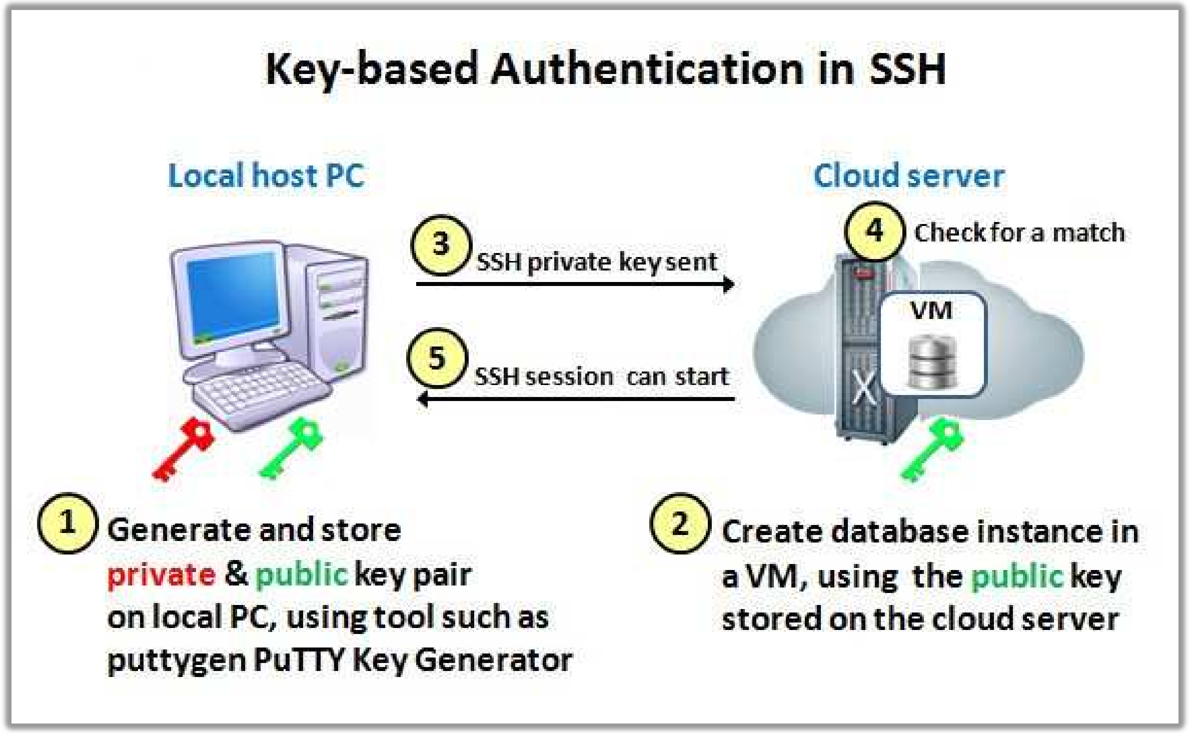
Docker 원격 개발환경 구성

SSH 비밀번호 없이 로그인 설정 (1/4)

- **SSH**
  - SSH(Secure Shell)는 원격지 호스트 컴퓨터에 접속하기 위해 사용되는 인터넷 프로토콜
  - 보안을 강화하기 위해 공개키 암호화 기능을 추가하여 CLI상의 명령을 암호화하여 전송하기 때문에 패킷이 노출되어도 안전함.
  - 원격접속을 위해 Public/Private Key-Pair가 필요함


Key-based Authentication in SSH
Local host PC
Cloud server
4 Check for a match
3 SSH private key sent
VM
5 SSH session can start
1 Generate and store private & public key pair on local PC, using tool such as puttygen PuTTY Key Generator
2 Create database instance in a VM, using the public key stored on the cloud server

https://medium.com/@hninja049/ssh-key-based-authentication-5816d6238c2

- **SSH 통신에 필요한 파일**

파일	설명
id_rsa	private key, 절대로 타인에게 노출되면 안됨
id_rsa.pub	public key, 접속하려는 리모트 머신의 authorized_keys에 입력
authorized_keys	리모트 머신의 .ssh 디렉토리 아래에 위치하면서 id_rsa.pub 키의 값을 저장

https://thisblogfor.me/web/ssh/

SSH 비밀번호 없이 로그인 설정 (2/4)

Client (windows)	Server (dockeredu)

✓ ssh 명령을 이용하여 dockeredu(원격서버) root 계정의 홈디렉토리 조회

```
C:\> ssh root@192.168.56.91 ls -l
The authenticity of host '192.168.56.91 (192.168.56.91)' can't be established.
ECDSA key fingerprint is SHA256:2o8VOam7Ps8tB1Xkf4qkNG/q30OudzZRCteUQ5aY2Yc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.91' (ECDSA) to the list of known hosts.
root@192.168.56.91's password: edu
total 20
drwxr-xr-x 2 root root    75 May  1  2019 gitlab-jenkins-sonarqube
drwxr-xr-x 2 root root    57 Jul 11  2019 guestbook
-rw-r--r-- 1 root root    14 Jul 11  2019 happy.txt
```

→ %USERPROFILE%/.ssh 폴더 생성

→ %USERPROFILE%/.ssh/known_hosts 파일에 fingerprint 저장

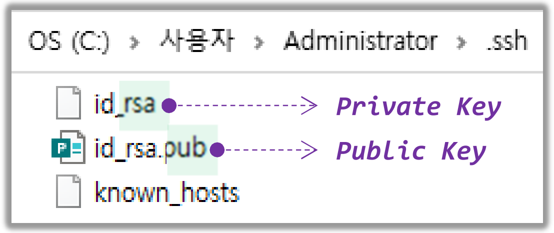
%USERPROFILE%/.ssh/known_hosts

```
192.168.56.91 ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAA
ABBBFd5PpojQHf+tcyAPfMnbhZQmNcK5xnlYePNiZhOynKkdK
Wqx0gmWVk+fWcXPdA5zEviKGAPHH1wMLEJde2uyDA=
```

✓ SSH Key-Pair (Public Key / Private Key) 생성

```
C:\> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Administrator/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Administrator/.ssh/id_rsa.
Your public key has been saved in C:\Users\Administrator/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:O1bzNh/BhA08GElkrw6iv/qoDbv+AKEcBchCwcTPHBs administrator@JHY
The key's randomart image is:
+---[RSA 3072]----+
|B=o.     o==.    |
|o+.E     .o.o+   |
|o.+ +       o.o  |
|o..=        . o  |
|o.     . S +   o |
| .   . . = o   . |
|  o .   + . + .  |
|   = o . . . o . |
|  .=+=o+.      . |
+----[SHA256]-----+
```

→ %USERPROFILE%/.ssh 폴더 하위에 Pubuic Key / Private Key 생성됨



SSH 비밀번호 없이 로그인 설정 (3/4)

Client (windows)

✓ ssh 명령을 이용하여 dockeredu(원격서버) root 계정의 홈디렉토리에 .ssh 폴더 생성

```
C:\> ssh root@192.168.56.91 mkdir /root/.ssh/
root@192.168.56.91's password: edu
```

✓ scp 명령을 이용하여 공개키(Public Key)를 dockeredu(원격서버) root 계정의 홈디렉토리/.ssh 폴더 하위에 authorized_keys 파일에 추가

```
C:\> scp %USERPROFILE%/.ssh/id_rsa.pub root@192.168.56.91:/root/.ssh/authorized_keys
root@192.168.56.91's password: edu
id_rsa.pub                100%  572   286.3KB/s   00:00

(MacOS 용)
yu3papa@yu3papaui-Mac ~ $ scp ~/.ssh/id_rsa.pub root@192.168.56.91:/root/.ssh/authorized_keys
root@192.168.56.91's password: edu
id_rsa.pub                100%  572   286.3KB/s   00:00
```

Server (dockeredu)

✓ root 계정의 홈경로에 .ssh 폴더가 생성되었는지 확인

```
[root@dockeredu ~]# ls -al ~/ | grep .ssh
drwxr-xr-x   2 root root     6 Jul 16 09:54 .ssh
```

✓ authorized_keys 파일에 공개키가 등록되었는지 확인

```
[root@dockeredu ~]# cat ~/.ssh/authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCwmZ6PRUcs0tPoCrIEqS0
OCMI4GCOjHya0mRzYYCZed5CLXxqoOPtXz7WKj4WstYiu7vRgN+
LtTqz828dl5WWc+piYuX1iOPBk5fwyD8M8YOvXp53Z3xrftqNOa
FpMaIDzzVRRkF+USbX9euO61M0IFWygL6YMXqZloAD3IsDGwUNh
IuHYiDliWPdCrKOiGDLxLgVbyaRKjUIwe6Zlu+bIvc+hkGYqOme
J9opl6JD/3DPWykB3hjaEExoRScSES8d+uYVobNO4LHpUPNHPob
g2pQ6pkbG920EShfF6eaFNZCOTzXX7UJMSLD6j8yTja9jxKXldb
ienxt8FPwGjgCQ7ztzn6U09jcSYR2B031gDKsnPMA+K6F5BfItg
9uOkstTaU1bLIg+TpK2VA36pQtL1P5b5HUUhd2bSez5CSzNkgzp
uG6MfuqI1bYDDVOayUTlrl6omej4qatVVzEV0Y3znqz3RXhLsAJ
NEJR0f59Y0FOtpEqz4kCMogGL8L7ZwYME=
administrator@JHY
```

