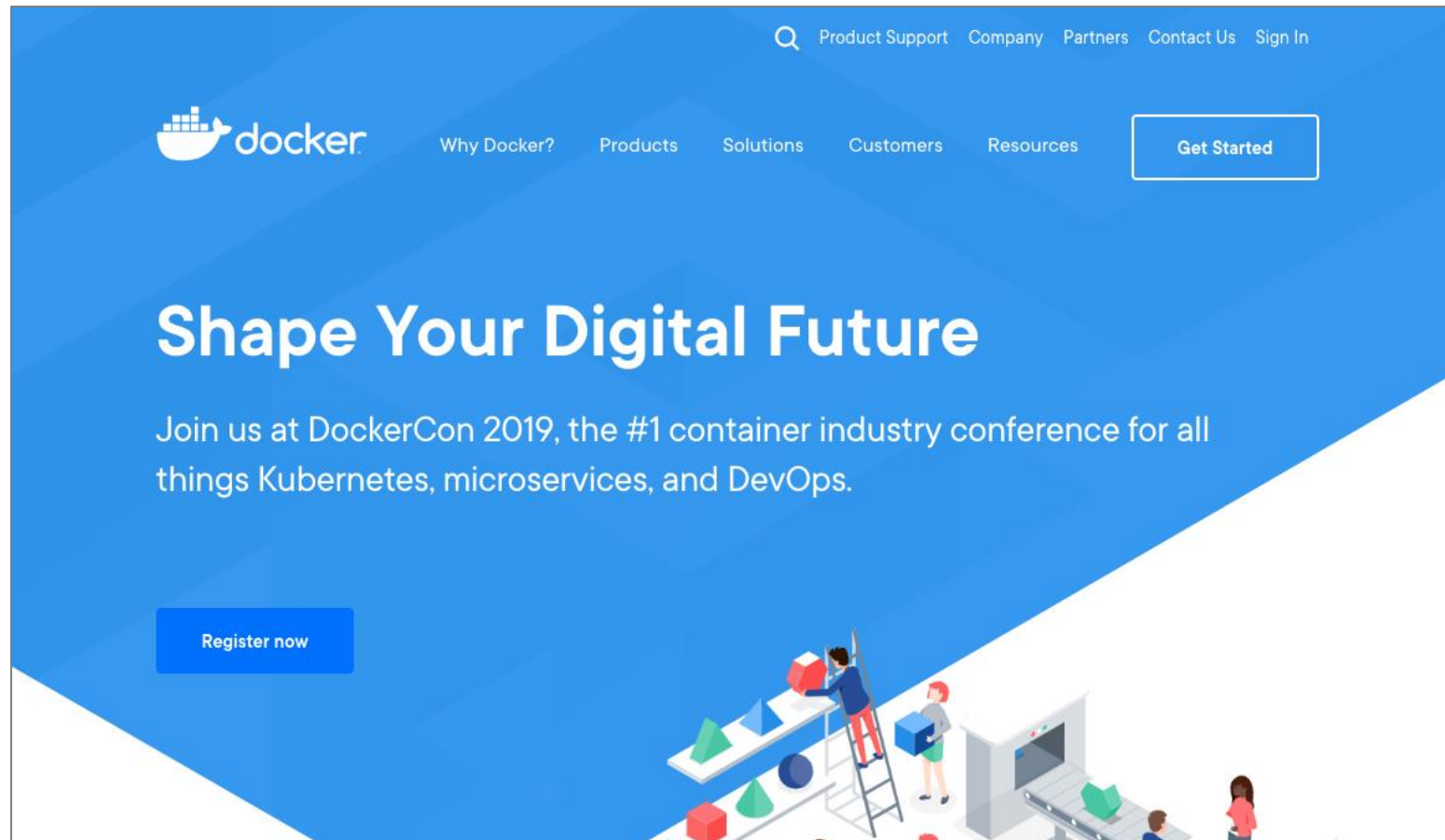
```
vagrant@ubuntu-xenial:~$ sudo timedatectl set-timezone Asia/Seoul
vagrant@ubuntu-xenial:~$ timedatectl
    Local time: Tue 2019-04-09 17:48:34 KST
    Universal time: Tue 2019-04-09 08:48:34 UTC
        RTC time: Tue 2019-04-09 08:48:34
    Time zone: Asia/Seoul (KST, +0900)
Network time on: yes
NTP synchronized: no
RTC in local TZ: no
vagrant@ubuntu-xenial:~$ date
Tue Apr  9 17:48:40 KST 2019
vagrant@ubuntu-xenial:~$
```

