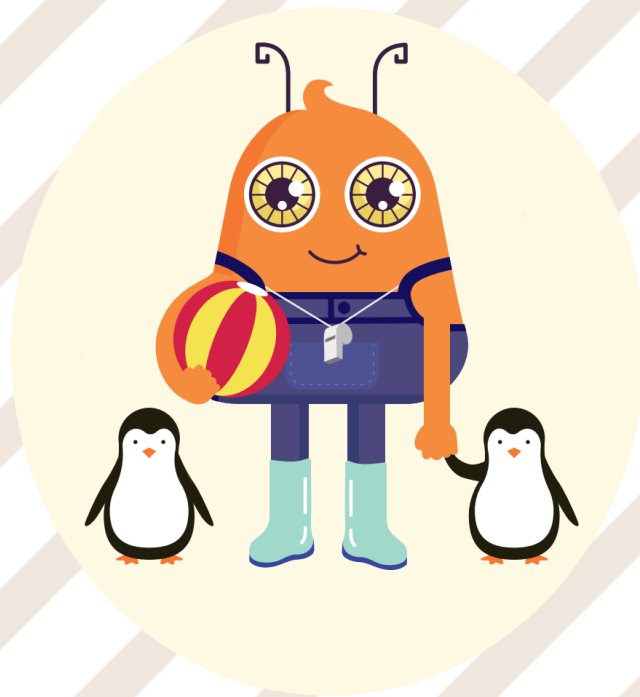


2일차

Linux

웹 서버, WAS

이희진



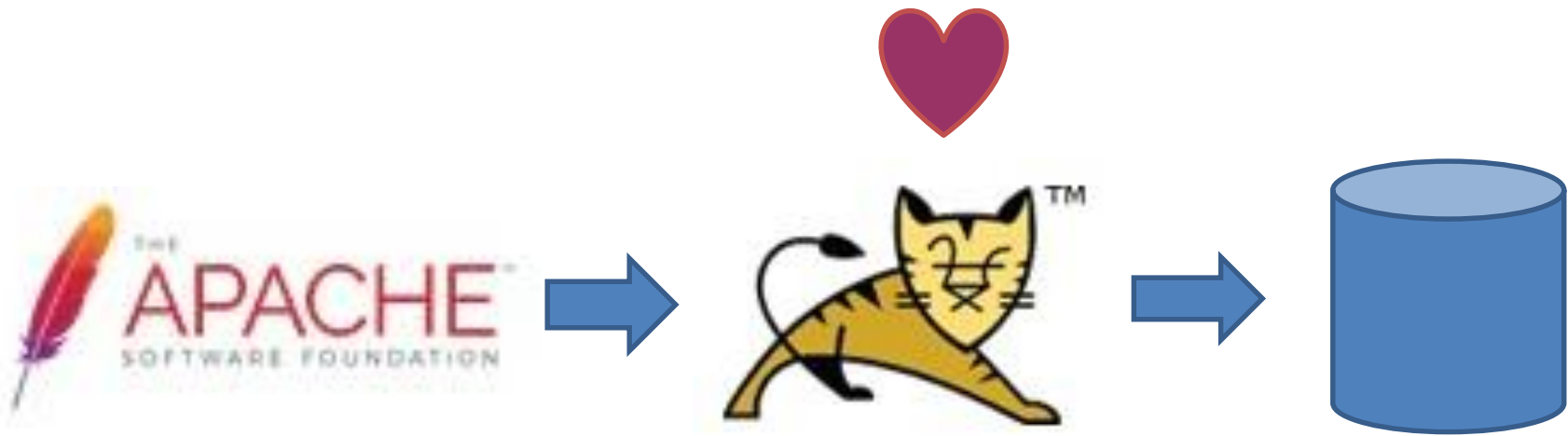
C.ontents

01 Webserver, WAS 설치와 운영

학습목표

- 웹 서버, WAS 를 설치한다.

3 tier architecture (WEB-WAS-DB)



1. Web server와 WAS

■ Web Server의 역할

■ Static resource

- Web server에서 static resource를 다루고, WAS 서버에서 dynamic 정보를 다룬다.
- DB까지 가지 않는 정보 전달. 그림파일, text 등

■ Security

- 방화벽 바깥에 webserver를 위치하고 WAS, DB서버는 방화벽 안쪽에 위치
- SSL 처리

■ Load balancing

- 하나의 Webserver가 여러 개의 WAS 서버에게 업무 분배

■ Web server의 장점

- Static resource 처리를 빨리 해주고,
- WAS 서버의 주소 노출이 안되어 보안이 강화되고 (port번호를 숨길수 있음)
- 여러 개의 WAS의 업무 분배해 주어 자원을 효율적으로 활용

[실습 1] apt-get 명령어로 웹 서버 설치하기

1. apt-get 명령어로 Apache 설치하기

```
$apt-get update  
$apt-get install apache2
```

2. 설치 후 버전 확인

```
$apache2 -v
```

3. Apache 서비스 기동/멈춤

```
$ service apache2 start  
$ service apache2 stop  
$ service apache2 restart
```

[실습 1] apt-get 명령어로 웹 서버 설치하기

4. Apache 서비스 가동/멈춤/상태 확인

```
$ systemctl status apache2
$ systemctl restart apache2
$ systemctl start apache2
$ systemctl stop apache2
```