SSH 비밀번호 없이 로그인 설정 (4/4)

Client (windows)	Server (dockeredu)

✓ Private Key를 이용하여 암호없이 로그인 되는지 확인

```
C:\> ssh -i %USERPROFILE%/.ssh/id_rsa root@192.168.56.91
Last login: Sat Jul 16 09:24:53 2022 from 192.168.56.1
[root@dockeredu ~]#
```

✓ ssh config 파일을 생성하여 HOST 앨리어스명으로 암호없이 로그인 설정

```
C:\> echo "" > %USERPROFILE%\.ssh\config
C:\> notepad %USERPROFILE%\.ssh\config
```

%USERPROFILE%/.ssh/config

```
Host dockeredu
    HostName 192.168.56.91
    User root
    IdentityFile C:/Users/Administrator/.ssh/id_rsa
```

사용자별 Private Key 파일 경로

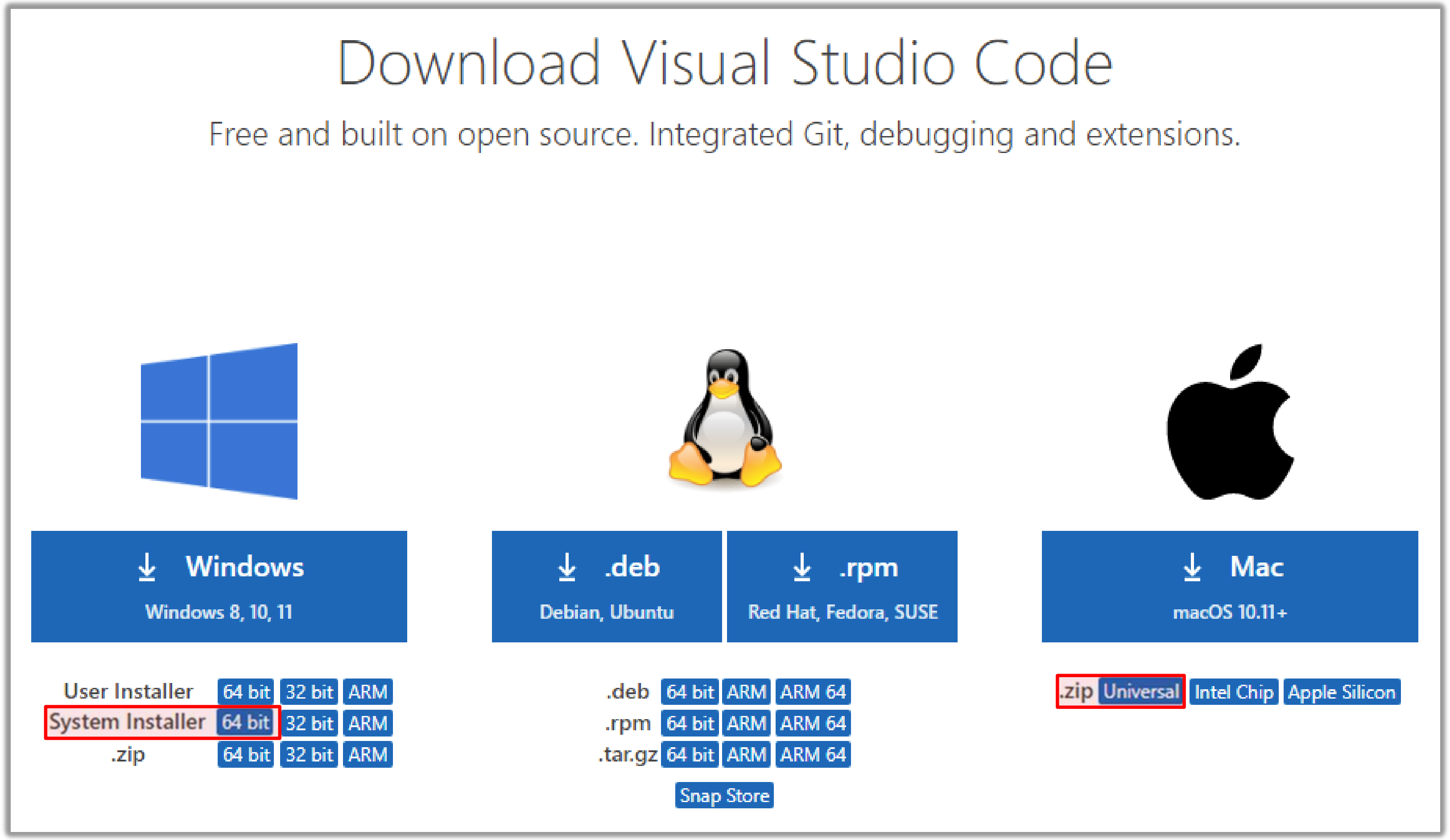
✓ HOST 앨리어스명으로 암호없이 로그인되는지 확인

```
C:\> ssh dockeredu
Last login: Sat Jul 16 10:05:46 2022 from 192.168.56.1

[root@dockeredu ~]# hostnamectl
   Static hostname: dockeredu
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 6cbda15e4b35d6478369c8e30c5f9cd3
           Boot ID: ce1a10e07731418e8b2a3655b7424a89
    Virtualization: kvm
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1160.el7.x86_64
      Architecture: x86-64
```

Visual Studio Code Remote – SSH (1/4)

- vscode 다운로드 + 설치
  - https://code.visualstudio.com/download


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

User Installer	64 bit	32 bit	ARM
System Installer	64 bit	32 bit	ARM
.zip	64 bit	32 bit	ARM

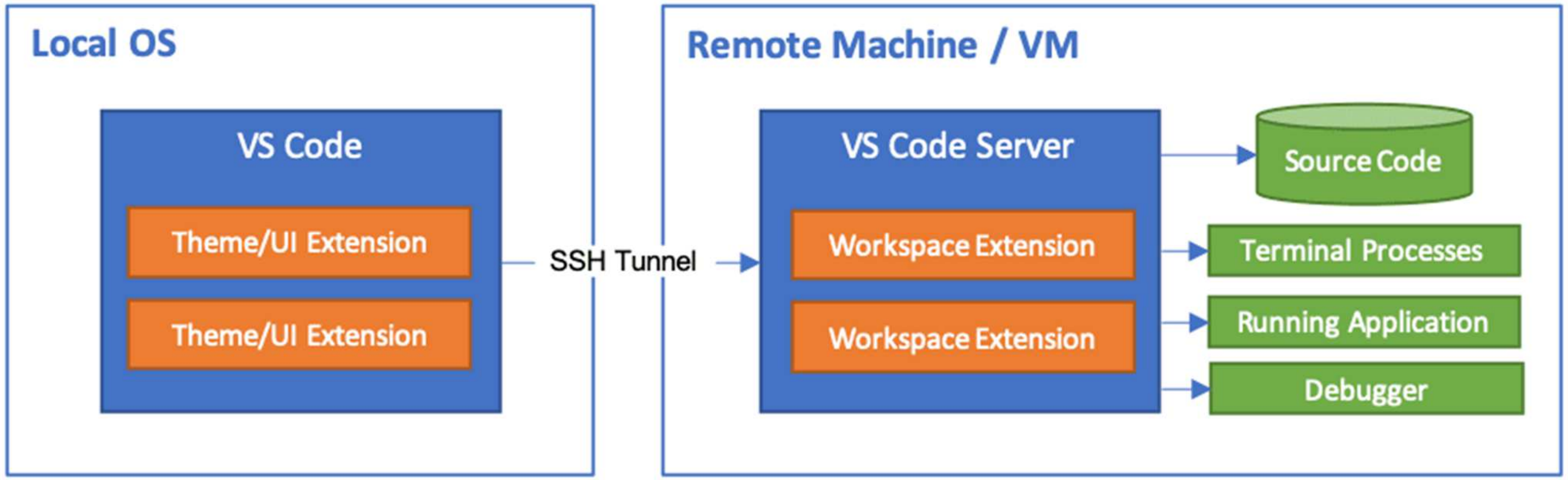
.deb	64 bit	ARM	ARM 64
.rpm	64 bit	ARM	ARM 64
.tar.gz	64 bit	ARM	ARM 64

Snap Store

.zip Universal Intel Chip Apple Silicon

Visual Studio Code Remote – SSH (2/4)

