--------------------------------------------

[시간표]

9시~

11시30분~오후 1시 (점심)

오후5시50분

-----------------------------------------------

\* 소프트웨어 개발 (품질, 테스트)

\* TDD, Test 자동화.

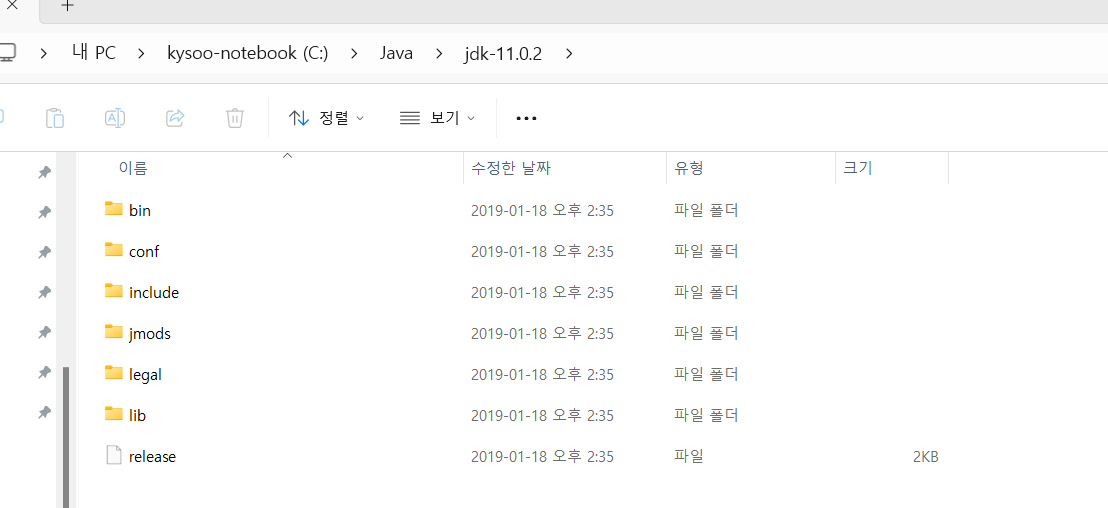
---------------------------------------

\* 환경변수 :

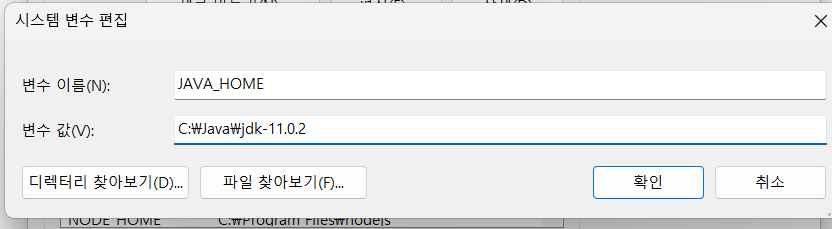
- 윈도우 : sysdm.cpl

- 리눅스 : export, $MYENV

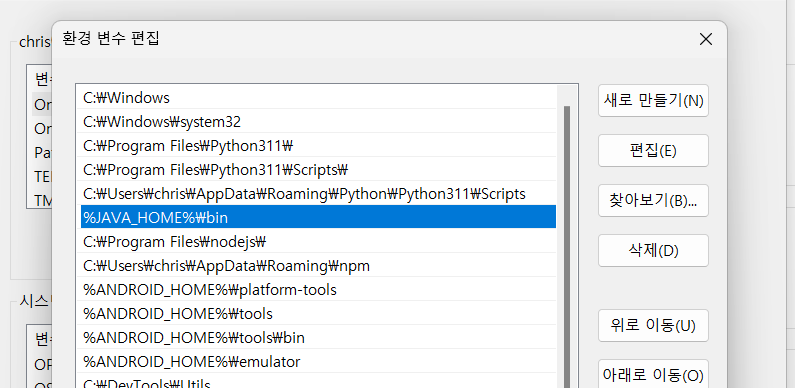
\* JDK를 설치 (압축풀고) JAVA\_HOME을 등록.



-> 환경변수 : JAVA\_HOME=C:\Java\jdk-11.0.2



PATH :



\* 가상머신 만들고 활용.

- 개발하다 보면, 운영 환경에 준한 가상머신(Host)가 필요

- 테스트를 위해서 테스트 환경 구축.

