



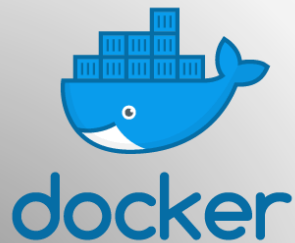
Block Chain

고급 프로젝트 정규교육

부록

하이퍼레저 패브릭 개발환경 설정

Hyperledger DApp 개발 필요구성요소



Back-end

Getting Started

Prerequisites

Install Samples, Binaries and Docker Images

Hyperledger Fabric SDKs

Hyperledger Fabric CA

curl://



Front-end



설치상세

① Curl, docker 설치



② Node.js 설치



③ Go언어 설치



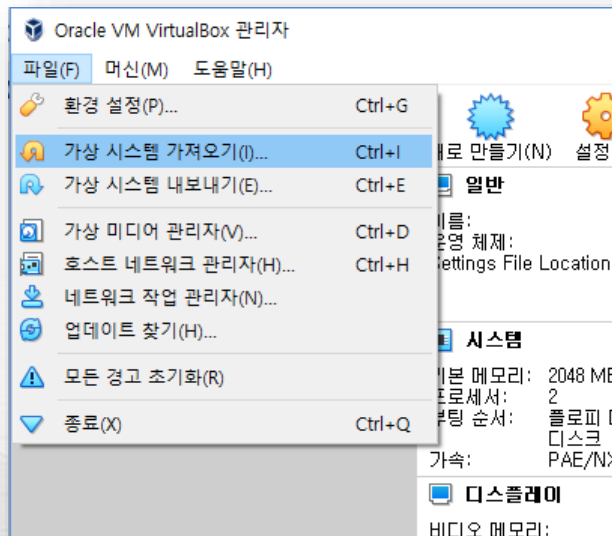