Q를 누르면 프롬프트가 다시 나타남

```
heejinlee@ubuntu:/etc/apache2$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Tue 2020-06-16 10:03:09 KST; 2min 25s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 6225 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
 Main PID: 6244 (apache2)
    Tasks: 55 (limit: 2285)
   Memory: 5.6M
    CGroup: /system.slice/apache2.service
            └─6244 /usr/sbin/apache2 -k start
              └─6245 /usr/sbin/apache2 -k start
                └─6246 /usr/sbin/apache2 -k start

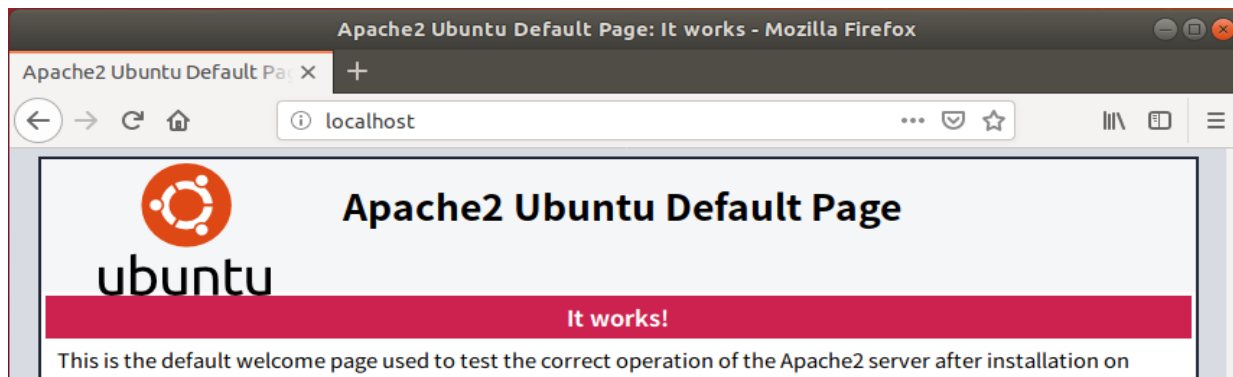
Jun 16 10:03:09 ubuntu systemd[1]: Starting The Apache HTTP Server...
Jun 16 10:03:09 ubuntu apachectl[6243]: AH00558: apache2: Could not reliably de>
Jun 16 10:03:09 ubuntu systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

[실습 1] apt-get 명령어로 웹 서버 설치하기

5. 웹 서버 동작 확인하기

5-1 파이어폭스 실행, 주소 창에 **http://localhost/** 또는 **http://127.0.0.1/** 입력

→ 다음과 같은 창이 나타나면 apache2 서비스가 정상적으로 작동하는 것



5-2 서버 동작 확인, listen port 80을 확인 하여 정상 동작 확인

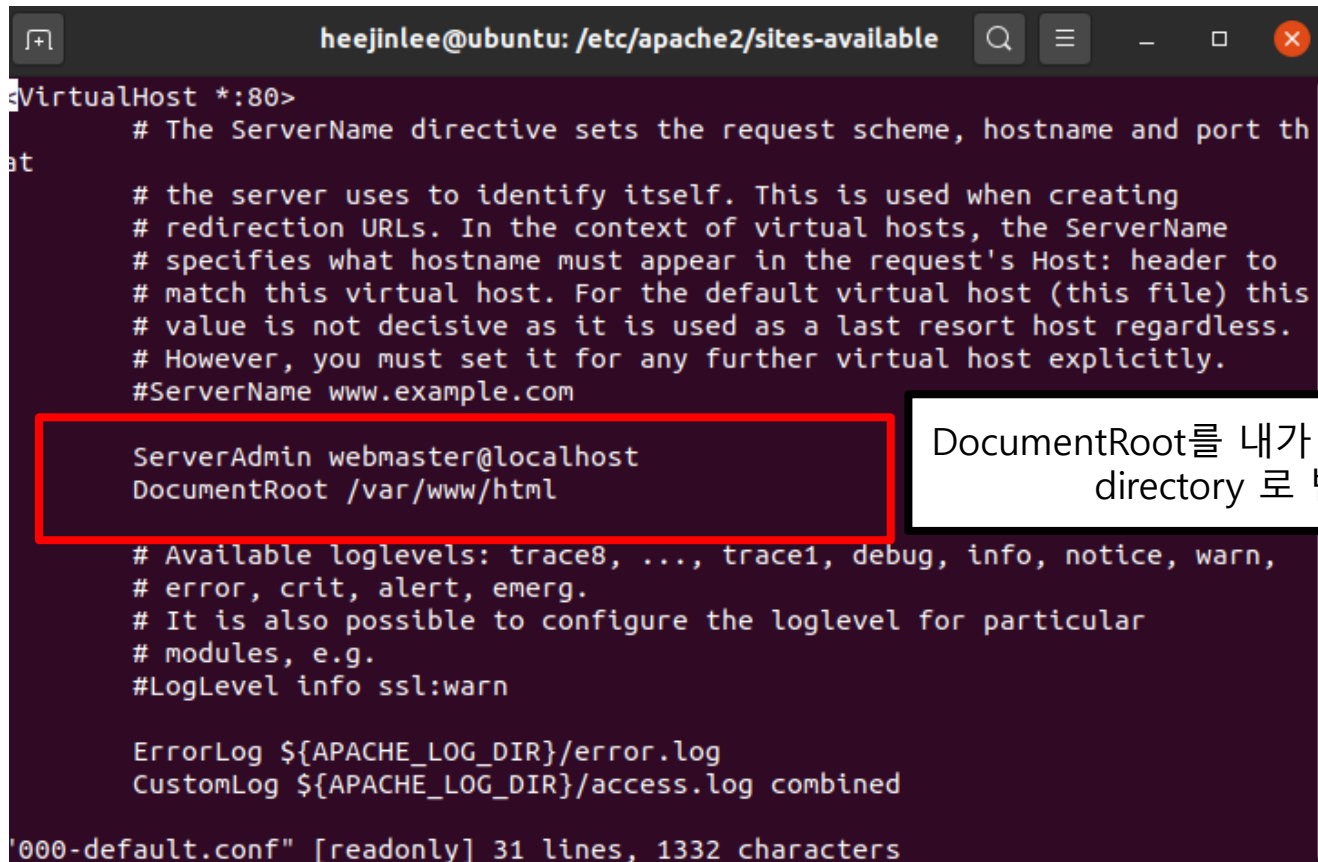
```
$netstat -ntlp
```

```
heejinlee@ubuntu:/etc/apache2$ netstat -ntlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::1:631                 :::*                    LISTEN      -
heejinlee@ubuntu:/etc/apache2$
```


[실습 2] 내 홈페이지 보여주기

- Apache server의 home directory 변경
 - 내 홈페이지 root 로 변경해야 함.
 - 설정을 변경했으니, apache2를 재기동 한다.