--------------------------------------------------------------

ㅇ 윈도우에서 리눅스를 사용하는 방법

- 클라우드의 리눅스저 접속

- WLS2, 리눅스 도커 컨테이너

- VirtuaBox

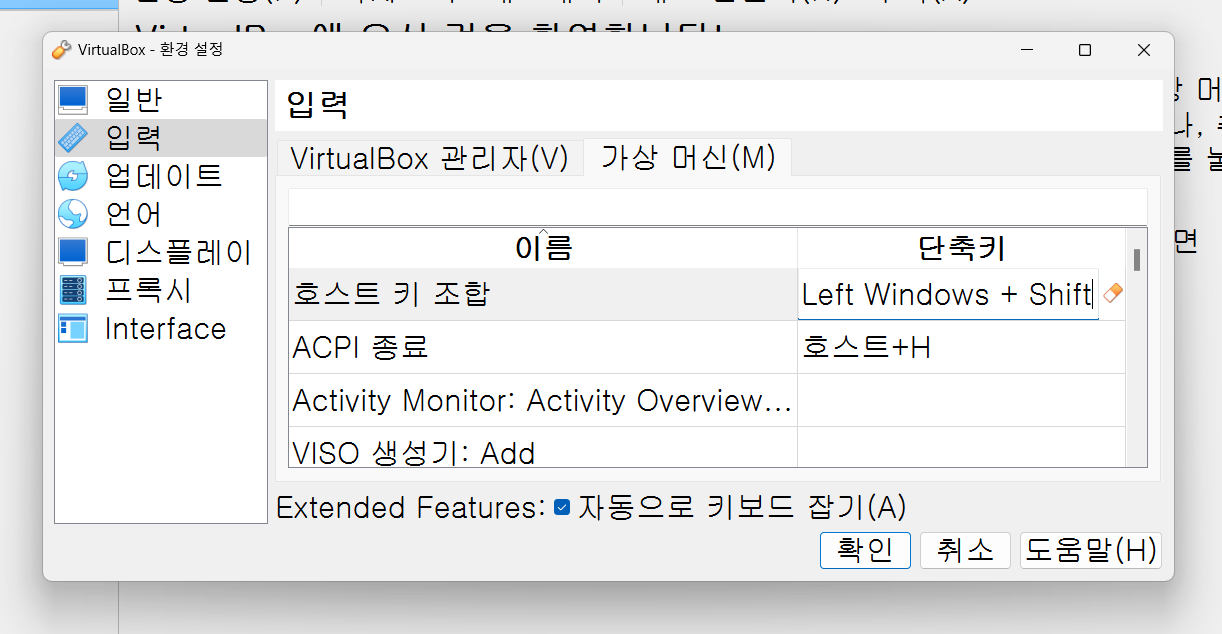
- VMWare, VMPlayer

- HyperVisor

----------

**<참고 : 윈도우 WSL2 와 Docker 설치>**

| ※ Windows 10 버전 2004 이상(빌드 19041 이상) 또는 Windows 11을 실행해야 합니다.  <하이퍼바이저 활성화>   * <https://docs.microsoft.com/ko-KR/virtualization/hyper-v-on-windows/reference/hyper-v-requirements>   => systeminfo 명령어   1. 관리자 권한으로 PowerShell 콘솔을 엽니다. 2. 다음 명령 실행:   <PowerShell복사>  **Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All**  -------------  <wsl 설정>   * <https://docs.microsoft.com/ko-kr/windows/wsl/install>   ※<https://docs.microsoft.com/ko-kr/windows/wsl/setup/environment>  관리자 파워쉘 : **wsl --install**  (※ 필요시 -d ubuntu 옵션 추가 지정)  => WSL2 활성화 및 Ubuntu 설치까지 한번에 완료      --------  <Docker>  - 설치 프로그램 사용 : [Docker: Accelerated, Containerized Application Development](https://www.docker.com/)    - <https://www.docker.com/get-started/> |
| --- |



ㅇ 개발환경 구성시 꼭 참고

- 한글X (계정명)