④ GIT설치



⑤ fabric-samples 설치
docker 이미지 설치

Image 가져오기 및 실행

- Virtual box 설치
- 이미지 다운로드
- 가상시스템 가져오기 ubuntu2004_fabric_01.ova
- 가상시스템 시작
- ID: bstudent Passwd:sys1234



가져오기 순서

📁 구글에서 virtual box 검색 후 다운로드 -> 설치

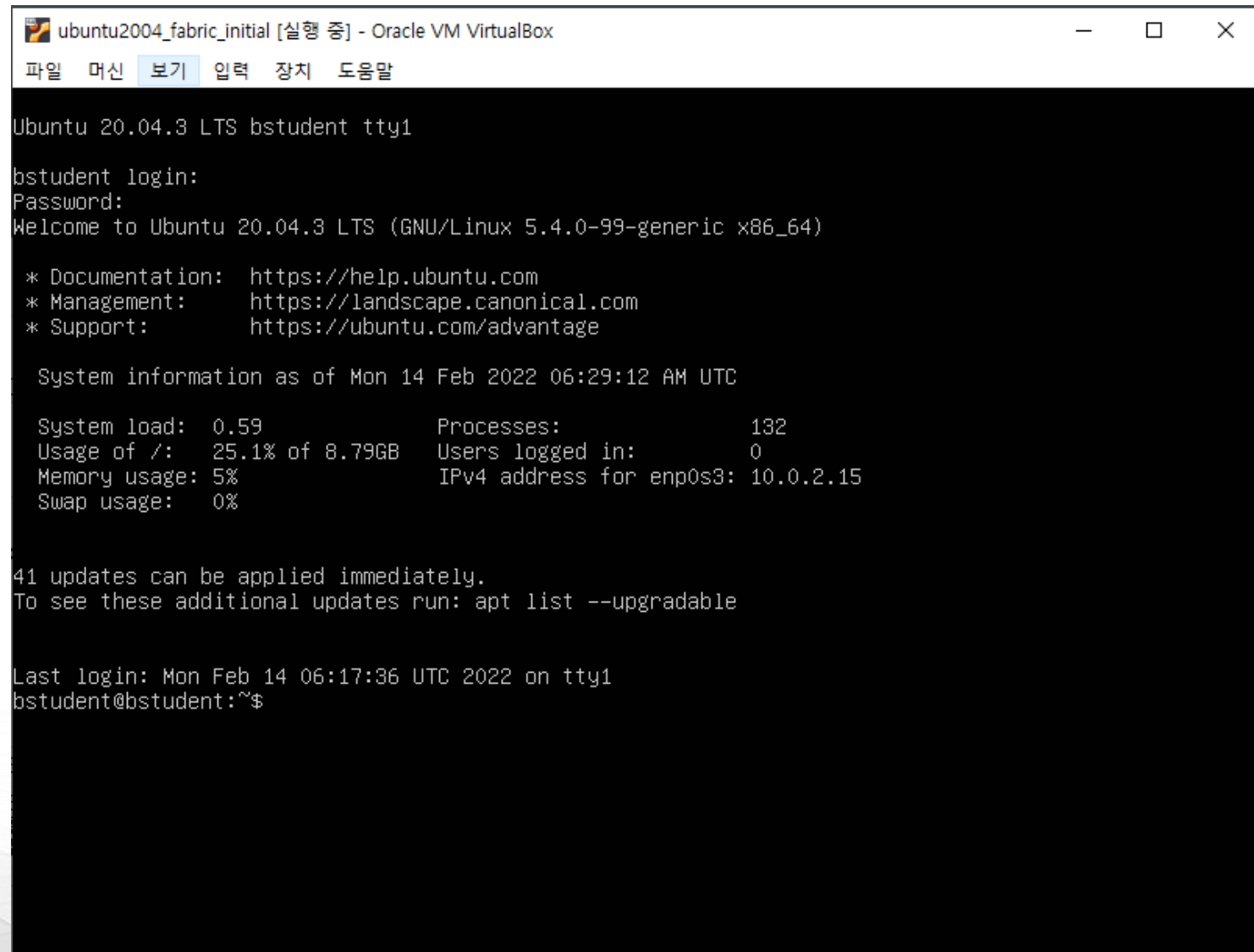
- ▣ <https://www.virtualbox.org/wiki/Downloads>
- ▣ virtual box 설치 이 후 VM VirtualBox Extension Pack 추가 설치