```
$vi /etc/apache2/sites-available/000-default.conf
```



```
VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port th
at
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

'000-default.conf" [readonly] 31 lines, 1332 characters
```

DocumentRoot를 내가 원하는 home
directory 로 변경

[실습 2] 내 홈페이지 보여주기

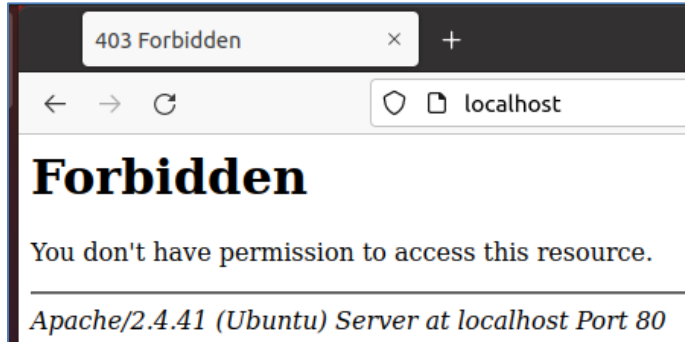
- 새로운 DocumentRoot에 모든 사용자에게 read, write, execute 권한을 준다.
- 만약 DocumentRoot를 /home/사용자id/html로 하였다면, 모든 경로에 일반사용자에게 read, write, execute 권한을 준다.

```
$chmod 777 /home/사용자id  
$chmod 777 /home/사용자id/html  
$chmod 777 /home/사용자id/html/index.html
```

```
heejin@heejin-virtual-machine:~$ chmod 777 /home/heejin/html  
heejin@heejin-virtual-machine:~$ chmod 777 /home/heejin/html/index.html  
heejin@heejin-virtual-machine:~$ ls -al /home/heejin/html/index.html  
-rwxrwxrwx 1 heejin heejin 37 3월 26 23:18 /home/heejin/html/index.html  
heejin@heejin-virtual-machine:~$
```

[실습 2] 내 홈페이지 보여주기

- DocumentRoot 를 바꾸려고 하면 Permission 에러가 난다.



- DocumentRoot를 바꿨는데, 예전 페이지가 나온다면?
 - 문제가 있다.
 - Apache2 재기동 - 설정을 변경하면, 반영하기 위하여 재기동 필수
 - Cache 지우기
 - 인내심 탑재
 - 무한 재시도
- 아무것도 안나온다?
 - https:// 는 안된다. http:// 로 변경하자.

[실습 2] 내 홈페이지 보여주기

- /etc/apache2/apache2.conf에서 <Directory>에 관한 permission 설정변경
 - denied -> granted 로 변경 (line 162)
 - 설정을 변경했으니, apache2를 재기동 한다.
- 정상 동작 확인

```
# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>
```

denied -> granted

[실습 2] 내 홈페이지 보여주기

- Apache error 확인
 - /var/log/apache2/error.log
 - tail error.log

```
heejin@heejin-registry:/var/log/apache2$ tail error.log
[Sun Mar 26 23:48:34.283579 2023] [core:error] [pid 7282:tid 140510484211264] (13)Permission denied: [client 127.0.0.1:56556] AH00035: access to /favicon.ico denied (filesystem path '/home/heejin/html') because search permissions are missing on a component of the path, referer: http://localhost/
[Sun Mar 26 23:48:43.723525 2023] [core:error] [pid 7283:tid 140510425462336] (13)Permission denied: [client 127.0.0.1:47862] AH00035: access to / denied (filesystem path '/home/heejin/html') because search permissions are missing on a component of the path
[Sun Mar 26 23:49:17.049406 2023] [core:error] [pid 7283:tid 140510450640448] (13)Permission denied: [client 127.0.0.1:41440] AH00035: access to / denied (filesystem path '/home/heejin/html') because search permissions are missing on a component of the path
[Sun Mar 26 23:49:17.213431 2023] [core:error] [pid 7283:tid 140510459033152] (13)Permission denied: [client 127.0.0.1:41440] AH00035: access to /favicon.ico denied (filesystem path '/home/heejin/html') because search permissions are missing on a component of the path, referer: http://localhost/
[Sun Mar 26 23:49:24.322777 2023] [core:error] [pid 7282:tid 140510459033152] (13)Permission denied: [client 127.0.0.1:41454] AH00035: access to / denied (filesystem path '/home/heejin/html') because search permissions are missing on a component of the path
[Sun Mar 26 23:49:34.795383 2023] [mpm_event:notice] [pid 7281:tid 140510656190336] AH00492: caught SIGWINCH, shutting down gracefully
[Sun Mar 26 23:49:34.908880 2023] [mpm_event:notice] [pid 7721:tid 140506482358144] AH00489: Apache/2.4.52 (Ubuntu) configured -- resuming normal operations
[Sun Mar 26 23:49:34.909130 2023] [core:notice] [pid 7721:tid 140506482358144] AH00094: Command line: '/usr/sbin/apache2'
[Sun Mar 26 23:49:50.644798 2023] [core:error] [pid 7723:tid 140506469340736] (13)Permission denied: [client 127.0.0.1:37328] AH00035: access to / denied (filesystem path '/home/heejin/html') because search permissions are missing on a component of the path
[Sun Mar 26 23:49:50.930220 2023] [core:error] [pid 7723:tid 140506460948032] (13)Permission denied: [client 127.0.0.1:37328] AH00035: access to /favicon.ico denied (filesystem path '/home/heejin/html') because search permissions are missing on a component of the path, referer: http://localhost/
heejin@heejin-registry:/var/log/apache2$ pwd
/var/log/apache2
```

[실습 2] 내 홈페이지 보여주기

방화벽 해제

외부에서 웹 서버에 접근할 수 있도록 **ufw allow 80** 명령으로 포트 허용

ifconfig ens32 명령으로 가상머신의 IP 주소 확인

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# ufw allow 80
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
root@server:~# ifconfig ens32
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.111.100 netmask 255.255.255.0 broadcast 192.168.111.255
    inet6 fe80::c8c6:fb2e:16d8:552e prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8b:32:6a txqueuelen 1000 (Ethernet)
    RX packets 24810 bytes 36212455 (36.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3702 bytes 299412 (299.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

[실습 3] 다른 홈페이지 보여주기

- 1. DocumentRoot 아래에 다른 디렉토리 만들기. (project)
 - 2. 그 안에 a.html만들기.
 - 내용은 간단한 text로 씀
 - 3. 이걸 실행하려면 Url에 어떻게 넣어야 하는가? 실행해보자
-
- index.html의 마법?
 - 경로까지만 쓰면, 기본으로 보여줌

[실습 4] WAS 설치 (Tomcat)

■ Tomcat10 설치

Tomcat 설치