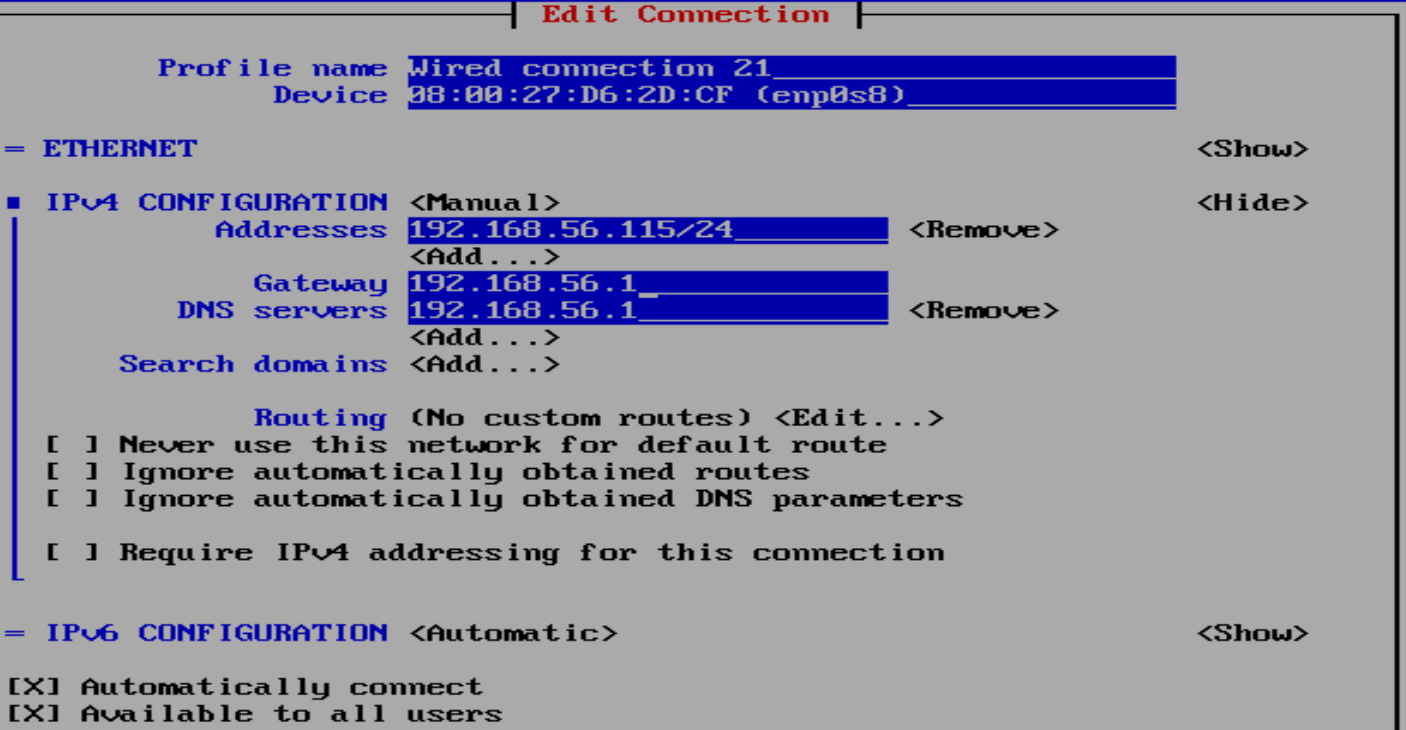
- 파일 확장자 보기 (활성화)

- “Hello World”

ㅇ CentOS의 네트워크 설정

### CentOS7 N/W 설정

* + 2개 I/F : NAT Network, VirtualBox Host Only Adapter
  + CentOS N/W 설정 명령
    - nmtui : 모두 수동으로 설치
    - [[Linux] CentOS 7 네트워크 설정 : 네이버 블로그 (naver.com)](https://m.blog.naver.com/tequini/220977723865)



| # reboot now  [root@localhost ~]# ifconfig  enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  **inet 10.0.2.55**  netmask 255.255.255.0 broadcast 10.0.2.255  inet6 fe80::a4ef:f26e:cb7:8e18 prefixlen 64 scopeid 0x20<link>  ether 08:00:27:08:e7:e9 txqueuelen 1000 (Ethernet)  RX packets 68 bytes 8816 (8.6 KiB)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 85 bytes 6728 (6.5 KiB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  **inet 192.168.56.115** netmask 255.255.255.0 broadcast 192.168.56.255  inet6 fe80::fc65:32b8:b20:51fc prefixlen 64 scopeid 0x20<link>  ether 08:00:27:63:0f:ff txqueuelen 1000 (Ethernet)  RX packets 61 bytes 7171 (7.0 KiB)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 62 bytes 7277 (7.1 KiB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 |
| --- |

### reboot now

192.168.56.115

----------------------------------------------------

ㅇ 리눅스 간단히 설명

- 배포판 : 레드햇 (RedHat, CentOS, Rocky...)