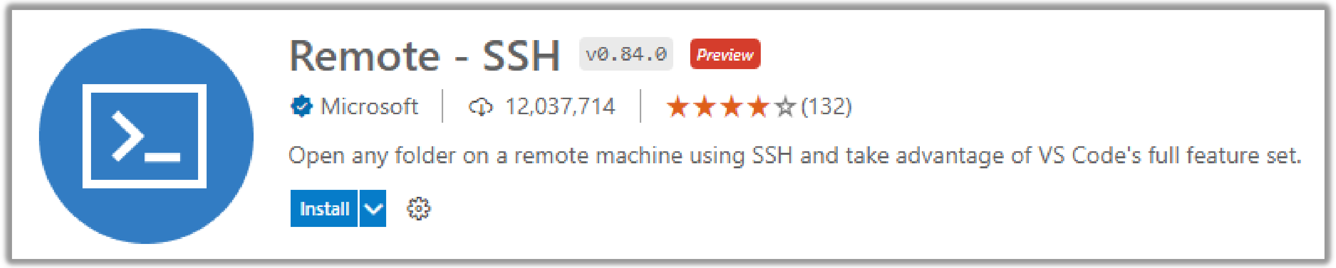
- **Remote - SSH extension**
  - The Remote - SSH extension lets you use any remote machine with a SSH server as your development environment. This can greatly simplify development and troubleshooting in a wide variety of situations. You can:
    - Develop on the same operating system you deploy to or use larger, faster, or more specialized hardware than your local machine.
    - Quickly swap between different, remote development environments and safely make updates without worrying about impacting your local machine.
    - Access an existing development environment from multiple machines or locations.
    - Debug an application running somewhere else such as a customer site or in the cloud.
  - No source code needs to be on your local machine to gain these benefits since the extension runs commands and other extensions directly on the remote machine.
  - You can open any folder on the remote machine and work with it just as you would if the folder were on your own machine.



IT Medical Center JADECROSS

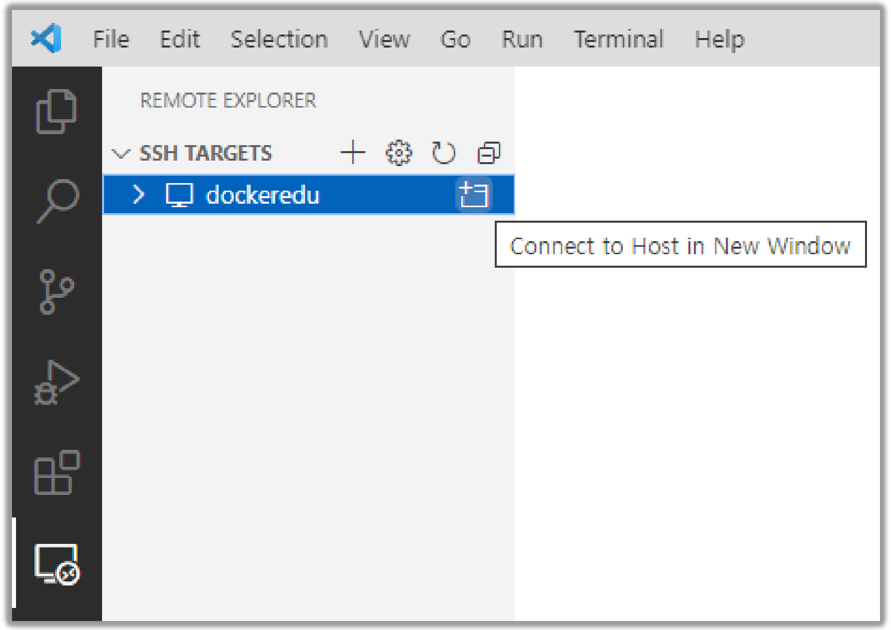
Visual Studio Code Remote – SSH (3/4)

- **Remote - SSH extension 설치**

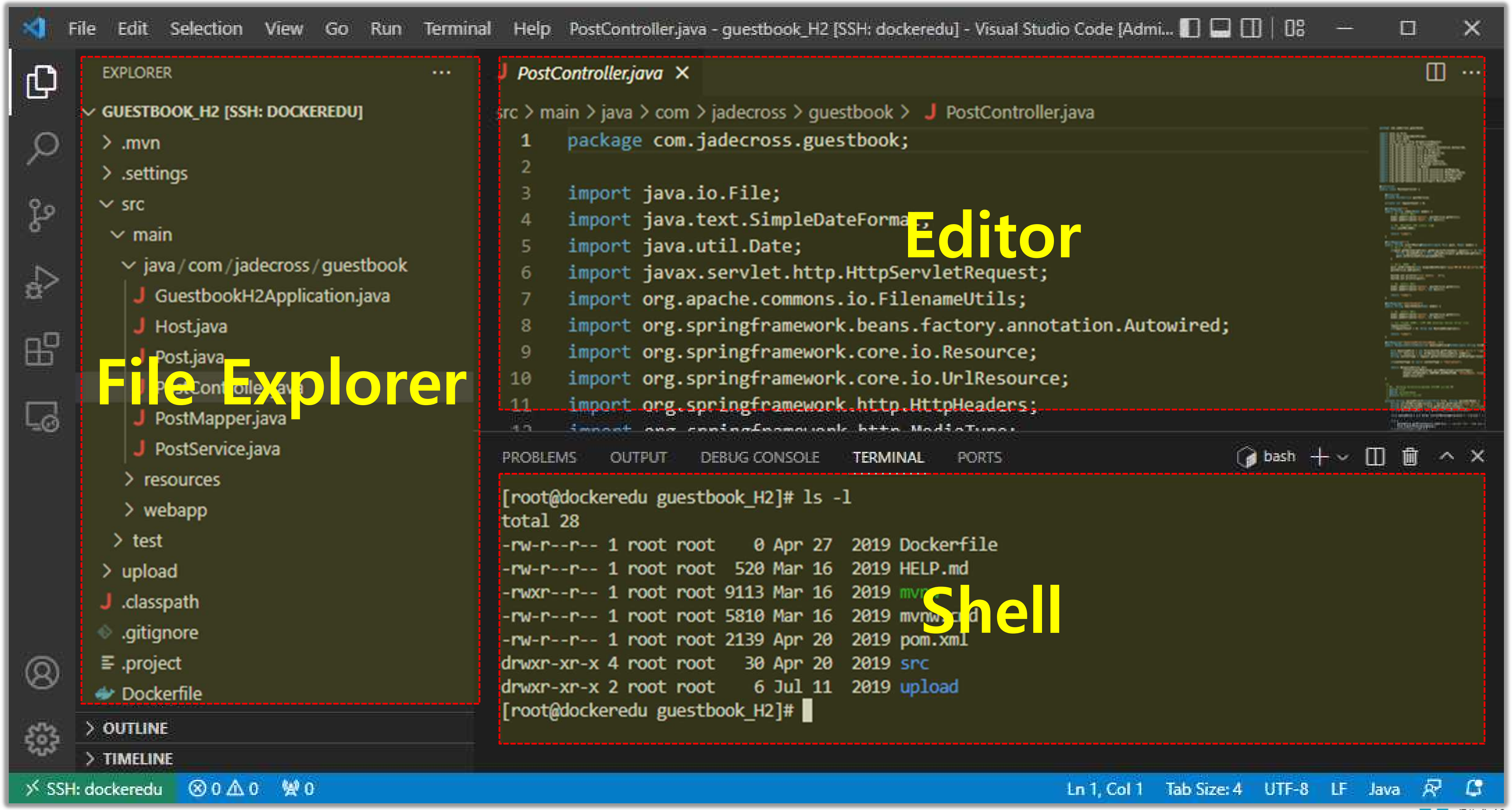


Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set.

- **Remote Explorer → dockeredu → Connect to Host In New Window**



Visual Studio Code Remote – SSH (4/4)


File Edit Selection View Go Run Terminal Help PostController.java - guestbook_H2 [SSH: dockeredu] - Visual Studio Code [Admi...
EXPLORER
GUESTBOOK_H2 [SSH: DOCKEREDU]
.mvn
.settings
src
main
java / com / jadecross / guestbook
GuestbookH2Application.java
Host.java
Post.java
PostController.java
PostMapper.java
PostService.java
resources
webapp
test
upload
.classpath
.gitignore
.project
Dockerfile
OUTLINE
TIMELINE
File Explorer
PostController.java
src > main > java > com > jadecross > guestbook > PostController.java
Editor
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Shell
SSH: dockeredu
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Java

```java
package com.jadecross.guestbook;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.http.HttpServletRequest;
import org.apache.commons.io.FilenameUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.Resource;
import org.springframework.core.io.UrlResource;
import org.springframework.http.HttpHeaders;
```

```
[root@dockeredu guestbook_H2]# ls -l
total 28
-rw-r--r-- 1 root root    0 Apr 27  2019 Dockerfile
-rw-r--r-- 1 root root  520 Mar 16  2019 HELP.md
-rwxr--r-- 1 root root 9113 Mar 16  2019 mvnw
-rw-r--r-- 1 root root 5810 Mar 16  2019 mvnw.cmd
-rw-r--r-- 1 root root 2139 Apr 20  2019 pom.xml
drwxr-xr-x 4 root root   30 Apr 20  2019 src
drwxr-xr-x 2 root root    6 Jul 11  2019 upload
[root@dockeredu guestbook_H2]#
```

도커 데몬

- **dockerd**
  - https://docs.docker.com/engine/reference/commandline/dockerd/
  - 디폴트 설정으로 로컬에서 Unix Domain Socket을 이용한 접속만 가능