```
$apt-get install tomcat10*
```

```
ubuntu@ip-172-31-29-252:~$ sudo apt-get install tomcat
tomcat-jakartaee-migration tomcat10 tomcat10-admin
tomcat10-docs tomcat10-examples tomcat10-user
```

Tomcat 실행/중지

```
$service tomcat10 start
$service tomcat10 stop
$service tomcat10 status
```

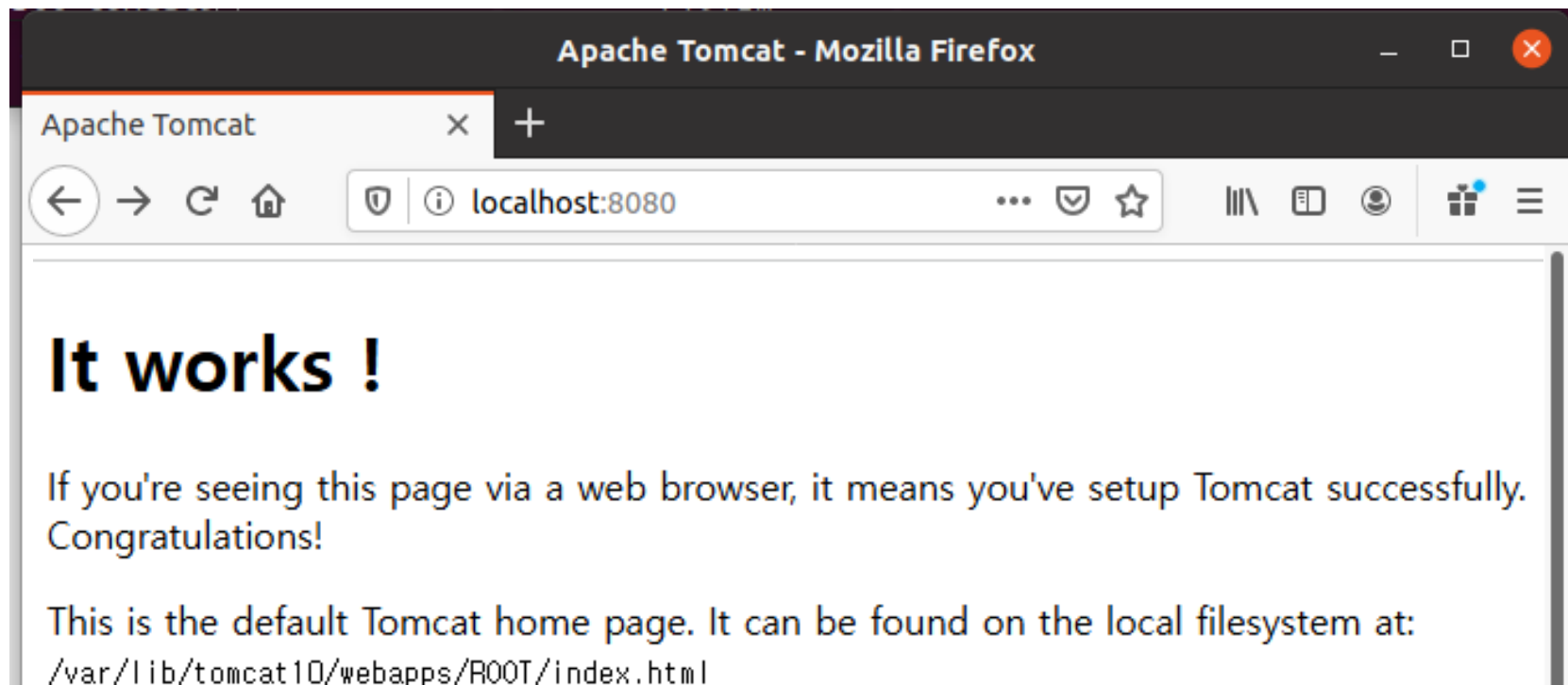
Tomcat 실행/중지