## Step2. Ubuntu Linux 게스트 OS 설치와 초기설정

- Guest OS 정지
  - > ***vagrant halt***
- Guest OS 중단, 재개
  - > ***vagrant suspend***
  - > ***vagrant resume***
- Guest OS 제거
  - > ***vagrant destroy***

# Step3. 필수 Tool, SW 설치

- Docker CE (and Docker Compose) 설치 (<https://www.docker.com/>)
  - 컨테이너형 가상화 SW
  - Docker Compose → 컨테이너형 애플리케이션이나 그 네트워크를 구성하는 툴



# Step3. 필수 Tool, SW 설치

- Docker CE (and Docker Compose) 설치 (<https://www.docker.com/>)

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

```
$ sudo apt -y install apt-transport-https ca-certificates curl software-properties-common
```

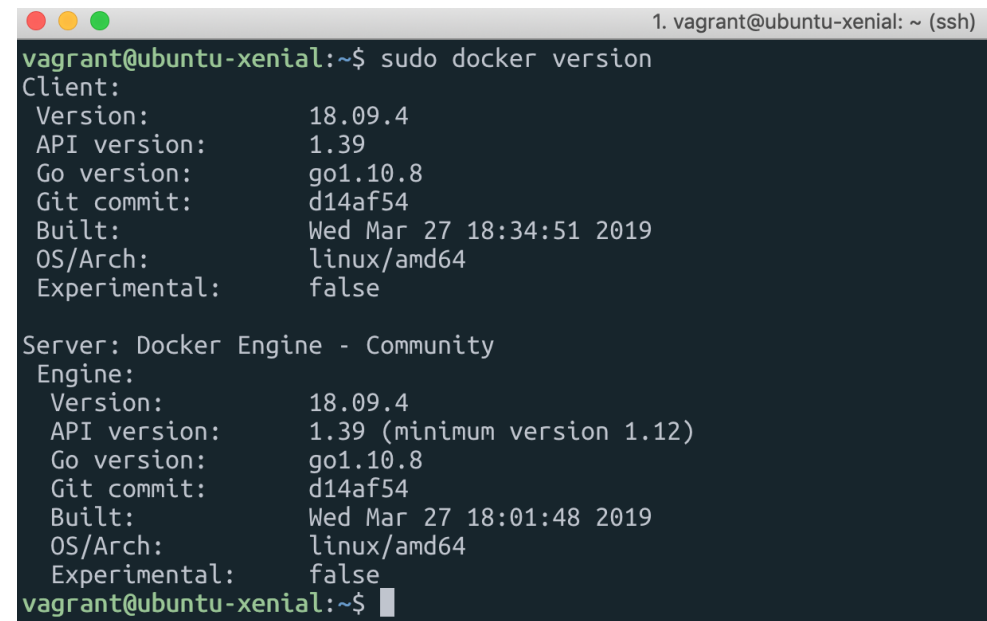
```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
$ sudo apt update
```

```
$ sudo apt -y install docker-ce
```

```
$ sudo docker version
```



```
1. vagrant@ubuntu-xenial: ~ (ssh)
vagrant@ubuntu-xenial:~$ sudo docker version
Client:
Version:      18.09.4
API version:  1.39
Go version:   go1.10.8
Git commit:   d14af54
Built:        Wed Mar 27 18:34:51 2019
OS/Arch:      linux/amd64
Experimental: false

Server: Docker Engine - Community
Engine:
Version:      18.09.4
API version:  1.39 (minimum version 1.12)
Go version:   go1.10.8
Git commit:   d14af54
Built:        Wed Mar 27 18:01:48 2019
OS/Arch:      linux/amd64
Experimental: false
vagrant@ubuntu-xenial:~$
```

# Step3. 필수 Tool, SW 설치

- docker group에 유저 추가

```
$ docker images
```

```
$ sudo usermod -aG docker vagrant
```

```
$ exit
```

```
1. downon@DOWON-MacBook: ~/Desktop/강의/hlf (zsh)
vagrant@ubuntu-xenial:~$ docker images
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get
http://%2Fvar%2Frun%2Fdocker.sock/v1.39/images/json: dial unix /var/run/docker.sock: connect: permission den
ied
vagrant@ubuntu-xenial:~$ sudo usermod -aG docker vagrant
vagrant@ubuntu-xenial:~$ exit
logout
Connection to 127.0.0.1 closed.
downon@DOWON-MacBook > ~/Desktop/강의/hlf
```