📁 virtual box 실행

- ▣ 메뉴 -> 가상시스템 가져오기
-> 다운로드 받은 ubuntu2004_fabric_01.ova파일 선택
- ▣ 설정 (자신의 컴퓨터 사양에 따라)
 - 시스템 이름 - fabric개발환경
 - vcpu - 2~4개
 - vram - 1048~4096MB
 - USB 체크 해제
- ▣ 가져오기 후 왼쪽 트리에서 fabric개발환경 선택후 녹색 화살표 시작 클릭

*** 주의 사항 : 하드디스크 여유공간 40기가 이상 꼭 확인**

Hyperledger Fabric 설치



```
ubuntu2004_fabric_initial [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

Ubuntu 20.04.3 LTS bstudent tty1

bstudent login:
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-99-generic x86_64)

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

System information as of Mon 14 Feb 2022 06:29:12 AM UTC

System load:  0.59               Processes:            132
Usage of /:   25.1% of 8.79GB    Users logged in:     0
Memory usage: 5%                IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%

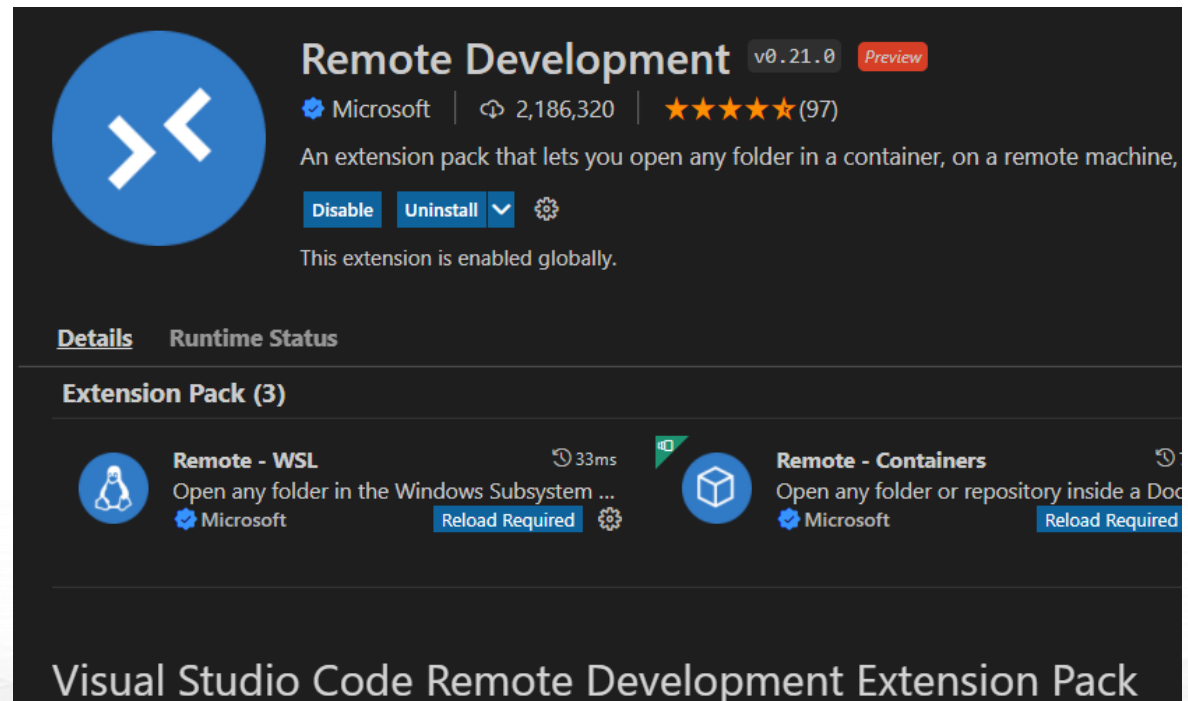
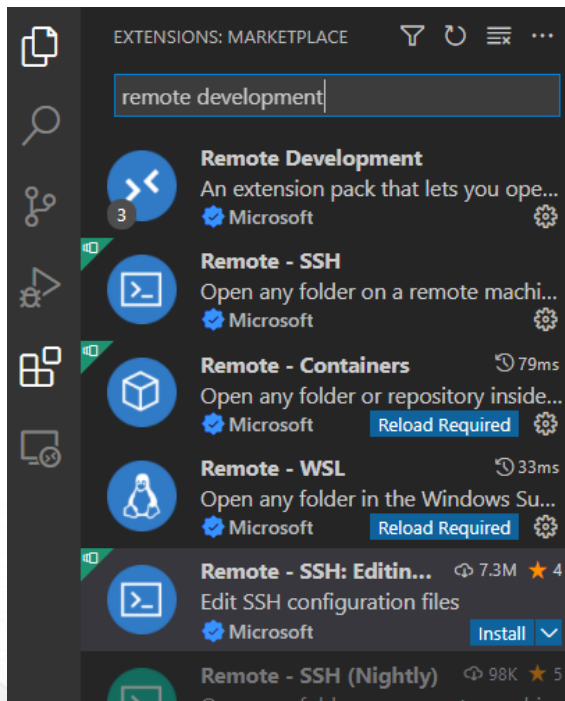
41 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Mon Feb 14 06:17:36 UTC 2022 on tty1
bstudent@bstudent:~$
```