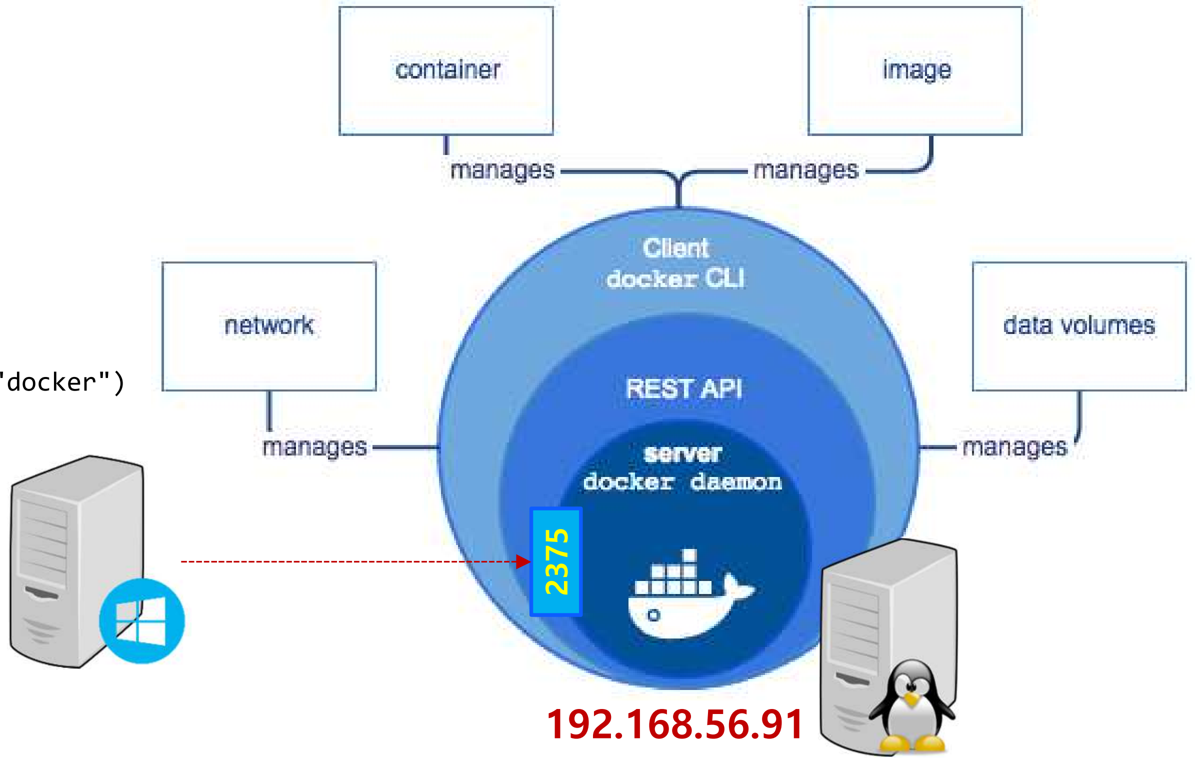
✓ dockerd 도움말을 출력하고 –H 옵션의 의미 파악

```
[root@dockeredu ~]# dockerd --help

Usage:  dockerd [OPTIONS]

A self-sufficient runtime for containers.

Options:
~~~ 중간생략 ~~~
  -G, --group string              Group for the unix socket (default "docker")
      --help                      Print usage
  -H, --host list                 Daemon socket(s) to connect to
~~~ 이하생략 ~~~
```



Docker 원격 실행 환경 (1/6)

- **Docker 데몬 설정**
  - 로컬에서는 Unix Domain Socket을 이용
  - 원격 클라이언트를 위해 TCP 소켓 이용

✓ dockeredu 서버에서 원격 클라이언트 접속을 위한 설정 추가
- **-H tcp://0.0.0.0**

```
[root@dockeredu ~]# vi /usr/lib/systemd/system/docker.service
```

/usr/lib/systemd/system/docker.service

```
~~~중간생략~~~
[Service]
Type=notify
# the default is not to use systemd for cgroups because the delegate issues still
# exists and systemd currently does not support the cgroup feature set required
# for containers run by docker
ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock -H tcp://0.0.0.0
ExecReload=/bin/kill -s HUP $MAINPID
~~~이하생략~~~
```

추가

✓ docker 데몬 재시작

```
[root@dockeredu ~]# systemctl daemon-reload
[root@dockeredu ~]# systemctl restart docker
[root@dockeredu ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2022-06-28 16:29:22 KST; 13s ago
     Docs: https://docs.docker.com
 Main PID: 11382 (dockerd)
    Tasks: 9
   Memory: 36.4M
   CGroup: /system.slice/docker.service
           └─11382 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock -H tcp://0.0.0.0
```

```
WARNING: API is accessible on http://0.0.0.0:2375 without encryption.
         Access to the remote API is equivalent to root access on the host. Refer
         to the 'Docker daemon attack surface' section in the documentation for
         more information: https://docs.docker.com/go/attack-surface/
```

IT Medical Center JADECROSS

Docker 원격 실행 환경 (2/6)

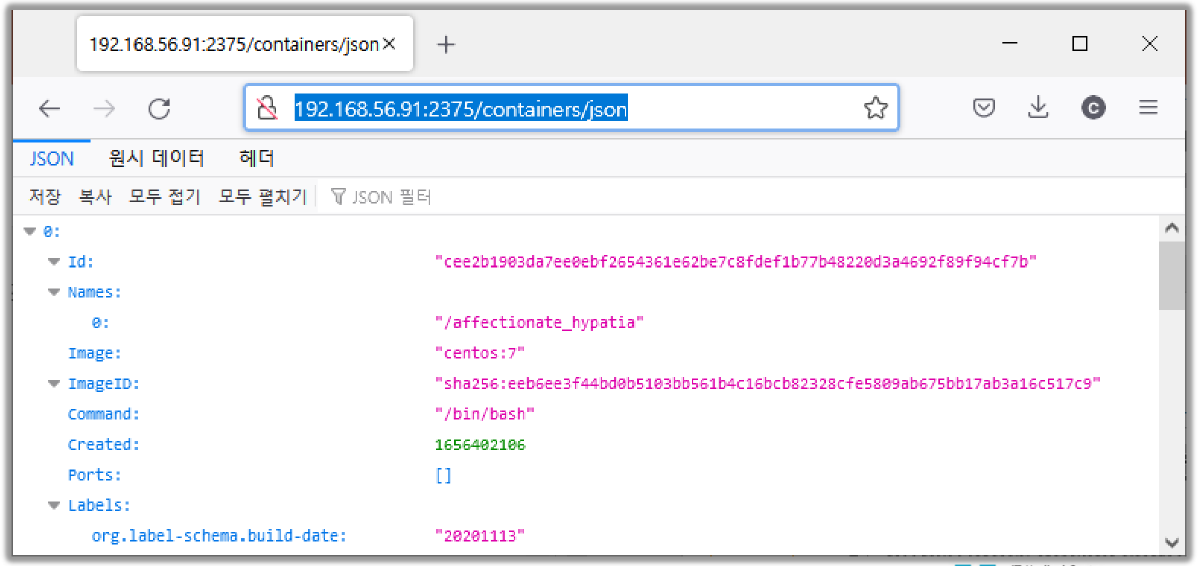
- **윈도우에서 REST API를 이용한 도커 데몬 접속**
  - 도커 REST API
    - https://docs.docker.com/engine/api/v1.41/

✓ dockeredu(192.168.56.91) 서버에서 실행중인 도커 데몬에 윈도우 머신에서 REST API를 이용하여 컨테이너 목록 출력

```
c:\> curl http://192.168.56.91:2375/containers/json
[{"Id":"cee2b1903da7ee0ebf2654361e62be7c8fdef1b77b48220d3a4692f89f94cf7b","Names":["/affectionate_hypatia"],"Image":"centos:7","ImageID":"sha256:eeb6e
e3f44bd0b5103bb561b4c16bcb82328cfe5809ab675bb17ab3a16c517c9","Command":"/bin/bash","Created":1656402106,"Ports":[],"Labels":{"org.label-schema.build-
date":"20201113","org.label-schema.license":"GPLv2","org.label-schema.name":"CentOS Base Image","org.label-schema.schema-version":"1.0","org.label-
schema.vendor":"CentOS","org.opencontainers.image.created":"2020-11-13 00:00:00+00:00","org.opencontainers.image.licenses":"GPL-2.0-
only","org.opencontainers.image.title":"CentOS Base Image","org.opencontainers.image.vendor":"CentOS"},"State":"running","Status":"Up 32
seconds","HostConfig":{"NetworkMode":"default"},"NetworkSettings":{"Networks":{"bridge":{"IPAMConfig":null,"Links":null,"Aliases":null,"NetworkID":"27
c52fb5e72b1949a1b33c6ddf59787668d1e359cfc4e650ee17019ccf512a4f","EndpointID":"302fac57f196fca1763a940a4831ce9617349c06e8219ba99f67e3b1784212d4","Gatew
ay":"172.17.0.1","IPAddress":"172.17.0.2","IPPrefixLen":16,"IPv6Gateway":"","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"MacAddress":"02:42:ac:11:0
0:02","DriverOpts":null}}},"Mounts":[]}]
```