- 로그인

```
> vagrant ssh
```

```
$ docker images
```

```
*** System restart required ***
Last login: Tue Apr  9 17:43:58 2019 from 10.0.2.2
vagrant@ubuntu-xenial:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
vagrant@ubuntu-xenial:~$
```

# Step3. 필수 Tool, SW 설치

- Docker Composer 설치

```
$ sudo apt -y install docker-compose
```

```
vagrant@ubuntu-xenial:~$ sudo apt -y install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

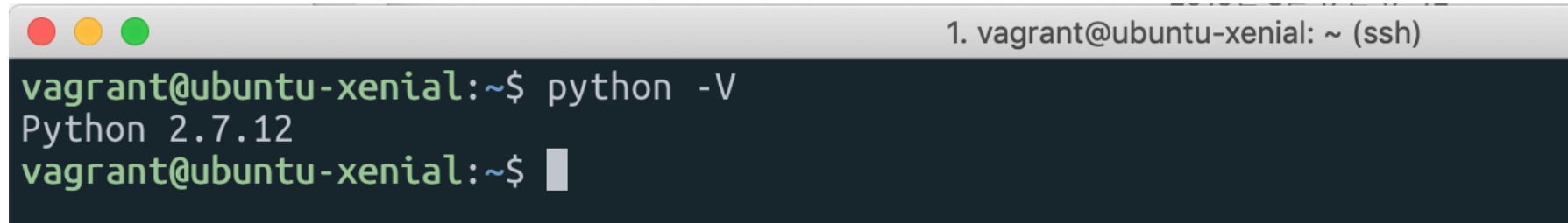
```
Setting up python-texttable (0.8.1-1) ...
Setting up python-yaml (3.11-3build1) ...
Setting up docker-compose (1.8.0-2~16.04.1) ...
Setting up python-cffi-backend (1.5.2-1ubuntu1) ...
Setting up python-idna (2.0-3) ...
Setting up python-pyasn1 (0.1.9-1) ...
Setting up python-cryptography (1.2.3-1ubuntu0.2) ...
Setting up python-openssl (0.15.1-2ubuntu0.2) ...
Setting up python-ndg-httpsclient (0.4.0-3) ...
vagrant@ubuntu-xenial:~$
```

```
1. vagrant@ubuntu-xenial: ~ (ssh)
vagrant@ubuntu-xenial:~$ docker-compose version
docker-compose version 1.8.0, build unknown
docker-py version: 1.9.0
CPython version: 2.7.12
OpenSSL version: OpenSSL 1.0.2g  1 Mar 2016
vagrant@ubuntu-xenial:~$
```



# Step3. 필수 Tool, SW 설치

- Python 2.7 설치
  - docker-compose 설치 시 같이 설치 되어 있음

A terminal window with a title bar showing '1. vagrant@ubuntu-xenial: ~ (ssh)'. The terminal content shows the command 'python -V' being executed, resulting in the output 'Python 2.7.12'. The prompt 'vagrant@ubuntu-xenial:~\$' is visible on both lines.

```
vagrant@ubuntu-xenial:~$ python -V
Python 2.7.12
vagrant@ubuntu-xenial:~$
```

(설치 되어있지 않을 경우)

\$ sudo apt -y install ***python***

# Step3. 필수 Tool, SW 설치

- Golang 설치