=> yum install httpd

데비안 (Ubuntu, Kali Linux, RPI OS)

=> apt (apt-get) install apache2

\* 서비스를 실행

# systemctl start/enable/status/stop/diable httpd

# firewall-cmd --permanent --add-port=80/tcp

# firewall-cmd --reload

[요구사항]

- 기능적 요구사항 : ID와 PW로 로그인.

- 품질적 제약사항 : 500명이 동시에 로그인을 시도해도 0.5초 이내 응답

- 보안적 요구사항 : ID를 고정하고 PW를 반복 입력하면 문제가 될 수 있으니, 방어해라.

\* 요구사항 점검 기준

- 완전성

- 명확성

- 구체적

\* 모듈 : 재사용 단위.

- 인터페이스 (약속)을 잘 지켜주는 것.

1) 결합도 : 2개 이상의 모듈간에는 느슨한 결합.

2) 응집도 : 하나의 모듈 내에는 필요한 기능만 모여 있어야 한다.

ㅇ V-Model : 테스트 레벨 (개발 단계별로 주요한 테스트 관점/기법)

1) 단위

2) 통합

3) 시스템

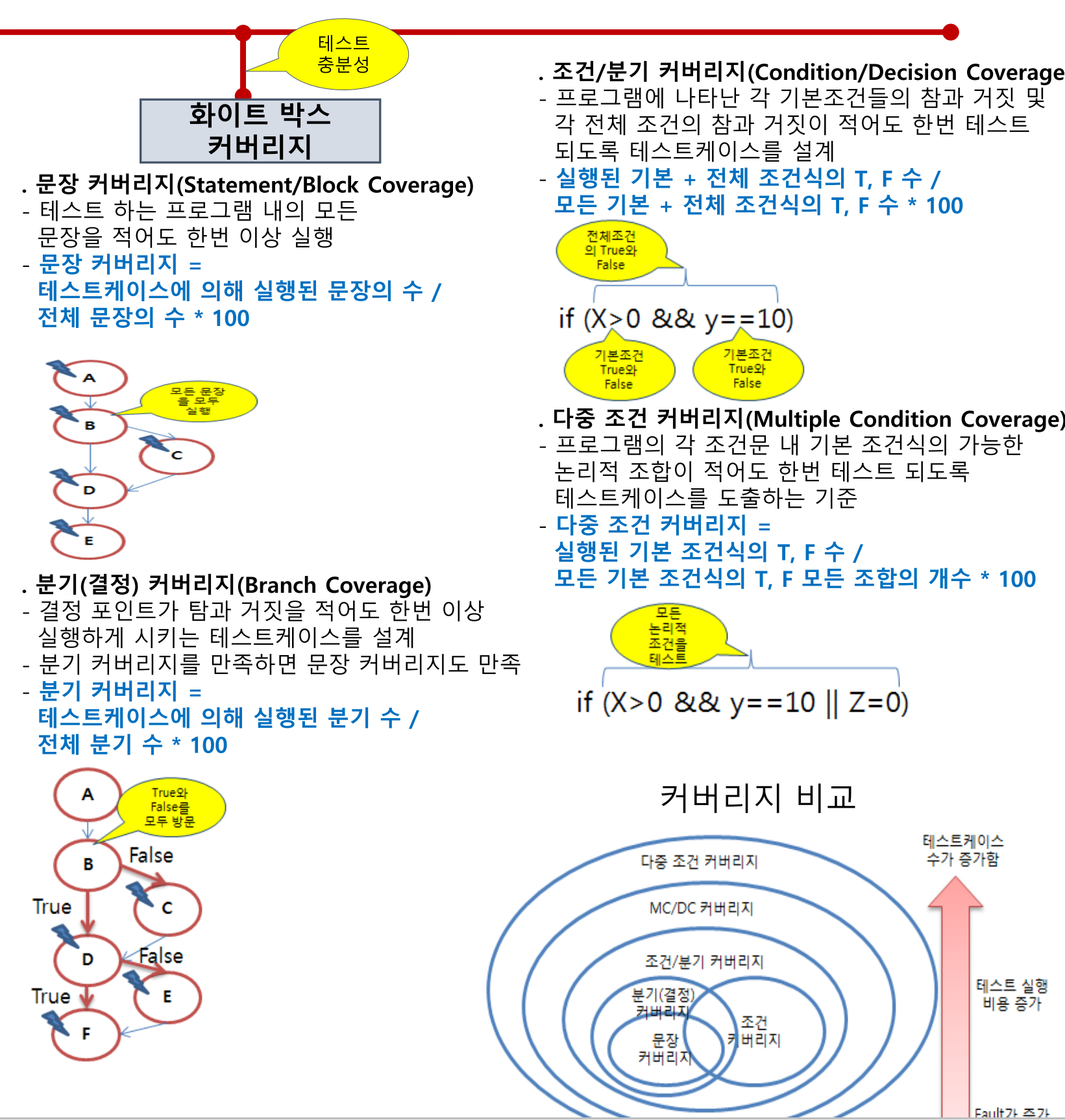
4) 인수

-----------------------

- Positive Test : 유효한 데이터 -> 유효한 형식, 유효한 구간

- Negative Test : 무효한 데이터 -> 무효한 형식, 무효한 구간.

- 경계값 분석 : 결과값이 변경되는 입력값 부근을 테스트 합니다..



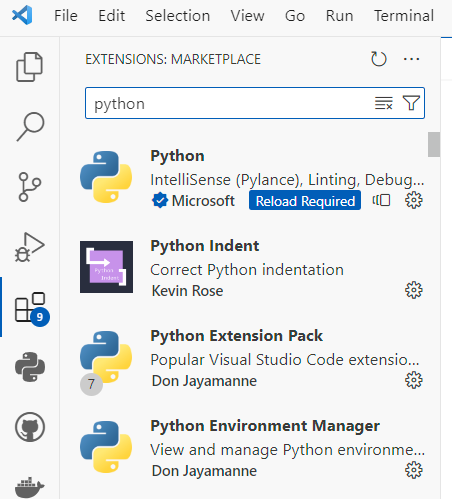
- 면허증의 속성을 갖는 자격증

1) 정보시스템 감리원

2) 감리사, 기술사

3) 개발보안진단원

**pip install virtualenv**

****

**\*** [**위키독스 (wikidocs.net)**](https://wikidocs.net/)

**(** [**https://wikidocs.net/**](https://wikidocs.net/) **)**

**\*** [**https://hey.technonia.com/**](https://hey.technonia.com/)

**\* 리눅스 종료 : shutdown -h now**