Vscode 원격 연동1

📌 Remote Development Extension 설치

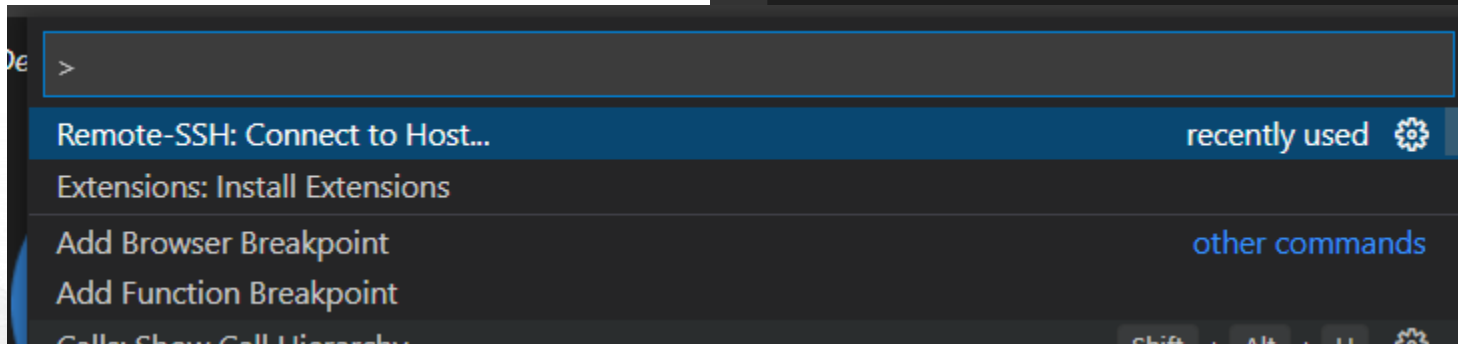
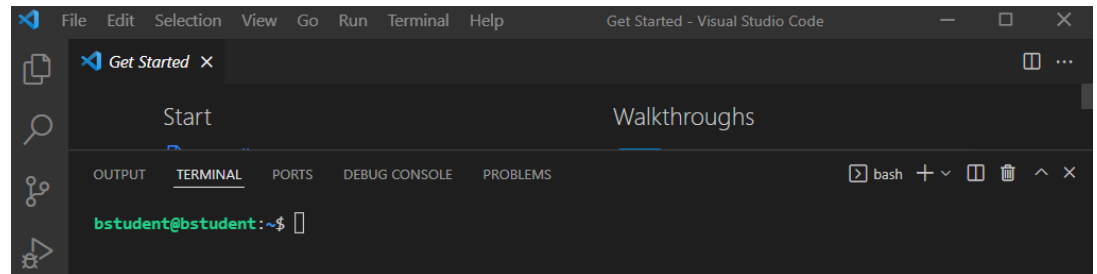
- ❑ vs.code 실행 -> extensions 탭 실행 -> Remote Development 검색
- ❑ 설치 수행



Vscode 원격 연동2

📌 Vscode 원격 연동하기

- ▣ vs.code 실행 -> F1 실행 -> Remote-SSH 실행
- ▣ bstudent@localhost 입력
- ▣ 새창에서 view -> 터미널 (원격 접속된 터미널 창 사용가능)



cURL 설치

📦 cURL 설치 명령어

- ❑ `sudo apt-get install curl`

```
selecting previously unselected package curl.  
Preparing to unpack .../curl_7.58.0-2ubuntu3.16_amd64.deb ...  
Unpacking curl (7.58.0-2ubuntu3.16) ...  
libcurl4:amd64 (7.58.0-2ubuntu3.16) 설정하는 중입니다 ...  
Processing triggers for libc-bin (2.27-3ubuntu1) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
curl (7.58.0-2ubuntu3.16) 설정하는 중입니다 ...  
bstudent@bstudentN1:~$
```

📦 cURL 설치 확인

- ❑ `curl --version`

```
bstudent@bstudentN1:~$ curl --version  
curl 7.58.0 (x86_64-pc-linux-gnu) libcurl/7.58.0 OpenSSL/1.1.1 zlib/1.2.11 libidn  
(+libidn2/2.0.4) nghttp2/1.30.0 librtmp/2.3  
Release-Date: 2018-01-24  
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s  
tp smtps telnet tftp  
Features: AsynchDNS IDN IPv6 Largefile GSS-API Kerberos SPNEGO NTLM NTLM_WB SSL  
xSockets HTTPS-proxy PSL  
bstudent@bstudentN1:~$
```

docker와 docker compose 설치

📦 docker 설치 명령어

- ❑ `sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg`
- ❑ `echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`
- ❑ `sudo apt-get update`
- ❑ `sudo apt-get install docker-ce=5:20.10.12~3-0~ubuntu-focal docker-ce-cli=5:20.10.12~3-0~ubuntu-focal containerd.io`

docker와 docker compose 설치

docker-compose 설치 명령어

- ❑ `sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
- ❑ `sudo chmod +x /usr/local/bin/docker-compose`

user에게 docker 실행 권한주기 (username == bstudent)