```
$ wget https://dl.google.com/go/go1.10.linux-amd64.tar.gz
```

```
$ sudo tar -C /usr/local -xvf go1.10.linux-amd64.tar.gz
```

```
$ echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.profile
```

```
$ . ~/.profile
```

```
$ go version
```

```
vagrant@ubuntu-xenial:~$ echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.profile  
vagrant@ubuntu-xenial:~$ . ~/.profile  
vagrant@ubuntu-xenial:~$ go version  
go version go1.10 linux/amd64  
vagrant@ubuntu-xenial:~$
```

# Step3. 필수 Tool, SW 설치

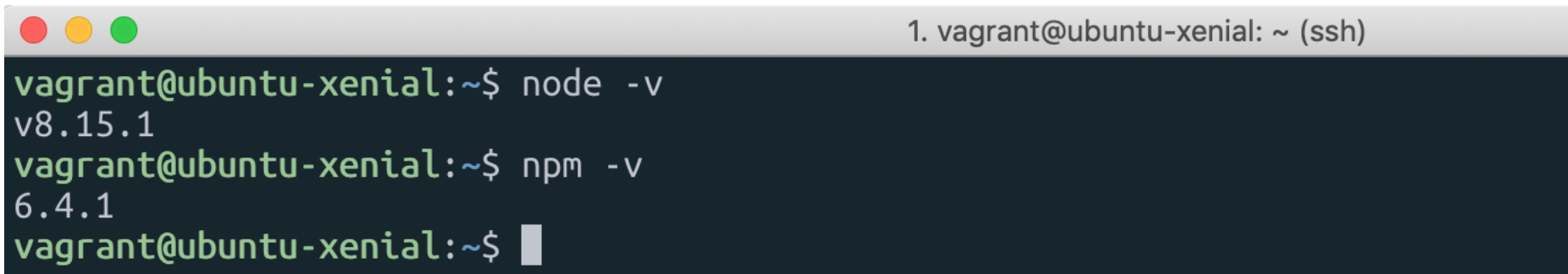
- Node.js, npm 설치

```
$ curl -sL https://deb.nodesource.com/setup 8.x | sudo bash -
```

```
$ sudo apt install nodejs
```

```
$ node -v
```

```
$ npm -v
```

A terminal window with a title bar showing three colored circles (red, yellow, green) and the text "1. vagrant@ubuntu-xenial: ~ (ssh)". The terminal content shows the following commands and outputs:

```
vagrant@ubuntu-xenial:~$ node -v
v8.15.1
vagrant@ubuntu-xenial:~$ npm -v
6.4.1
vagrant@ubuntu-xenial:~$
```

# Step3. 필수 Tool, SW 설치

- GNU make, gcc/g++, libtool 설치
  - npm 커맨드로 native module 빌드하기 위해 필요

\$ sudo ***apt -y install make gcc g++ libtool***

```
vagrant@ubuntu-xenial:~$ sudo apt -y install make gcc g++ libtool
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libuv1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  autotools-dev binutils cpp cpp-5 g++-5 gcc-5 libasan2 libatomic1 libc-dev-bin libc6-dev libcc1-0
  libcilkrts5 libgcc-5-dev libgomp1 libisl15 libitm1 liblsan0 libltdl-dev libmpc3 libmpx0 libquadmath0
  libstdc++-5-dev libtsan0 libubsan0 linux-libc-dev manpages-dev
```

```
Setting up g++ (4:5.3.1-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up libltdl-dev:amd64 (2.4.6-0.1) ...
Setting up libtool (2.4.6-0.1) ...
Setting up make (4.1-6) ...
Setting up manpages-dev (4.04-2) ...
Processing triggers for libc-bin (2.23-0ubuntu11) ...
vagrant@ubuntu-xenial:~$
```

# Step4. Hyperledger Fabric 설치, 동작확인

- CLI, Docker 컨테이너 설치

```
$ mkdir fabric
```

```
$ cd fabric
```

```
$ curl -sSL http://bit.ly/2ysbOFE | bash -s 1.3.0
```

```
vagrant@ubuntu-xenial:~/fabric$ curl -sSL http://bit.ly/2ysbOFE | bash -s 1.3.0
```

```
Installing hyperledger/fabric-samples repo
```

```
==> Cloning hyperledger/fabric-samples repo and checkout v1.3.0
```

```
Cloning into 'fabric-samples'...
```

```
remote: Enumerating objects: 2874, done.
```

```
remote: Total 2874 (delta 0), reused 0 (delta 0), pack-reused 2874
```

```
Receiving objects: 100% (2874/2874), 1004.81 KiB | 192.00 KiB/s, done.
```

```
Resolving deltas: 100% (1423/1423), done.
```

```
Checking connectivity... done.
```

```
Note: checking out 'v1.3.0'.
```

```
Installing Hyperledger Fabric binaries
```

```
==> Downloading version 1.3.0 platform specific fabric binaries
```

```
==> Downloading: https://nexus.hyperledger.org/content/repositories/releases/org/hyperledger/fabric/hyperledger-fabric/linux-amd64-1.3.0/hyperledger-fabric-linux-amd64-1.3.0.tar.gz
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
3 43.3M	3 1686k	0 0	159k 0	0:04:38	0:00:10	0:04:28	202k