✓ 브라우저를 이용하여 원격 도커 데몬에 REST API를 이용하여 컨테이너 목록 출력

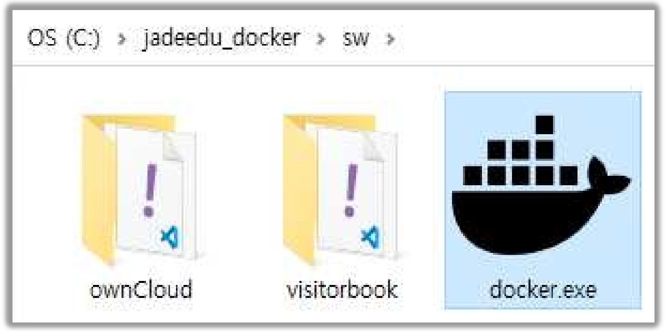
http://192.168.56.91:2375/containers/json


192.168.56.91:2375/containers/json
192.168.56.91:2375/containers/json
JSON
원시 데이터
헤더
저장
복사
모두 접기
모두 펼치기
JSON 필터

```
0:
  Id:                              "cee2b1903da7ee0ebf2654361e62be7c8fdef1b77b48220d3a4692f89f94cf7b"
  Names:
    0:                             "/affectionate_hypatia"
  Image:                           "centos:7"
  ImageID:                         "sha256:eeb6ee3f44bd0b5103bb561b4c16bcb82328cfe5809ab675bb17ab3a16c517c9"
  Command:                         "/bin/bash"
  Created:                         1656402106
  Ports:                           []
  Labels:
    org.label-schema.build-date:   "20201113"
```

Docker 원격 실행 환경 (3/6)

- 윈도우용 docker 클라이언트 프로그램 이용
  - 윈도우용 도커 클라이언트 프로그램 다운로드
    - https://download.docker.com/win/static/stable/x86_64/



✓ "C:\jadeedu_docker\sw" 경로에서 윈도우용 docker.exe 프로그램을 이용하여 dockeredu(192.168.56.91)에서 실행중인 도커 컨테이너 목록 조회

```
C:\jadeedu_docker\sw> docker -H 192.168.56.91 container ls
CONTAINER ID   IMAGE      COMMAND       CREATED          STATUS         PORTS     NAMES
cee2b1903da7   centos:7   "/bin/bash"   7 minutes ago    Up 7 minutes             affectionate_hypatia

C:\jadeedu_docker\sw> docker -H 192.168.56.91 image ls
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
centos        7         eeb6ee3f44bd   9 months ago    204MB
```

✓ 윈도우 OS상의 PATH 환경변수 경로중 한곳에 docker.exe 파일 복사

```
C:\jadeedu_docker\sw> copy docker.exe %SystemRoot%\
C:\jadeedu_docker\sw> cd C:\
C:\> docker -H 192.168.56.91 image ls
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
centos        7         eeb6ee3f44bd   9 months ago    204MB
```

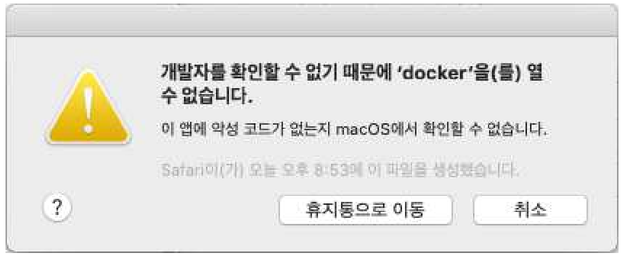
Mac OS용) Docker 원격 실행 환경 (3/6)

- **docker 클라이언트 프로그램 이용**

✓ (사전준비) Mac OS 보안 설정 – 다운로드된 앱 AnyWhere 실행

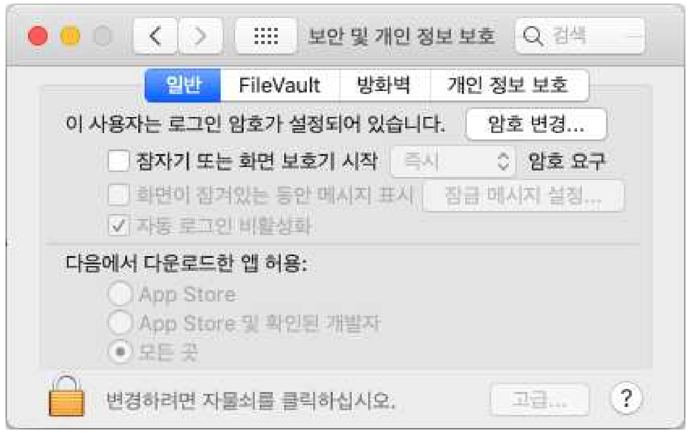
```
yu3papa@yu3papaui-Mac ~ % sudo spctl --master-disable
Password:
```

→ Mac OS 로그 아웃 → 재 로그인 후 "보안 및 개인 정보 보호" 확인


개발자를 확인할 수 없기 때문에 'docker'을(를) 열 수 없습니다.

이 앱에 악성 코드가 없는지 macOS에서 확인할 수 없습니다.

Safari이(가) 오늘 오후 8:53에 이 파일을 생성했습니다.



이 사용자는 로그인 암호가 설정되어 있습니다. 암호 변경...

☐ 잠자기 또는 화면 보호기 시작 즉시 암호 요구

☐ 화면이 잠겨있는 동안 메시지 표시 잠금 메시지 설정...

✓ 자동 로그인 비활성화

다음에서 다운로드한 앱 허용:
- ○ App Store
- ○ App Store 및 확인된 개발자
- [✓ 모든 곳]

변경하려면 자물쇠를 클릭하십시오.

✓ Mac OS 용 도커 클라이언트 프로그램 다운로드 후 압축 해제

```
yu3papa@yu3papaui-Mac ~ % curl -O https://download.docker.com/mac/static/stable/x86_64/docker-19.03.9.tgz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 42.8M  100 42.8M    0     0  8053k      0  0:00:05  0:00:05 --:--:-- 9330k
yu3papa@yu3papaui-Mac ~ % tar xfz docker-19.03.9.tgz
yu3papa@yu3papaui-Mac ~ % cd docker
```

✓ docker 클라이언트 프로그램을 이용하여 dockeredu(192.168.56.91)에서 실행중인 도커 컨테이너 목록 조회

```
yu3papa@yu3papaui-Mac docker % ./docker -H 192.168.56.91 container ls
CONTAINER ID   IMAGE      COMMAND       CREATED         STATUS         PORTS     NAMES
cee2b1903da7   centos:7   "/bin/bash"   7 minutes ago   Up 7 minutes             affectionate_Hypatia
```

✓ Mac OS상의 PATH 환경변수 경로중 한곳에 docker 실행파일 복사 후 docker image 조회

```
yu3papa@yu3papaui-Mac docker % sudo cp docker /usr/local/bin

yu3papa@yu3papaui-Mac ~ % docker -H 192.168.56.91 image ls
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
centos       7         eeb6ee3f44bd   9 months ago   204MB
```

참고주소 - https://minmong.tistory.com/170

Docker 원격 실행 환경 (4/6)

- **docker context (1/3)**
  - https://docs.docker.com/engine/reference/commandline/context/

✓ docker context 도움말을 확인하고 사용 가능한 sub command 확인

```
C:\> docker context --help

Usage:  docker context COMMAND

Manage contexts

Commands:
  create      Create a context
  export      Export a context to a tar or kubeconfig file
  import      Import a context from a tar or zip file
  inspect     Display detailed information on one or more contexts
  ls          List contexts
  rm          Remove one or more contexts
  update      Update a context
  use         Set the current docker context

Run 'docker context COMMAND --help' for more information on a command.
```

Docker 원격 실행 환경 (5/6)

- **docker context (2/3)**
  - https://docs.docker.com/engine/reference/commandline/context/