```
$systemctl start tomcat10
$systemctl stop tomcat10
$systemctl status tomcat10
$systemctl restart tomcat10
```


[실습 4] WAS 설치 (Tomcat)

■ 설치 확인

- <http://localhost:8080> (8080은 tomcat에서 정한 port)
- Tomcat 기본 홈페이지 경로 - `/var/lib/tomcat10/webapps/ROOT/index.html`



[실습 4] WAS 설치 (Tomcat)

■ Example을 확인해 보자

Servlet Examples

localhost:8080/examples/servlets/

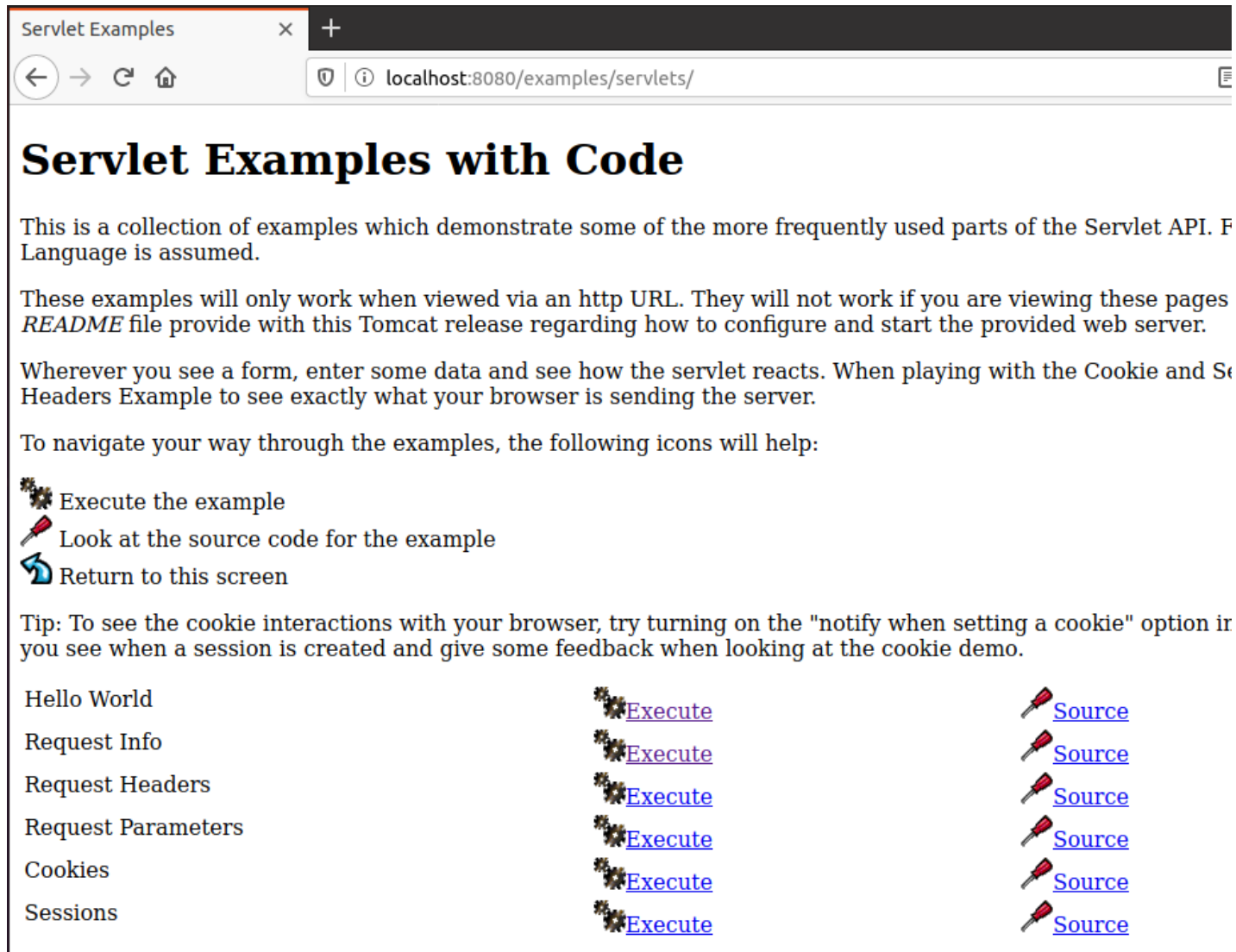


Servlet Examples with Code

This is a collection of examples which demonstrate some of the more frequently used parts of the Servlet API. For simplicity, Java 5.0 Language is assumed.













These examples will only work when viewed via an http URL. They will not work if you are viewing these pages from a file. The *README* file provide with this Tomcat release regarding how to configure and start the provided web server.

Wherever you see a form, enter some data and see how the servlet reacts. When playing with the Cookie and Session Examples, use the Headers Example to see exactly what your browser is sending the server.

To navigate your way through the examples, the following icons will help:

-  Execute the example
-  Look at the source code for the example
-  Return to this screen

Tip: To see the cookie interactions with your browser, try turning on the "notify when setting a cookie" option in your browser. You will see when a session is created and give some feedback when looking at the cookie demo.

Hello World	 Execute	 Source
Request Info	 Execute	 Source
Request Headers	 Execute	 Source
Request Parameters	 Execute	 Source
Cookies	 Execute	 Source
Sessions	 Execute	 Source

[실습 4] WAS 설치 (Tomcat)

- Tomcat-admin – GUI 관리도구
 - Tomcat에 app을 deploy, 설정 변경
 - 주소: Localhost:8080/manager
 - manager 페이지를 사용하기 위한 권한 부여
 - /var/lib/tomcat10/conf/tomcat-users.xml 에 manager-gui role 추가

```
<role rolename="manager-gui"/>  
<user username="myid" password="mypasswd " roles="manager-gui"/>
```

Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy

[실습 4] WAS 설치 (Tomcat)

- App을 tomcat에서 deploy하는 방법
 - Tomcat admin 을 사용하여 deploy
 - 직접 app을 폴더에 넣어서 deploy
 - /var/lib/tomcat10/webapps/ 내에 war 파일 위치시킴
 - Tomcat이 압축을 해제하여 directory 형태로 변경함
- WAR(Web Application aRchive)
 - Web application을 묶는 확장자
 - 애플리케이션 실행을 위한 컴파일된 모든 클래스 파일, 설정 파일들이 모두 포함
 - 웹 어플리케이션 설정을 정의한 배포 명세서 (web.xml)이 존재
- Sample application을 Deploy해보고 확인해 본다.
 - <https://tomcat.apache.org/tomcat-10.0-doc/appdev/sample/>

[실습 4] WAS 설치 (Tomcat)

■ Deploy된 application확인