# Step4. Hyperledger Fabric 설치, 동작 확인

- PATH에 Hyperledger Fabric Command 추가

```
$ cd fabric-samples
```

```
$ ls bin
```

```
$ echo 'export PATH=$PATH:$HOME/fabric/fabric-samples/bin' >> ~/.profile
```

```
$ . ~/.profile
```

- 설치된 Docker 컨테이너 확인

```
$ docker images
```

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hyperledger/fabric-zookeeper	0.4.15	20c6045930c8	3 weeks ago	1.43GB
hyperledger/fabric-zookeeper	latest	20c6045930c8	3 weeks ago	1.43GB
hyperledger/fabric-kafka	0.4.15	b4ab82bbaf2f	3 weeks ago	1.44GB
hyperledger/fabric-kafka	latest	b4ab82bbaf2f	3 weeks ago	1.44GB
hyperledger/fabric-couchdb	0.4.15	8de128a55539	3 weeks ago	1.5GB
hyperledger/fabric-couchdb	latest	8de128a55539	3 weeks ago	1.5GB
hyperledger/fabric-ca	1.4.0	1a804ab74f58	2 months ago	244MB
hyperledger/fabric-ca	latest	1a804ab74f58	2 months ago	244MB
hyperledger/fabric-javaenv	1.3.0	2476cefaf833	6 months ago	1.7GB
hyperledger/fabric-javaenv	latest	2476cefaf833	6 months ago	1.7GB
hyperledger/fabric-tools	1.3.0	c056cd9890e7	6 months ago	1.5GB
hyperledger/fabric-tools	latest	c056cd9890e7	6 months ago	1.5GB
hyperledger/fabric-ccenv	1.3.0	953124d80237	6 months ago	1.38GB
hyperledger/fabric-ccenv	latest	953124d80237	6 months ago	1.38GB
hyperledger/fabric-orderer	1.3.0	f430f581b46b	6 months ago	145MB
hyperledger/fabric-orderer	latest	f430f581b46b	6 months ago	145MB
hyperledger/fabric-peer	1.3.0	f3ea63abddaa	6 months ago	151MB
hyperledger/fabric-peer	latest	f3ea63abddaa	6 months ago	151MB

# Lab1. fabcar 실행

- Hyperledger Fabric 동작 확인
- Hyperledger Fabric SDK for Node.js를 개발된 자동차 거래 애플리케이션
- npm tool로 의존성 모듈 설치 → 설치 스크립트 실행

```
$ cd fabcar
```

```
$ npm install → Fabric SDK 설치 (fabric-ca-client, fabric-client)
```

```
$ ./startFabric.sh
```

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ ./startFabric.sh
```

```
# don't rewrite paths for Windows Git Bash users
export MSYS_NO_PATHCONV=1
```

```
docker-compose -f docker-compose.yml down
Removing network net_basic
WARNING: Network net_basic not found.
```

```
docker-compose -f docker-compose.yml up -d ca.example.com orderer.example.com peer0.org1.example.com couchdb
Creating network "net_basic" with the default driver
Creating orderer.example.com
Creating ca.example.com
Creating couchdb
Creating peer0.org1.example.com
```

# Lab1. fabcar 실행

- Organization X 1
- Certificate Authority X 1
- Orderer X 1
- Peer X 1
- CouchDB X1

```
1. vagrant@ubuntu-xenial: ~/fabric/fabric-samples/fabcar (ssh)
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ docker ps --format "table {{.Names}}\t{{.Command}}" --no-trunc
NAMES                                COMMAND
dev-peer0.org1.example.com-fabcar-1.0 "chaincode -peer.address=peer0.org1.example.com:7052"
cli                                   "/bin/bash"
peer0.org1.example.com                "peer node start"
couchdb                               "tini -- /docker-entrypoint.sh /opt/couchdb/bin/couchdb"
ca.example.com                        "sh -c 'fabric-ca-server start -b admin:adminpw'"
orderer.example.com                   "orderer"
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$
```