**[둘째날]**

\* [🧪 TDD 방법론 (테스트 주도 개발) - 알기 쉽게 정리 (tistory.com)](https://inpa.tistory.com/entry/QA-%F0%9F%93%9A-TDD-%EB%B0%A9%EB%B2%95%EB%A1%A0-%ED%85%8C%EC%8A%A4%ED%8A%B8-%EC%A3%BC%EB%8F%84-%EA%B0%9C%EB%B0%9C)

( <https://inpa.tistory.com/entry/QA-%F0%9F%93%9A-TDD-%EB%B0%A9%EB%B2%95%EB%A1%A0-%ED%85%8C%EC%8A%A4%ED%8A%B8-%EC%A3%BC%EB%8F%84-%EA%B0%9C%EB%B0%9C> )

\* [Web Service (devsmile.com)](http://lecture.devsmile.com/simpleweb/)

( <http://lecture.devsmile.com/simpleweb/> )

\* 준비 : simpleWebPython을 소스코드 구성 (가상환경 포함)

pip install -r requirements.txt

=> 10시 10분~ 시작하겠습니다.

------------

## <VirtualBox + CentOS7>

### VirtualBox 이미지 백업 내용 (OVA를 import하여 사용)

* + CentOS7\_CICD\_team1234\_v3.0.ova

| **[v3.0]**  [Docker images]  sonarsource/sonar-scanner-cli  node:lts-alpine  sonarqube  kchris000/simple-web-python  nginx  jenkins/jenkins:lts-jdk11  python:3.11.4  consol/tomcat-7.0  [Services]  docker (active running)  jenkins (active running on docker)  nginx (active running + jenkins proxy)  [Etc]  docker-compose  telnet  nano  ※ Working Folder : /var/studyDevOps , /var/studyDevOps/jenkins\_home  ※ 방화벽 : 80/tcp open |
| --- |