```
# 사용자가 deploy한 application 들을 볼수 있다.  
# /var/lib/tomcat10/webapps
```

```
root@ubuntu:/var/lib/tomcat9/webapps# ls -al  
total 24  
drwxrwxr-x 4 tomcat tomcat 4096 Mar 17 14:28 .  
drwxr-xr-x 5 root    root    4096 Mar 17 13:23 ..  
drwxr-xr-x 3 root    root    4096 Mar 16 11:05 ROOT  
drwxr-x--- 5 tomcat tomcat 4096 Mar 17 14:28 sample  
-rw-r----- 1 tomcat tomcat 4606 Mar 17 14:28 sample.war
```

```
# 사용자가 deploy한 application뿐만 아니라, tomcat에 설치되어  
있는 기본 application도 볼 수 있다.  
#/var/lib/tomcat10/work/Catalina/localhost
```

```
root@ubuntu:/var/lib/tomcat9/work/Catalina/localhost# ls  
docs  examples  host-manager  manager  ROOT  sample
```

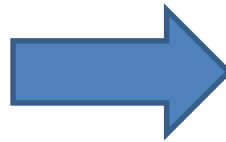
[실습 4] WAS 설치 (Tomcat)

- Tomcat 서버의 설정
 - /var/lib/tomcat10/conf/server.xml
 - Tomcat 접속 port를 확인해보자.
 - Connector? Port? Protocol?
- Line 69 을 살펴 볼 것

Q1. Tomcat으로 homepage를 구동하려는데,
8080 말고 다른 port로 접속할 수 있나요?

Q2. Tomcat이랑 Apache랑 동시에 다른 홈페이지
구동할 수 있나요?

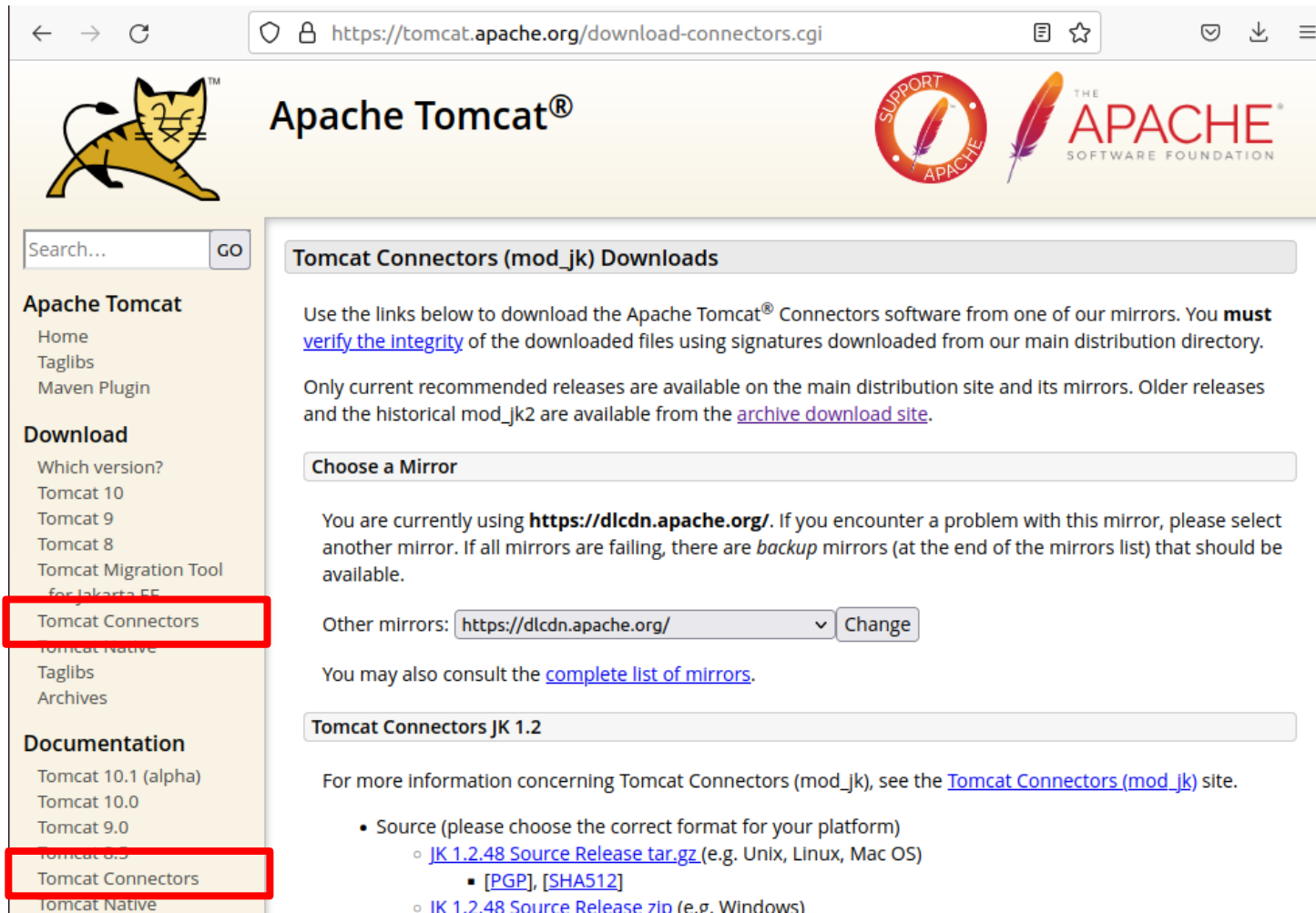
[실습 5] Web server와 WAS서버 연결



[실습 5] Web server와 WAS서버 연결

- Apache와 Tomcat을 연결하는 connector 모듈이 필요