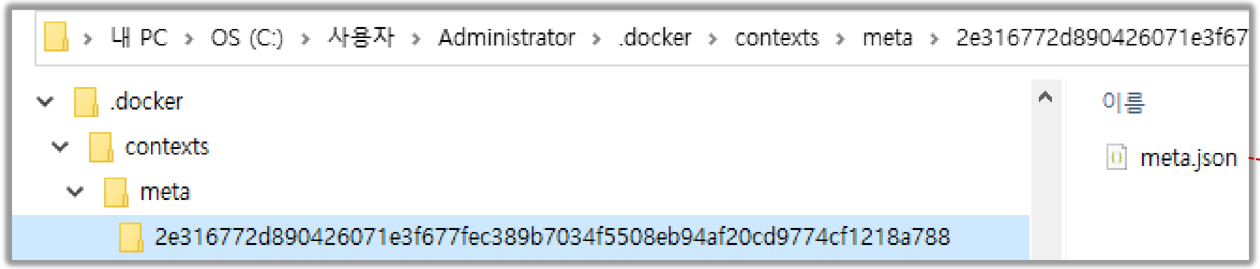
✓ 현재 docker.exe 클라이언트 머신의 context 확인

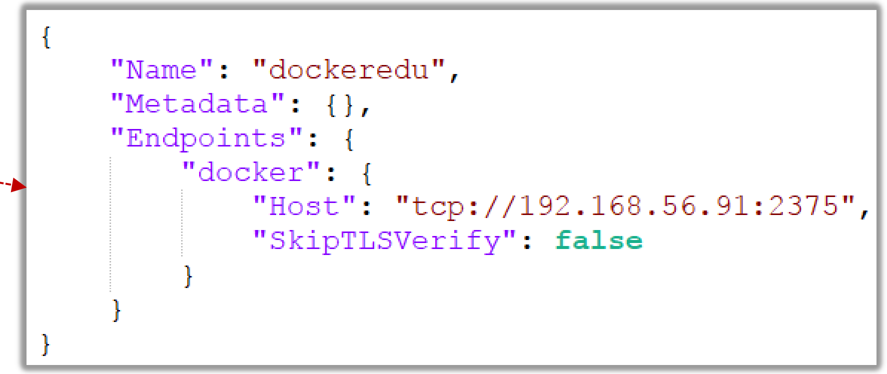
```
C:\> docker context ls
NAME         DESCRIPTION                                DOCKER ENDPOINT                KUBERNETES ENDPOINT   ORCHESTRATOR
default      Current DOCKER_HOST based configuration    npipe:////./pipe/docker_engine                        swarm
```

✓ dockeredu(192.168.56.91) 서버의 도커 데몬에 대한 context 를 생성하고, 조회

```
C:\> docker context create --docker host=tcp://192.168.56.91:2375 dockeredu
dockeredu
Successfully created context "dockeredu“
```

→ docker context 가 생성되면 윈도우 OS 사용자의 홈경로\.docker\contexts\meta\<UUID>\meta.json 파일이 생성됨


내 PC › OS (C:) › 사용자 › Administrator › .docker › contexts › meta › 2e316772d890426071e3f67
.docker
contexts
meta
2e316772d890426071e3f677fec389b7034f5508eb94af20cd9774cf1218a788
이름
meta.json



```
{
    "Name": "dockeredu",
    "Metadata": {},
    "Endpoints": {
        "docker": {
            "Host": "tcp://192.168.56.91:2375",
            "SkipTLSVerify": false
        }
    }
}
```

```
C:\> docker context ls
NAME         DESCRIPTION                                DOCKER ENDPOINT                KUBERNETES ENDPOINT   ORCHESTRATOR
default *    Current DOCKER_HOST based configuration    npipe:////./pipe/docker_engine                        swarm
dockeredu                                               tcp://192.168.56.91:2375
```

Docker 원격 실행 환경 (6/6)

- **docker context (3/3)**
  - https://docs.docker.com/engine/reference/commandline/context/

✓ dockeredu 컨텍스트 사용 설정하고 docker 명령을 이용하여 실행중인 컨테이너 목록 조회

```
C:\jadeedu_docker\sw> docker context use dockeredu
dockeredu
Current context is now "dockeredu"
```

→ docker context 가 설정되면 홈경로\.docker\config.json 의 내용이 변경됨

```
{
    "auths": {},
    "currentContext": "dockeredu"
}
```

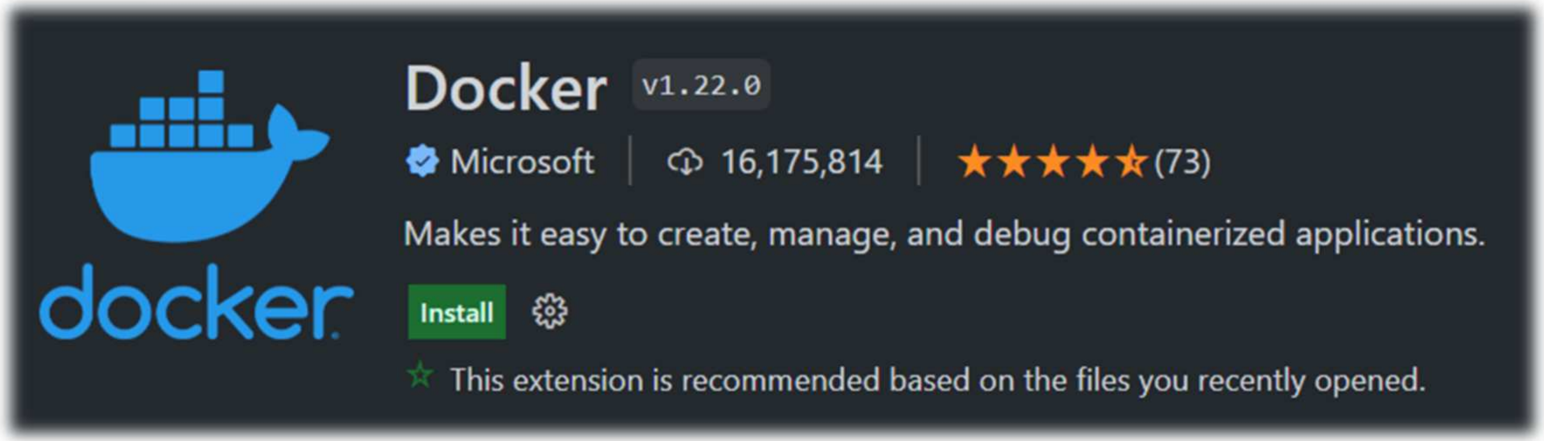
```
C:\jadeedu_docker\sw> docker context ls
NAME          DESCRIPTION                               DOCKER ENDPOINT                    KUBERNETES ENDPOINT   ORCHESTRATOR
default       Current DOCKER_HOST based configuration   npipe:////./pipe/docker_engine                           swarm
dockeredu *                                             tcp://192.168.56.91:2375

C:\jadeedu_docker\sw> docker container ls
CONTAINER ID   IMAGE      COMMAND       CREATED          STATUS          PORTS     NAMES
cee2b1903da7   centos:7   "/bin/bash"   37 minutes ago   Up 37 minutes             affectionate_hypatia
```

vscode로 원격 도커 컨테이너 개발환경 설정 (1/4)

- **Docker in Visual Studio Code**
  - https://code.visualstudio.com/docs/containers/overview

- **Extension 설치**
  - Docker



Docker v1.22.0

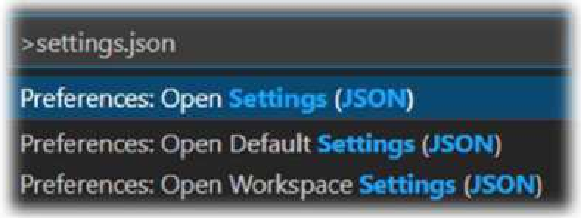
Microsoft | 16,175,814 | ★★★★☆(73)

Makes it easy to create, manage, and debug containerized applications.

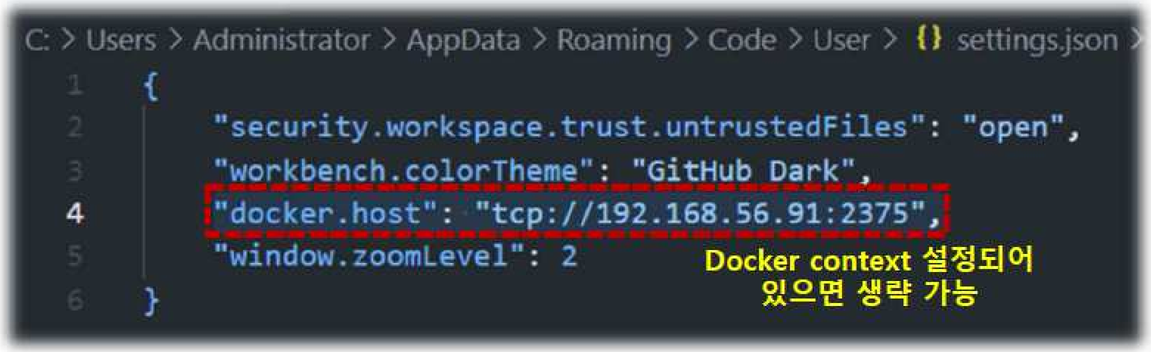
This extension is recommended based on the files you recently opened.

vscode로 원격 도커 컨테이너 개발환경 설정 (2/4)

- 옵션) settings.json 수정
  - Default 나 Workspace의 settings.json 수정하면 안됨!!!



```
>settings.json
Preferences: Open Settings (JSON)
Preferences: Open Default Settings (JSON)
Preferences: Open Workspace Settings (JSON)
```