- ❑ `sudo usermod -aG docker <username>`

재부팅하여 설정 적용

- ❑ `sudo shutdown -r now`

Docker 설치 확인

📦 docker 설치 확인 명령어

- ❑ docker version
- ❑ docker server 세팅까지 완료 되었는지 확인

📦 docker-compose 설치 확인

```
bstudent@bstudentN1:~$ docker-compose version
docker-compose version 1.17.1, build unknown
docker-py version: 2.5.1
CPython version: 2.7.17
OpenSSL version: OpenSSL 1.1.1 11 Sep 2018
bstudent@bstudentN1:~$
```

```
Client:
Version:      20.10.7
API version:  1.41
Go version:   go1.13.8
Git commit:   20.10.7-0ubuntu5~18.04.3
Built:        Mon Nov 1 01:04:14 2021
OS/Arch:      linux/amd64
Context:      default
Experimental: true

Server:
Engine:
Version:      20.10.7
API version:  1.41 (minimum version 1.12)
Go version:   go1.13.8
Git commit:   20.10.7-0ubuntu5~18.04.3
Built:        Fri Oct 22 00:57:37 2021
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.5.5-0ubuntu3~18.04.1
GitCommit:
runc:
Version:      1.0.1-0ubuntu2~18.04.1
GitCommit:
docker-init:
Version:      0.19.0
GitCommit:
bstudent@bstudentN1:~$
```

node.js 설치

📦 nodejs 설치 명령어

- ❑ `curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash`
- ❑ `sudo apt-get install -y nodejs`
- ❑ `sudo apt-get install -y build-essential`

📦 node.js 설치 확인 명령어

- ❑ `node -v`
- ❑ `npm -v`

```
bstudent@bstudentN1:~$ node -v
v16.13.1
bstudent@bstudentN1:~$ npm -v
8.1.2
bstudent@bstudentN1:~$
```

go 설치

go 소스코드 직접 설치

- ❑ curl -LO <https://golang.org/dl/go1.16.13.linux-amd64.tar.gz>

```
bstudent@bstudentN1:~$ curl -LO https://golang.org/dl/go1.17.linux-amd64.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100    78    100    78      0      0    197      0 --:--:-- --:--:-- --:--:--   197
100    73    100    73      0      0     80      0 --:--:-- --:--:-- --:--:--     0
100 128M    100 128M      0      0 23.5M      0 0:00:05 0:00:05 --:--:-- 33.6M
bstudent@bstudentN1:~$
```

- ❑ sudo tar -C /usr/local -xzf go1.16.13.linux-amd64.tar.gz
- ❑ rm go1.16.13.linux-amd64.tar.gz

go 설치경로를 PATH 환경변수에 추가

- ❑ code ~/.profile
- ❑ .profile 마지막에 “PATH=\$PATH:/usr/local/go/bin” 추가 후 저장
- ❑ source ~/.profile

go 실행을 위한 .profile

```
Get Started X $ .profile X
home > bstudent > $ .profile
11 # if running bash
12 if [ -n "$BASH_VERSION" ]; then
13     # include .bashrc if it exists
14     if [ -f "$HOME/.bashrc" ]; then
15         . "$HOME/.bashrc"
16     fi
17 fi
18
19 # set PATH so it includes user's private bin if it exists
20 if [ -d "$HOME/bin" ] ; then
21     PATH="$HOME/bin:$PATH"
22 fi
23
24 # set PATH so it includes user's private bin if it exists
25 if [ -d "$HOME/.local/bin" ] ; then
26     PATH="$HOME/.local/bin:$PATH"
27 fi
28
29 PATH=$PATH:/usr/local/go/bin
```

추가 후 저장

go 설치 확인

go 설치 확인 명령어

go version

```
bstudent@bstudentN1: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
bstudent@bstudentN1:~$ go version  
go version go1.17.6 linux/amd64  
bstudent@bstudentN1:~$
```

Git 설치

Git 설치 명령어

- ▣ sudo apt-get install git

```
Unpacking git-man (1:2.17.1-1ubuntu0.9) ...  
Selecting previously unselected package git.  
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.9_amd64.deb ...  
Unpacking git (1:2.17.1-1ubuntu0.9) ...  
git-man (1:2.17.1-1ubuntu0.9) 설정하는 중입니다 ...  
liberror-perl (0.17025-1) 설정하는 중입니다 ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
git (1:2.17.1-1ubuntu0.9) 설정하는 중입니다 ...  
bstudent@bstudentN1:~$
```