```
sudo apt-get install libapache2-mod-jk*
```



Apache Tomcat®

Tomcat Connectors (mod_jk) Downloads

Use the links below to download the Apache Tomcat® Connectors software from one of our mirrors. You **must** [verify the integrity](#) of the downloaded files using signatures downloaded from our main distribution directory.

Only current recommended releases are available on the main distribution site and its mirrors. Older releases and the historical mod_jk2 are available from the [archive download site](#).

Choose a Mirror

You are currently using **https://dlcdn.apache.org/**. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors:

You may also consult the [complete list of mirrors](#).

Tomcat Connectors JK 1.2

For more information concerning Tomcat Connectors (mod_jk), see the [Tomcat Connectors \(mod_jk\)](#) site.

- Source (please choose the correct format for your platform)
 - [JK 1.2.48 Source Release tar.gz](#) (e.g. Unix, Linux, Mac OS)
 - [\[PGP\]](#), [\[SHA512\]](#)
 - [JK 1.2.48 Source Release zip](#) (e.g. Windows)

[실습 5] Web server와 WAS서버 연결



- 1. apache-tomcat connector에 tomcat들의 별명을 지어주고, 정보를 설정
 - Connector 설정 파일 - /etc/libapache2-mod-jk/workers.properties
 - Tomcat의 home directory, port, 통신 protocol를 설정
- 2. webserver apache에 apache-tomcat connector에 설정된 tomcat 으로 요청을 보내도록 설정 (별명 사용)
 - /etc/apache2/sites-available/000-default.conf
 - /etc/apache2/mods-available/jk.conf
 - /etc/apache2/mods-available/jk.load
- 3. WAS tomcat에 apache에서 오는 요청을 받으라고 설정
 - 특정 protocol과 특정 port로 오는 요청을 받으라고 설정
 - AJP1.3으로 연결되고 port 8009로 오는 요청을 받으라고 설정
 - Tomcat 설정 파일 - /var/lib/tomcat10/conf/server.xml
 - Deploy할 app 위치 - /var/lib/tomcat10/webapps

[실습 5] Web server와 WAS서버 연결

- 1. apache-tomcat connector에 tomcat들의 별명을 지어주고, 정보를 설정
 - /etc/libapache2-mod-jk/workers.properties (설치시 자동 생성됨)
 - tomcat10 home과 java home 을 변경 – 본인 pc에서 맞추어 설정, 해당 경로가 존재하는지 확인
 - Load balance 관련 설정 삭제



```
# /etc/libapache2-mod-jk/workers.properties
```

```
workers.tomcat_home=/usr/share/tomcat10
```

```
workers.java_home=/usr/lib/jvm/default-java
```

```
worker.list=ajp13_worker
```

```
worker.ajp13_worker.port=8009
```

```
worker.ajp13_worker.host=localhost
```

```
worker.ajp13_worker.type=ajp13
```

```
#worker.ajp13_worker.lbfactor=1
```

```
#worker.loadbalancer.type=lb
```

```
#worker.loadbalancer.balance_workers=ajp13_worker
```

****설정을 변경하면
재기동!!! **
Apache?tomcat?**

[실습 5] Web server와 WAS서버 연결

- 2. webserver apache에 apache-tomcat connector에 설정된 tomcat 으로 요청을 보내도록 설정 (별명 사용)



```
#/etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80>
```

```
ServerAdmin webmaster@localhost
```

```
#DocumentRoot /var/www/html
```

```
JkMount /* ajp13_worker
```

```
</VirtualHost>
```

DocumentRoot 없애도 됨.
왜죠?

#JkMount url_pattern worker=>
url_pattern은 worker에게 보냄

```
# /etc/apache2/mods-available/jk.conf 확인
```

```
<IfModule jk_module>
```

```
JkWorkersFile /etc/libapache2-mod-jk/workers.properties
```

```
JkLogFile /var/log/apache2/mod_jk.log
```

```
JkLogLevel info
```

```
JkShmFile /var/log/apache2/jk-runtime-status
```

****설정을 변경하면
재기동!!! ****

```
# /etc/apache2/mods-available/jk.load 확인
```

```
LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so
```

[실습 5] Web server와 WAS서버 연결

- 3. WAS tomcat에 특정 port와 특정 protocol로 오는 요청을 받으라고 설정

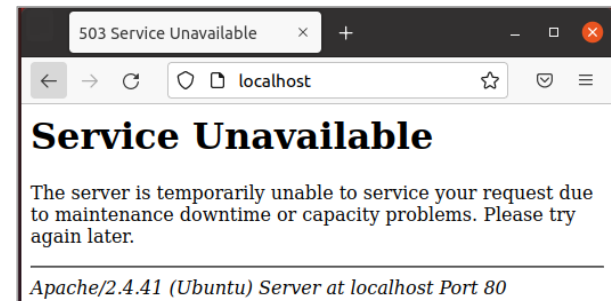


- `/var/lib/tomcat10/conf/server.xml` – 103 line 주석 해제

```
# /var/lib/tomcat10/conf/server.xml  
  
<Connector protocol="AJP/1.3"  
    address=":::1"  
    port="8009"  
    redirectPort="8443"  
    maxParameterCount="1000"  
/>
```

****설정을 변경하면
재기동!!! ****

- [Http://localhost](http://localhost) 로 동작 -> service unavailable error, 503 error
- 뭔가 오래 걸리면 잘 된 것임!!



[실습 5] Web server와 WAS서버 연결

- 3. WAS tomcat에 특정 port와 특정 protocol로 오는 요청을 받으라고 설정
 - /var/lib/tomcat10/conf/server.xml