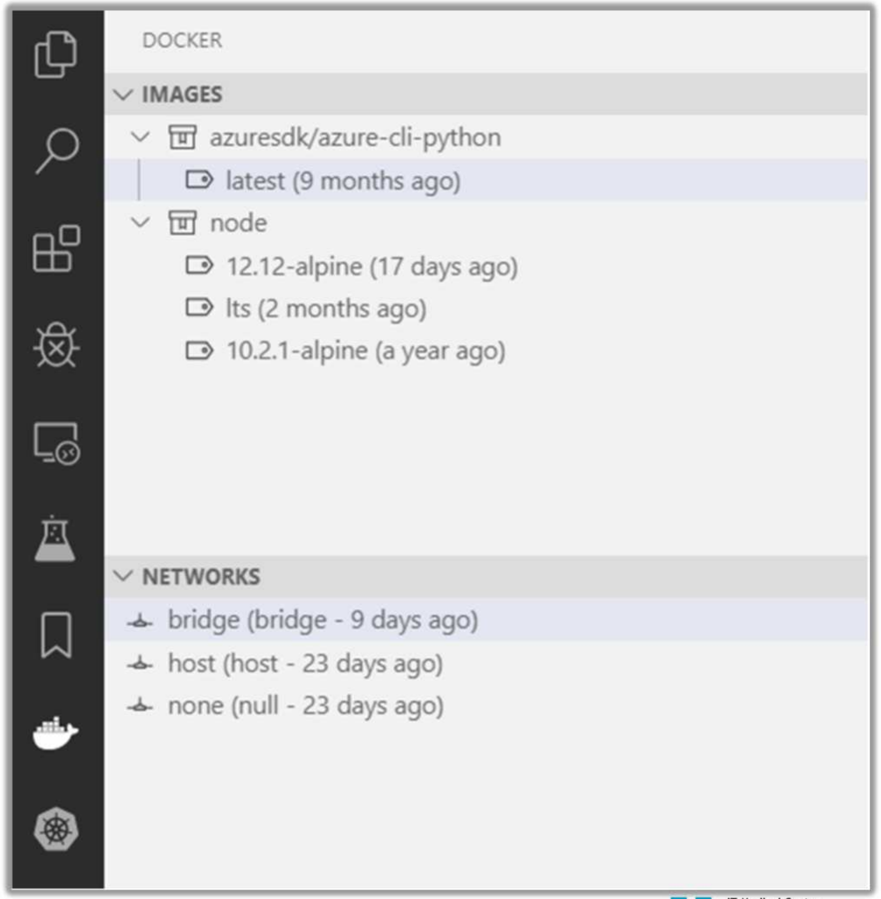


C: > Users > Administrator > AppData > Roaming > Code > User > {} settings.json >

```
{
    "security.workspace.trust.untrustedFiles": "open",
    "workbench.colorTheme": "GitHub Dark",
    "docker.host": "tcp://192.168.56.91:2375",
    "window.zoomLevel": 2
}
```

Docker context 설정되어 있으면 생략 가능

- Docker Explorer

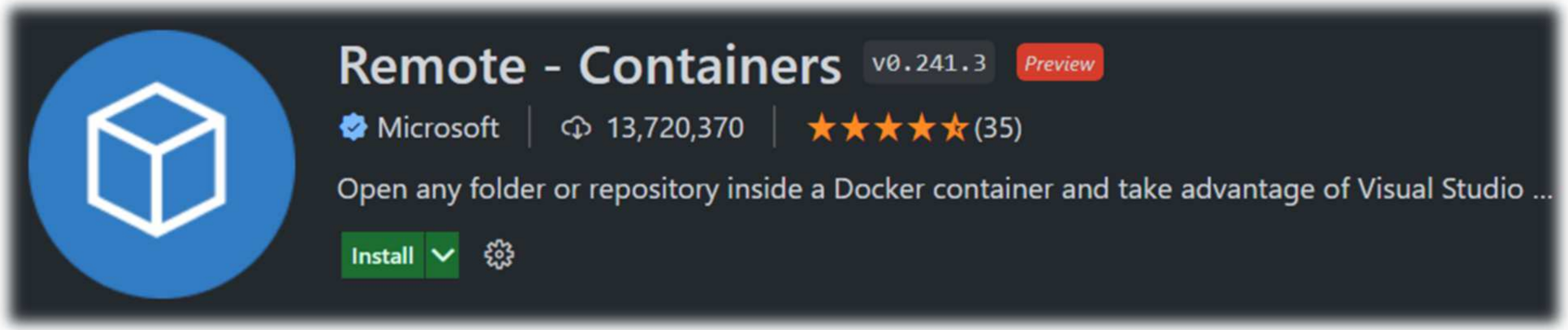


```
DOCKER
IMAGES
  azuresdk/azure-cli-python
    latest (9 months ago)
  node
    12.12-alpine (17 days ago)
    lts (2 months ago)
    10.2.1-alpine (a year ago)
NETWORKS
  bridge (bridge - 9 days ago)
  host (host - 23 days ago)
  none (null - 23 days ago)
```

vscode로 원격 도커 컨테이너 개발환경 설정 (3/4)

- **Developing inside a Container**
  - https://code.visualstudio.com/docs/remote/containers

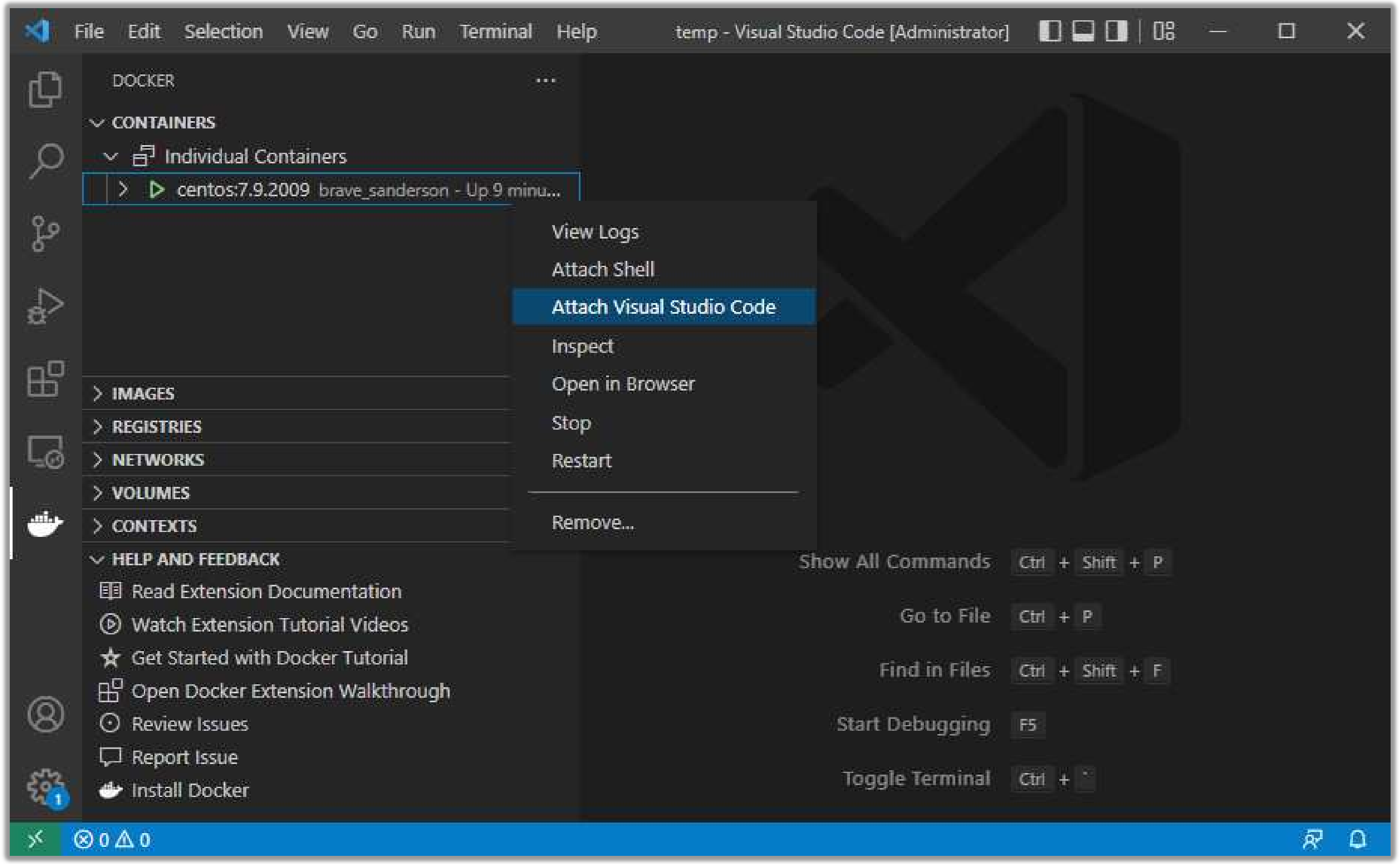
- **Extension 설치**
  - Remote - Containers



Open any folder or repository inside a Docker container and take advantage of Visual Studio ...

vscode로 원격 도커 컨테이너 개발환경 설정 (4/4)