```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$
```

- 관리 유저 admin 등록  
\$ ***node enrollAdmin.js***
- 일반 유저 user1 등록  
\$ ***node registerUser.js***

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ node enrollAdmin.js
Store path:/home/vagrant/fabric/fabric-samples/fabcar/hfc-key-store
(node:22345) DeprecationWarning: grpc.load: Use the @grpc/proto-loader module with grpc.loadPackageDefinition instead
Successfully enrolled admin user "admin"
Assigned the admin user to the fabric client ::{"name":"admin","mspid":"Org1MSP","roles":null,"affiliation":"","enrollmentSecret":"","enrollment":{"signingIdentity":"69a7e03273ad3ef35576fce302bb049b66d1ed46f80cf47fa8c4afe4e23bb73c","identity":{"certificate":"-----BEGIN CERTIFICATE-----\nMIICAJCCAaigAwIBAgIUdT31fSqXADndpK0pRdf3d3jKRMkwCgYIKoZIZj0EAWIw\nczE  
LMAKGA1UEBhMCVVMxEzARBgNVBAgTCkNhbgGlb2wJuaWExFjAUBgNVBARTDVNh\nbiBGcmFuY2ZyZ28xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAa  
BgNVBAMT\nE2NhLm9yZzEuZXhhbXBsZS5jb20wHhcNMjkwNDAsMTQxMjAwHhcnNzAwWjAhMQ8wDQYDVQLEWwjbGllbnQxZjAMBgNVBAM  
TBWFKbWlufMFkwEwYHKoZI\nzj0CAQYIKoZIZj0DAQDQgAEyNhoPffbt+P4BuHiPGoIL9gTrpWk6HqdAdAjpDrs\n\nCCJ6LsyBGqffe0S27gf2R9ZsyH1mW2  
oWShnoz7Hket6T9aNsMgowDgYDVR0PAQH\n\nBAQDAgeAMAwGA1UdEwEB\n\nwQCMAAwHQYDVRO0BBYEFN6o2r0uYGEJVau1APNBhzy\n\ni3lsMCsGA1UdIwQkM  
CKAIEI5qg3NdtruuLoM2nAYUdFFBNMarRst3dusalC2Xkl8\n\nMAoGCCqGSM49BAMCA0gAMEUCIQCAIwW20T7dsLhT98GBNc4iIM0eJtCUDU6lKgkK\n\n9Lmn  
xAIgWuOkdeq2HQtwV\n\nl3NvVr5FFfsXbSbgNLORVU9KptvUQ=\n\n-----END CERTIFICATE-----\n\n"}}}
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ node registerUser.js
Store path:/home/vagrant/fabric/fabric-samples/fabcar/hfc-key-store
(node:22361) DeprecationWarning: grpc.load: Use the @grpc/proto-loader module with grpc.loadPackageDefinition instead
Successfully loaded admin from persistence
Successfully registered user1 - secret:jGnSTgBbNqSv
Successfully enrolled member user "user1"
User1 was successfully registered and enrolled and is ready to interact with the fabric network
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$
```

# Lab1. fabcar 실행

- 유저 user1로 모든 데이터를 조회 → 결과 출력

\$ *node query.js*

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ node query.js
Store path:/home/vagrant/fabric/fabric-samples/fabcar/hfc-key-store
(node:22379) DeprecationWarning: grpc.load: Use the @grpc/proto-loader module with grpc.loadPackageDefinition instead
Successfully loaded user1 from persistence
Query has completed, checking results
Response is [{"Key":"CAR0", "Record":{"colour":"blue","make":"Toyota","model":"Prius","owner":"Tomoko"}},{ "Key":"CAR1",
, "Record":{"colour":"red","make":"Ford","model":"Mustang","owner":"Brad"}},{ "Key":"CAR2", "Record":{"colour":"green",
make":"Hyundai","model":"Tucson","owner":"Jin Soo"}},{ "Key":"CAR3", "Record":{"colour":"yellow","make":"Volkswagen","mo
del":"Passat","owner":"Max"}},{ "Key":"CAR4", "Record":{"colour":"black","make":"Tesla","model":"S","owner":"Adriana"}},
{"Key":"CAR5", "Record":{"colour":"purple","make":"Peugeot","model":"205","owner":"Michel"}},{ "Key":"CAR6", "Record":{"
colour":"white","make":"Chery","model":"S22L","owner":"Aarav"}},{ "Key":"CAR7", "Record":{"colour":"violet","make":"Fiat
","model":"Punto","owner":"Pari"}},{ "Key":"CAR8", "Record":{"colour":"indigo","make":"Tata","model":"Nano","owner":"Val
eria"}},{ "Key":"CAR9", "Record":{"colour":"brown","make":"Holden","model":"Barina","owner":"Shotaro"}}]
```

# Lab1. fabcar 실행

- 새로운 자동차 등록

\$ *vi invoke.js* → 새로운 자동차 정보 입력