### CentOS7 N/W 설정

* + 2개 I/F : NAT Network, VirtualBox Host Only Adapter
  + CentOS N/W 설정 명령
    - nmtui : 모두 수동으로 설치
    - [[Linux] CentOS 7 네트워크 설정 : 네이버 블로그 (naver.com)](https://m.blog.naver.com/tequini/220977723865)

| [root@localhost ~]# ifconfig  enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  **inet 10.0.2.55**  netmask 255.255.255.0 broadcast 10.0.2.255  inet6 fe80::a4ef:f26e:cb7:8e18 prefixlen 64 scopeid 0x20<link>  ether 08:00:27:08:e7:e9 txqueuelen 1000 (Ethernet)  RX packets 68 bytes 8816 (8.6 KiB)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 85 bytes 6728 (6.5 KiB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  **inet 192.168.56.115** netmask 255.255.255.0 broadcast 192.168.56.255  inet6 fe80::fc65:32b8:b20:51fc prefixlen 64 scopeid 0x20<link>  ether 08:00:27:63:0f:ff txqueuelen 1000 (Ethernet)  RX packets 61 bytes 7171 (7.0 KiB)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 62 bytes 7277 (7.1 KiB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 |
| --- |

### Docker 설치 및 실습에 필요한 이미지 다운로드

* + 기존에 설치하 Docker가 있으면 먼저 삭제한 후 최신 버전의 Docker 설치
  + [Install Docker Engine on CentOS | Docker Documentation](https://docs.docker.com/engine/install/centos/)

| $ sudo yum remove docker \  docker-client \  docker-client-latest \  docker-common \  docker-latest \  docker-latest-logrotate \  docker-logrotate \  docker-engine  $ sudo yum install -y yum-utils  $ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo  $ sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin |
| --- |
| $ systemctl start docker  $ systemctl enable docker  $ systemctl status docker |
| $ docker pull consol/tomcat-7.0  $ docker pull node:lts-alpine  $ docker pull python:3.11.4  $ docker pull jenkins/jenkins:lts-jdk11  $ docker pull nginx  $ docker pull kchris000/simple-web-python:latest  $ docker pull sonarqube:latest  $ docker pull sonarsource/sonar-scanner-cli  --------  [root@localhost ~]# docker images  REPOSITORY TAG IMAGE ID CREATED SIZE  sonarsource/sonar-scanner-cli latest 74ae0c5a4cee 4 hours ago 358MB  node lts-alpine 50c7e33a9de1 2 days ago 176MB  sonarqube latest 806094f6f03a 4 days ago 702MB  kchris000/simple-web-python latest 7d297dd688f5 6 days ago 1.05GB  nginx latest 89da1fb6dcb9 2 weeks ago 187MB  jenkins/jenkins lts-jdk11 a40a8916af1d 2 weeks ago 471MB  python 3.11.4 a5fee9aa0e3a 3 weeks ago 1.01GB  consol/tomcat-7.0 latest 7c34bafd1150 8 years ago 601MB |

### 실습에 필요한 패키지 다운로드

* + nano, telnet 설치

| yum -y install nano telnet |
| --- |

* + nginx : Docker가 아닌 기본 설치 버전을 사용하기로 함. (단, 설치만 하고 실행하진 않음)

| nano /etc/yum.repos.d/nginx.repo  ---------  [nginx]  name=nginx.repo  baseurl=http://nginx.org/packages/centos/$releasever/$basearch/  gpgcheck=0  enabled=1  ---------  setsebool -P httpd\_can\_network\_connect on  yum install nginx |
| --- |

# mkdir /var/studyDevOps

(윈도우) scp simpleWebPython.zip [root@](mailto:root@centos.devsmile.com)192.168.56.115:/var/studyDevOps/

압축을 푸시고

# cd /var/studyDevOps/simpleWebPython

# source ./dockerBuild.sh

# docker run -rm -d -p 7070:7070 --name=simpleWebPython simple-web-python

1) docker : 도커 명령

2) run : 실행(처음으로 실행)