- **Attach Visual Studio Code**


File Edit Selection View Go Run Terminal Help
temp - Visual Studio Code [Administrator]
DOCKER
CONTAINERS
Individual Containers
centos:7.9.2009 brave_sanderson - Up 9 minu...
View Logs
Attach Shell
Attach Visual Studio Code
Inspect
Open in Browser
Stop
Restart
Remove...
IMAGES
REGISTRIES
NETWORKS
VOLUMES
CONTEXTS
HELP AND FEEDBACK
Read Extension Documentation
Watch Extension Tutorial Videos
Get Started with Docker Tutorial
Open Docker Extension Walkthrough
Review Issues
Report Issue
Install Docker
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + `
0 0

IT Medical Center JADECROSS

LAB) vscode로 원격 도커 컨테이너 nodejs 개발 (1/2)

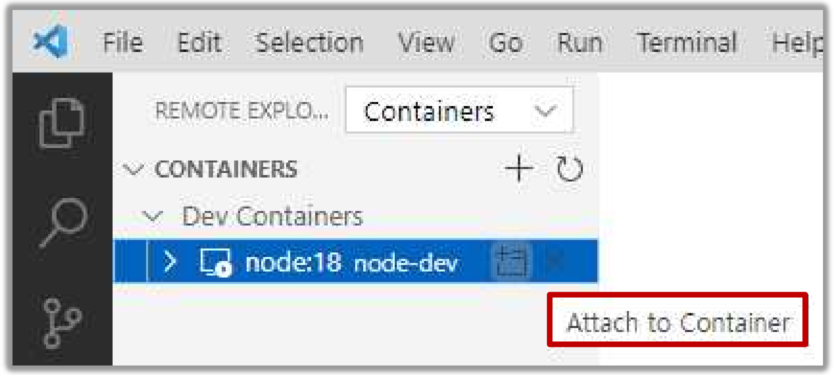
✓ nodejs용 작업 폴더 생성

```
[root@dockeredu ~]# mkdir -p ~/workspace/hellonodejs
```

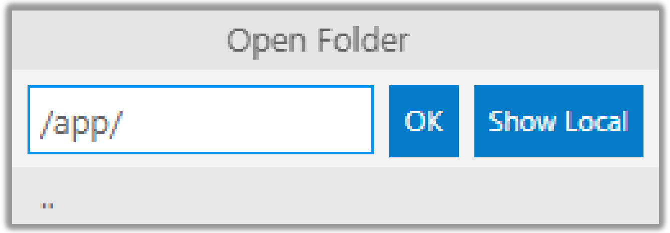
✓ nodejs 런타임이 설치된 컨테이너 실행

```
[root@dockeredu ~]# docker container run -d -it \
 --name=node-dev \
 -v /root/workspace/hellonodejs:/app \
 --entrypoint=/bin/bash \
 node:18
1654c98bc765b4a257ca3426788103228749fe0407a298efa62f0fe004adb825
```

✓ vscode에서 원격 컨테이너 Attach



✓ /app 폴더 오픈

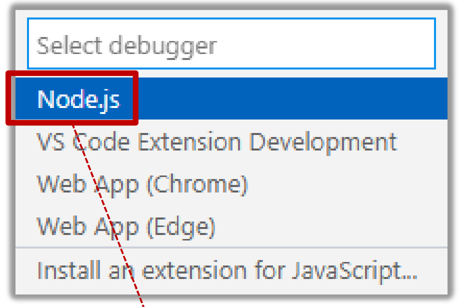


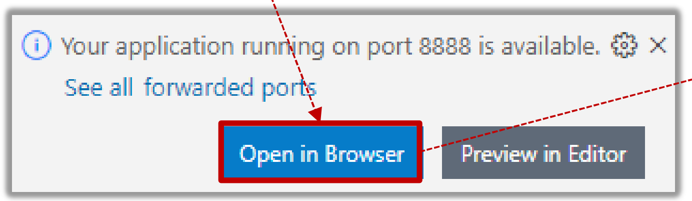
LAB) vscode로 원격 도커 컨테이너 nodejs 개발 (2/2)

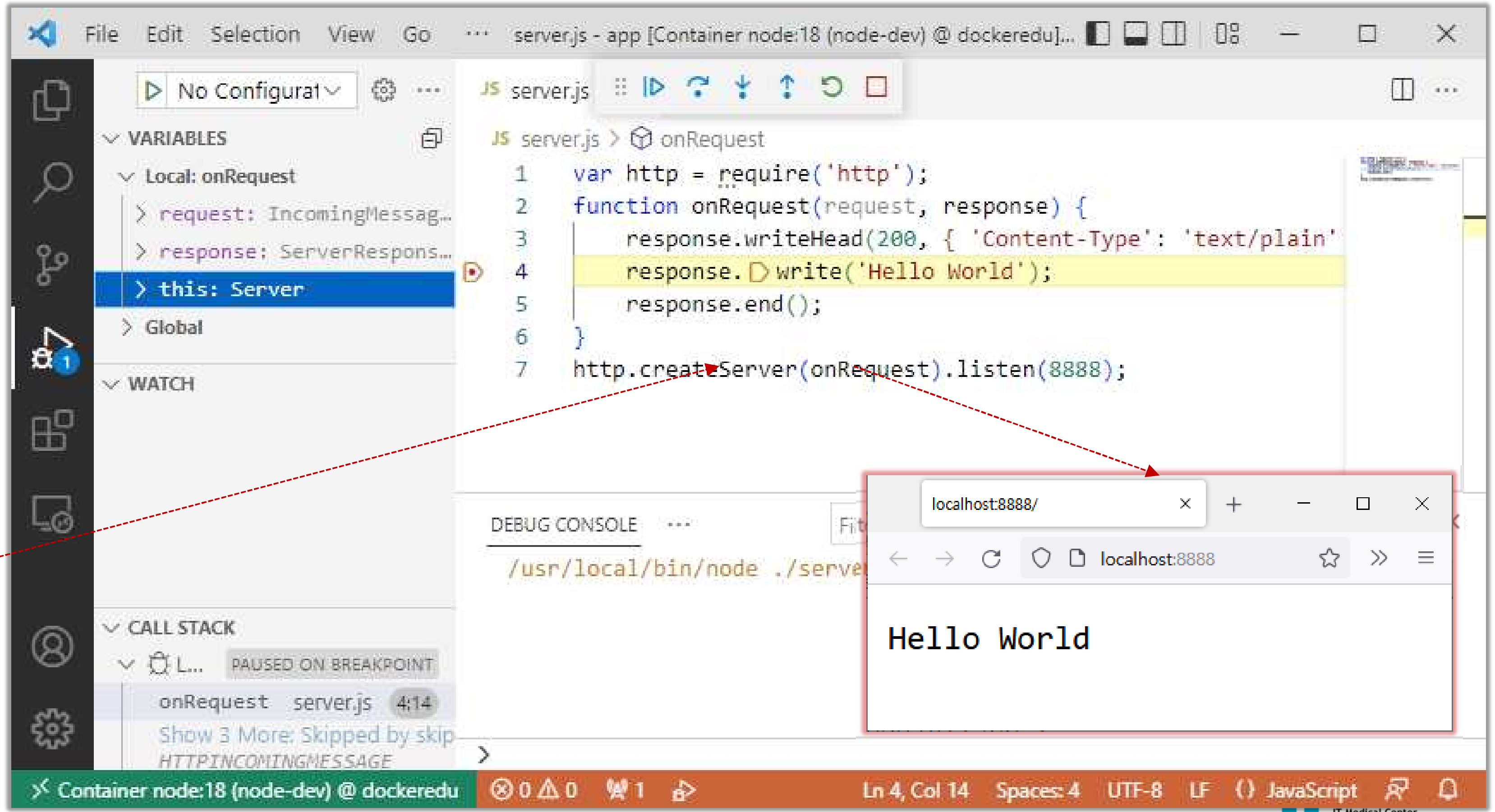
✓ server.js 파일 추가

```
var http = require('http');
function onRequest(request, response) {
    response.writeHead(200, { 'Content-Type': 'text/plain' });
    response.write('Hello World');
    response.end();
}
http.createServer(onRequest).listen(8888);
```

✓ 디버그 포인트 설정 후 "F5" 버튼 클릭
→ "Node.js" 디버거 선택
→ "Open in Browser"


Select debugger
Node.js
VS Code Extension Development
Web App (Chrome)
Web App (Edge)
Install an extension for JavaScript...


Your application running on port 8888 is available.


File Edit Selection View Go ...
server.js - app [Container node:18 (node-dev) @ dockeredu]...
No Configurati
VARIABLES
Local: onRequest
request: IncomingMessag...
response: ServerRespons...
this: Server
Global
WATCH
CALL STACK
PAUSED ON BREAKPOINT
onRequest server.js 4:14
Show 3 More: Skipped by skip
HTTPINCOMINGMESSAGE
server.js > onRequest
DEBUG CONSOLE
Container node:18 (node-dev) @ dockeredu
Ln 4, Col 14 Spaces: 4 UTF-8 LF JavaScript
localhost:8888/
localhost:8888

```
var http = require('http');
function onRequest(request, response) {
    response.writeHead(200, { 'Content-Type': 'text/plain'
    response.write('Hello World');
    response.end();
}
http.createServer(onRequest).listen(8888);
```

```
/usr/local/bin/node ./serve
```

```
Hello World
```

JADECROSS