```
58      // createCar chaincode function - requires 5 args, ex: args: ['CAR12', 'Honda', 'Accord', 'Black', 'Tom'],
59      // changeCarOwner chaincode function - requires 2 args , ex: args: ['CAR10', 'Dave'],
60      // must send the proposal to endorsing peers
61      var request = {
62          //targets: let default to the peer assigned to the client
63          chaincodeId: 'fabcar',
64          fcn: 'createCar',
65          args: ['CAR12', 'Honda', 'Accord', 'Black', 'Tom'],
66          chainId: 'mychannel',
67          txId: tx_id
68      };
```

\$ *node invoke.js*

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ node invoke.js
Store path:/home/vagrant/fabric/fabric-samples/fabcar/hfc-key-store
(node:22411) DeprecationWarning: grpc.load: Use the @grpc/proto-loader module with grpc.loadPackageDefinition instead
Successfully loaded user1 from persistence
Assigning transaction_id: 010cc080b7abf26880b47877d283b3e4612bc25ba07fb34240abdff4b4fb7abe
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message - ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$
```

# Lab1. fabcar 실행

- 새로운 추가된 자동차 검색

\$ *vi query.js* → 새로운 추가된 자동차 정보 입력

```
52      // queryCar chaincode function - requires 1 argument, ex: args: ['CAR4'],
53      // queryAllCars chaincode function - requires no arguments , ex: args: [''],
54      const request = {
55          //targets : --- letting this default to the peers assigned to the channel
56          chaincodeId: 'fabcar',
57          fcn: 'queryCar',
58          args: ['CAR12']
59      };
60
```

\$ *node query.js*

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ node query.js
Store path:/home/vagrant/fabric/fabric-samples/fabcar/hfc-key-store
(node:22436) DeprecationWarning: grpc.load: Use the @grpc/proto-loader module with grpc.loadPackageDefinition instead
Successfully loaded user1 from persistence
Query has completed, checking results
Response is  {"colour":"Black","make":"Honda","model":"Accord","owner":"Tom"}
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$
```

# Lab1. fabcar 실행

## Q) CAR12의 소유권 변경

- CAR12의 소유권을 "Tom" → "Barry"로 변경

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ node invoke.js
Store path:/home/vagrant/fabric/fabric-samples/fabcar/hfc-key-store
(node:22475) DeprecationWarning: grpc.load: Use the @grpc/proto-loader module with grpc.loadPackageDefinition instead
Successfully loaded user1 from persistence
Assigning transaction_id: c3057f596bd000c998a2e9eb4cbd5237893a3b6c742ef4e3e2fd5b8ce2523bd3
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message - ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
vagrant@ubuntu-xenial:~/fabric/fabric-samples/fabcar$ node query
Store path:/home/vagrant/fabric/fabric-samples/fabcar/hfc-key-store
(node:22488) DeprecationWarning: grpc.load: Use the @grpc/proto-loader module with grpc.loadPackageDefinition instead
Successfully loaded user1 from persistence
Query has completed, checking results
Response is  {"colour":"Black","make":"Honda","model":"Accord","owner":"Barry"}
```

# Lab1. fabcar 실행

- fabcar 샘플 삭제
  - basic-network에 포함되어 있는 스크립트 사용

```
$ cd ../basic-network
```

```
$ ./stop.sh
```

```
$ docker rm $(docker ps -aq) → 체인코드 컨테이너 삭제
```

```
$ ./teardown.sh
```

```
vagrant@ubuntu-xenial:~/fabric/fabric-samples/basic-network$ ./stop.sh
# Shut down the Docker containers that might be currently running.
docker-compose -f docker-compose.yml stop
Stopping cli ... done
Stopping peer0.org1.example.com ... done
Stopping couchdb ... done
Stopping ca.example.com ... done
Stopping orderer.example.com ... done
vagrant@ubuntu-xenial:~/fabric/fabric-samples/basic-network$ docker rm $(docker ps -qa)
be04be50b251
bfb9000094c2
0d0ed35cea1f
4f472850b6e9
33cb81858685
2f1c686fcb8f
vagrant@ubuntu-xenial:~/fabric/fabric-samples/basic-network$ ./teardown.sh
Removing network net_basic
```