Git 설치 확인

- ▣ git version

```
bstudent@bstudentN1: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
bstudent@bstudentN1:~$ git version  
git version 2.17.1
```

fabric-samples 설치

📦 Samples, Binaries, Docker Images 설치 명령어

❑ `curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.2.3 1.5.2`


<fabric_version> <fabric-ca_version>

📦 fabric 설치경로를 PATH 환경변수에 추가

📦 (username == bstudent)

- ❑ `code ~/.profile`
- ❑ `.profile` 마지막에 “`PATH=/home/<username>/fabric-samples/bin:$PATH`” 추가 후 저장
- ❑ `source ~/.profile`

fabric 실행을 위한 .profile

```
Get Started  $ .profile x
home > bstudent > $ .profile
11  # if running bash
12  if [ -n "$BASH_VERSION" ]; then
13      # include .bashrc if it exists
14      if [ -f "$HOME/.bashrc" ]; then
15          . "$HOME/.bashrc"
16      fi
17  fi
18
19  # set PATH so it includes user's private bin if it exists
20  if [ -d "$HOME/bin" ] ; then
21      PATH="$HOME/bin:$PATH"
22  fi
23
24  # set PATH so it includes user's private bin if it exists
25  if [ -d "$HOME/.local/bin" ] ; then
26      PATH="$HOME/.local/bin:$PATH"
27  fi
28
29  PATH=$PATH:/usr/local/go/bin
30  PATH=/home/bstudent/fabric-samples/bin:$PATH
```

추가 후 저장

```
bstudent@bstudent:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hyperledger/fabric-ca	1.5	4ea287b75c63	5 months ago	69.8MB
hyperledger/fabric-ca	1.5.2	4ea287b75c63	5 months ago	69.8MB
hyperledger/fabric-ca	latest	4ea287b75c63	5 months ago	69.8MB
hyperledger/fabric-tools	2.2	b6acbfaac0ef	9 months ago	431MB
hyperledger/fabric-tools	2.2.3	b6acbfaac0ef	9 months ago	431MB
hyperledger/fabric-tools	latest	b6acbfaac0ef	9 months ago	431MB
hyperledger/fabric-peer	2.2	17cd44c37a94	9 months ago	52.6MB
hyperledger/fabric-peer	2.2.3	17cd44c37a94	9 months ago	52.6MB
hyperledger/fabric-peer	latest	17cd44c37a94	9 months ago	52.6MB
hyperledger/fabric-orderer	2.2	345ca8c4b4b1	9 months ago	36.5MB
hyperledger/fabric-orderer	2.2.3	345ca8c4b4b1	9 months ago	36.5MB
hyperledger/fabric-orderer	latest	345ca8c4b4b1	9 months ago	36.5MB
hyperledger/fabric-ccenv	2.2	bd77faf79f92	9 months ago	514MB
hyperledger/fabric-ccenv	2.2.3	bd77faf79f92	9 months ago	514MB
hyperledger/fabric-ccenv	latest	bd77faf79f92	9 months ago	514MB
hyperledger/fabric-baseos	2.2	f98af14c474f	9 months ago	6.86MB
hyperledger/fabric-baseos	2.2.3	f98af14c474f	9 months ago	6.86MB
hyperledger/fabric-baseos	latest	f98af14c474f	9 months ago	6.86MB

```
bstudent@bstudent:~$ tree -L 1 fabric-samples/
```

```
fabric-samples/
```

```
├── asset-transfer-abac
├── asset-transfer-basic
├── asset-transfer-events
├── asset-transfer-ledger-queries
├── asset-transfer-private-data
├── asset-transfer-sbe
├── asset-transfer-secured-agreement
├── auction
├── bin
├── chaincode
├── CHANGELOG.md
├── ci
├── CODE_OF_CONDUCT.md
├── CODEOWNERS
├── commercial-paper
├── config
├── CONTRIBUTING.md
├── fabcar
├── high-throughput
├── interest_rate_swaps
├── LICENSE
├── MAINTAINERS.md
├── off_chain_data
├── README.md
├── scripts
├── SECURITY.md
├── test-application
├── test-network
├── token-erc-20
└── token-utxo
```