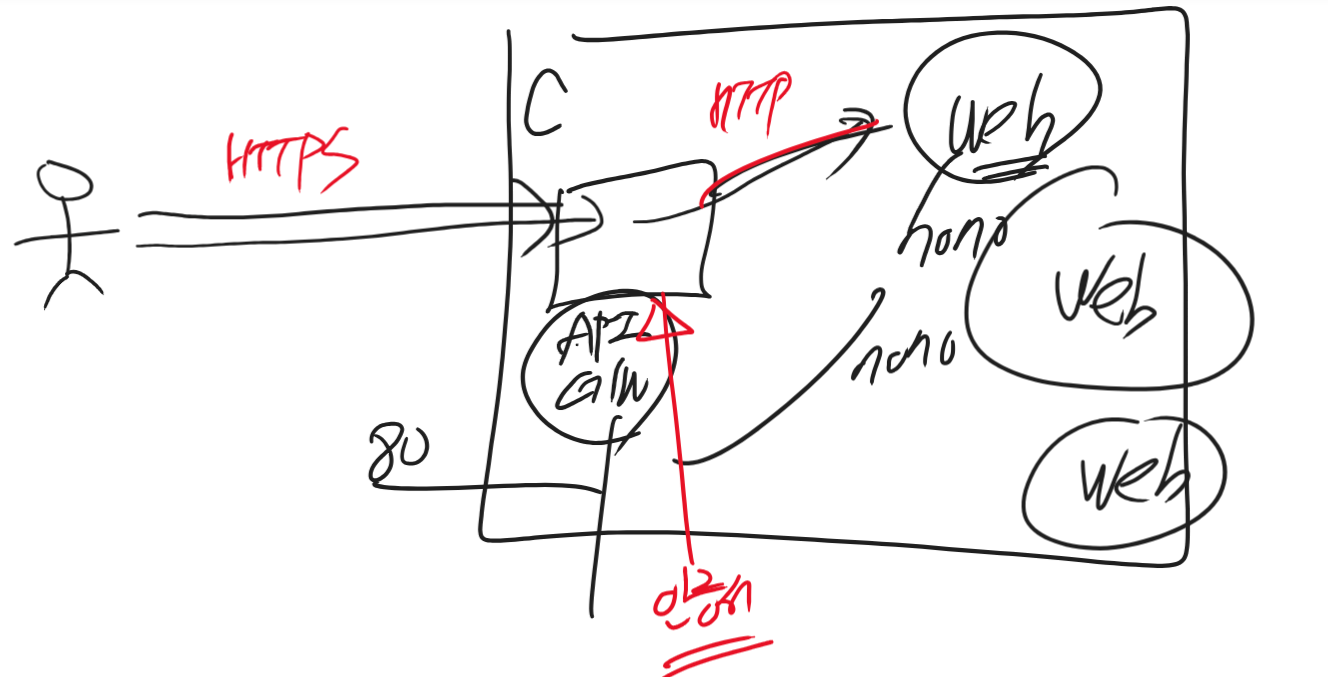
3) -rm : 종료시 인스턴스 정보를 삭제

4) -d : 백그라운드로 실행 (-it : 나한테 터미널을 다오)

5) -p 7070:7070 : (CentOS에서 7070으로 접속하면, Docker 7070으로 보내라)

6) --name=simpleWebPython : 실행한 서비스에 별칭

7) simple-web-python : 사용할 컨테이너 이름



# systemctl status nginx

### nginx 설정 파일 수정 및 적용

| # nano /etc/nginx/proxy\_params  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  proxy\_set\_header Cookie $http\_cookie;  proxy\_set\_header X-Session-Id $http\_x\_session\_id;  proxy\_pass\_request\_headers on;  proxy\_pass\_request\_body on; |
| --- |
| # nano /etc/nginx/conf.d/default.conf  nano default.conf    upstream simpleWebPython {  server localhost:7070;  keepalive 32;  }  server {  listen 80;  server\_name 192.168.56.115 localhost 127.0.0.1;  location /simpleweb/ {  include /etc/nginx/proxy\_params;  proxy\_pass http://simpleWebPython/;  sub\_filter 'href="/' 'href="/simpleweb/';  sub\_filter 'action="/' 'action="/simpleweb/';  sub\_filter\_once off;  proxy\_redirect / /simpleweb/;  }  location / {  root /usr/share/nginx/html;  index index.html index.htm;  }  error\_page 404 /;  error\_page 500 502 503 504 /;  } |
| # systemctl restart nginx |
| # curl http://192.168.56.115/simpleweb/ (웹 브라우저로 접속) |

----------------------------------------

\* [unittest vs pytest | 방성범 블로그 (bangseongbeom.com)](https://www.bangseongbeom.com/unittest-vs-pytest.html)

\* [PyTest 튜토리얼: 정의, 설치 방법, 프레임워크, 어설션 (guru99.com)](https://www.guru99.com/ko/pytest-tutorial.html)

( <https://www.guru99.com/ko/pytest-tutorial.html> )

\* [Python 자동 테스트를 위한 Pytest 사용법과 예제 (tistory.com)](https://kibua20.tistory.com/227)

( <https://kibua20.tistory.com/227> )

\* [파이썬으로 TDD 진행해보기(feat. unittest 예제) :: 뇌님의 관심사 (tistory.com)](https://brain-nim.tistory.com/66)

( <https://brain-nim.tistory.com/66> )

[기능]

\* 우리가 운영하는 서비스를 모니터링해서, 문제가 있으면 Slack으로 메시지를 전송하는

기능을 구현하고 싶다.

1) Slack Message 전송하는 기능

2) 사이트를 모니터링 기능

- 만일 응답이 없으면, 1분에 한씩 점검하고

5회 실패하면 최종 응답없음(문제로) 한다.

- 실패한 상황이 발생하면, 1분에 1번씩 점검해서 정상여부를 계속 확인한다.

3) 설정 파일을 관리

4) 10분에 1번씩 점검한다.

=> 반복.

\* requirements.txt

----------

certifi==2023.7.22

charset-normalizer==3.2.0

colorama==0.4.6

idna==3.4

iniconfig==2.0.0

packaging==23.1

pluggy==1.2.0

pytest==7.4.0

requests==2.31.0

schedule==1.2.0

urllib3==2.0.4

-----------------

서브폴더 : \_\_init\_\_.py

\* 슬랙채널 : [Slack Webhook API 생성하기 (tistory.com)](https://jojoldu.tistory.com/552)

ㅇ test\_example.py

------------

import unittest

class TestExample(unittest.TestCase):

def setUp(self):

print("Testing example", flush=True)

self.msg = "Hello Test"

def test\_msg\_is\_hello\_test(self):

msg = self.msg

self.assertEqual(msg, "Hello Test")

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

-----------

ㅇ test\_slackMsg.py

------------------

SLACK\_CHANNEL = 'https://hooks.slack.com/services/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'

def test\_slackMsg\_with\_invalidChannel\_Failed(self):

self.assertEqual(slackMsg.post\_slack\_channel("Test", "http://invalid\_url"),"FAILED")

def test\_slackMsg\_with\_validChannel\_OK(self):

-------------------

ㅇ slackMsg.py

-------------------

# Slack Channle에 메시지 보내기

# ref : https://jojoldu.tistory.com/552

import json

import urllib.request

def post\_slack\_channel(argStr, channelUrl):

message = argStr

send\_data = {

"text": message,

}

send\_text = json.dumps(send\_data)

request = urllib.request.Request(

channelUrl,

data=send\_text.encode('utf-8'),

)

try:

with urllib.request.urlopen(request) as response:

slack\_message = response.read()

return slack\_message

except Exception as e:

print(f'slack error : {e}', flush=True)

return 'FAILED'

------------------------------

ㅇ test\_healthCheck.py

----------------------

def test\_healthCheck\_with\_invalidUrl\_Failed(self):

---------------------

ㅇ monitor.py

-----------

import requests

def HealthCheck(url):

try:

response = requests.get(url)

if response.status\_code == 200:

return True

else:

return False

except:

return False

-------------

==> 4시 15분~ 다시 시작하겠습니다.

* [Clean Code 클린 코드 - 예스24 (yes24.com)](https://www.yes24.com/Product/Goods/11681152)
* [Yooii-Studios/Clean-Code: Clean Code](https://github.com/Yooii-Studios/Clean-Code/tree/master)

# [파이선 자동화 - 도구/활용 등]

### Codetorial (Good~) : [Codetorial](https://codetorial.net/index.html)

* + <https://codetorial.net/index.html>
  + 파이썬 기본 튜토리얼, Numpy Tutorial, Matplotlib Tutorial, PyQT5 Tutorial (GUI 개발)  
    BeautifulSoup Tutorial, xlrd/xllwt Tutorial (엑셀), Pillow tutorial (이미지 편집 등)  
    Googletrans Tuorial (구글 번역 API), PyWin32 Tutorial (Win32 API / 윈도우 기능 활용)
  + PyAutoGUI (시스템 조작), Pyperclip Tutorial (클립보드)
  + TensorFlow Tutorial (머신러닝)
  + Tips & Examples

### 사장님 몰래 하는 파이썬 업무자동화 : [들키면 일 많아짐](https://wikidocs.net/book/6353)

* + <https://wikidocs.net/book/6353>
  + 엑셀, 워드, 이미지 컨텐트 작성
  + PyAutoGUI (시스템 조작), 파일/경로 다루기, 클롤링 등