```
22 directories, 8 files
```

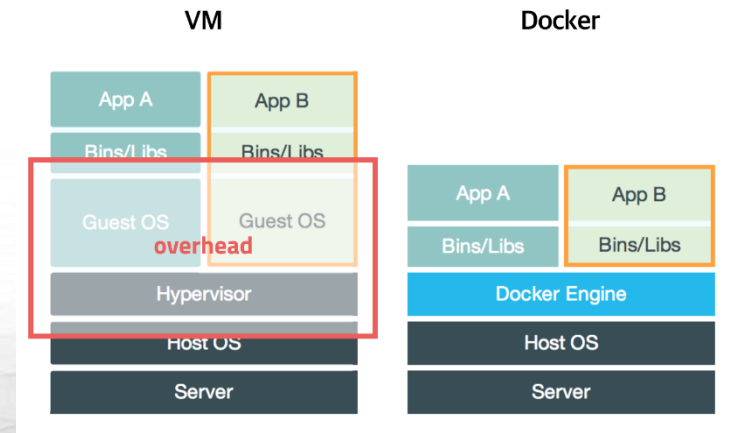
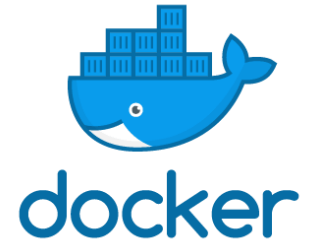
```
bstudent@bstudent:~$
```

Docker 란?

❖ 컨테이너 기반의 오픈소스 가상화 플랫폼

- ❑ 실행 프로그램을 배포하기 위한 프로그램
- ❑ 실행 프로그램+실행 시 필요한 환경
- ❑ Image : 실행을 시키기 위한 프로그램
- ❑ Container : Image를 받아 실행시킨 프로세스

❖ 컨테이너, 오버레이 네트워크, 유니온 파일 시스템등 이미 존재하는 기술을 잘 조합



간단한 Docker사용법



용어

□ 이미지

- 컨테이너 실행에 필요한 파일과 설정값등을 포함하고 있는 것
- 현재 공개된 도커 이미지는 50만개가 넘고 Docker hub의 이미지 다운로드 수는 80억회이상

□ 컨테이너

- 하나의 프로세스 (이미지의 인스턴스)
- 실행중인 프로그램

□ 볼륨

- 컨테이너에 마운트해서 사용할 수 있는 디렉토리

□ docker-compose (docker container실행 자동화 도구)

- 컨테이너 실행에 필요한 옵션을 docker-compose.yml에 작성
- 컨테이너 간 실행 순서나 의존성 관리

설치

- `curl -fsSL https://get.docker.com/ | sudo sh`

간단한 Docker사용법



주요명령

기능	형식, 예
컨테이너 목록 확인하기	<code>docker ps -a</code>
컨테이너 실행하기	<code>docker run [OPTIONS] IMAGE[:TAG @DIGEST] [COMMAND] [ARG...] docker run ubuntu:16.04</code>
컨테이너 중지하기	<code>docker stop [OPTIONS] CONTAINER [CONTAINER...]</code>
컨테이너 제거하기	<code>docker rm [OPTIONS] CONTAINER [CONTAINER...]</code>
이미지 목록 확인하기	<code>docker images [OPTIONS] [REPOSITORY[:TAG]]</code>
이미지 다운로드하기	<code>docker pull [OPTIONS] NAME[:TAG @DIGEST] docker pull ubuntu:14.04</code>
이미지 삭제하기	<code>docker rmi [OPTIONS] IMAGE [IMAGE...]</code>
컨테이너 로그 보기	<code>docker logs [OPTIONS] CONTAINER</code>
컨테이너 명령어 실행하기	<code>docker exec [OPTIONS] CONTAINER COMMAND [ARG...] docker exec -it mysql /bin/bash</code>

Run script를 포함한 virtual box 이미지배포

- Virtual box 이미지
- Ubuntu 20.04
- cURL, node.js, python, go-language, docker, docker-compose, vim, tree ...
- Hyperledger v2.2.3
- How to get the image

Image URL (발표 자료에 포함)

test-network 실행 테스트

네트워크 실행

- ▣ `cd ~/fabric-samples/test-network`
- ▣ `./network.sh up`

네트워크 종료

- ▣ `./network.sh down`