```
# /var/lib/tomcat10/conf/server.xml
<Connector protocol="AJP/1.3"
    address="0.0.0.0"
    port="8009"
    redirectPort="8443"
    maxParameterCount="1000"
    packetsize="65536"
    secretRequired="false"
/>
```

- Localhost:8080으로 들어갔을때와 localhost로 들어간 경우
동일한 화면 tomcat 화면이 나옴을 확인
 - Apache와 tomcat이 정상 연결 ! 정상 동작!!

[실습 5] Web server와 WAS서버 연결

Q1. Apache와 Tomcat이 다른 server에 있는 경우는?

➔ Tomcat의 host를 나타내는 ip를 127.0.0.1(or localhost)에서 tomcat서버의 ip로 변경

Q2. Apache-Tomcat 연동을 했는데, http://localhost 하면 tomcat으로 연결이 안될때?

➔ **https://**가 아닌가 확인한다.

➔ Netstat -ntlp => **8009** 이 열려있는가? 확인한다.

➔ Tomcat 구동중인지 status를 확인한다.

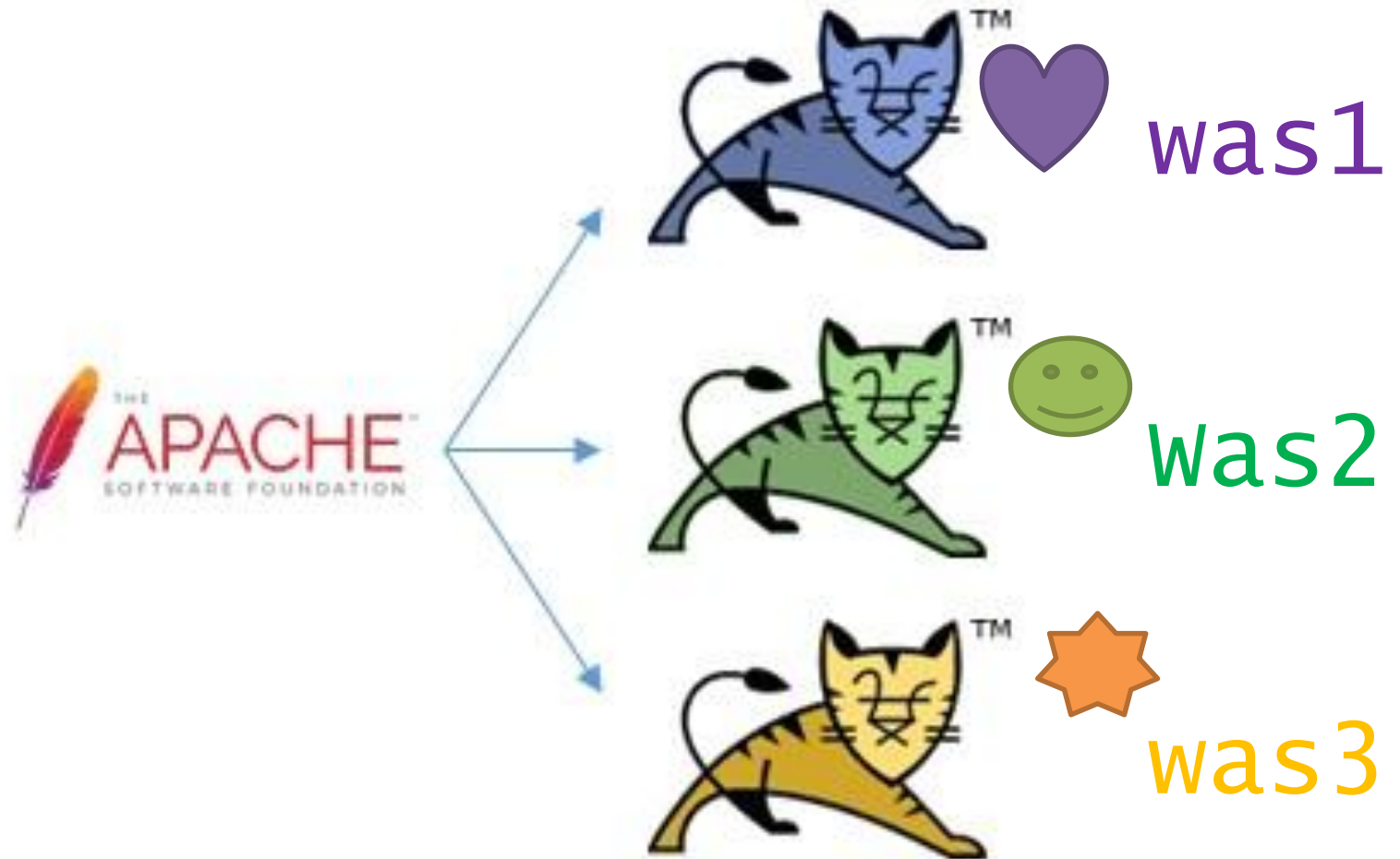
Q3. http://localhost:8009는 왜 안되죠?

➔

Q4. Apache의 DocumentRoot에 있는 index.html은 왜 동작 안하는 거죠?

➔

[실습 6] was가 여러 개인데, 각각 다른 app이 있는경우



[실습 6] was가 여러 개인데, 각각 다른 app이 있는경우

■ 1. apache-tomcat connector 설정

- /etc/libapache2-mod-jk/workers.properties (설치시 자동 생성됨)
- tomcat10 home과 java home 을 변경 – 본인 pc에서 맞추어 설정, 해당 경로가 존재하는지 확인

```
#/etc/libapache2-mod-jk/workers.properties
```

```
workers.tomcat_home=/usr/share/tomcat10
```

```
workers.java_home=/usr/lib/jvm/default-java
```

```
worker.list=was1,was2,was3
```

```
worker.was1.port=port1(해당 port)
```

```
worker.was1.host=ip1(해당 ip)
```

```
worker.was1.type=ajp13
```

```
worker.was2.port=port2(해당 port)
```

```
worker.was2.host=ip2 (해당 ip)
```

```
worker.was2.type=ajp13
```

```
...
```


[실습 6] was가 여러 개인데, 각각 다른 app이 있는경우

worker.list=was1,was2,was3

```
#/etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80>
```

```
    ServerName 127.0.0.1
```

```
    ServerAdmin webmaster@localhost
```

```
    DocumentRoot /var/www/html
```

```
    JkMount /app1/* was1
```

```
    JkMount /app2/* was2
```

```
    JkMount /app3/* was3
```

```
</VirtualHost>
```

JkMount url_pattern worker

=> url_pattern은 worker에게 보냄

=> * 중요- 모든 request를 해당 was로 보냄

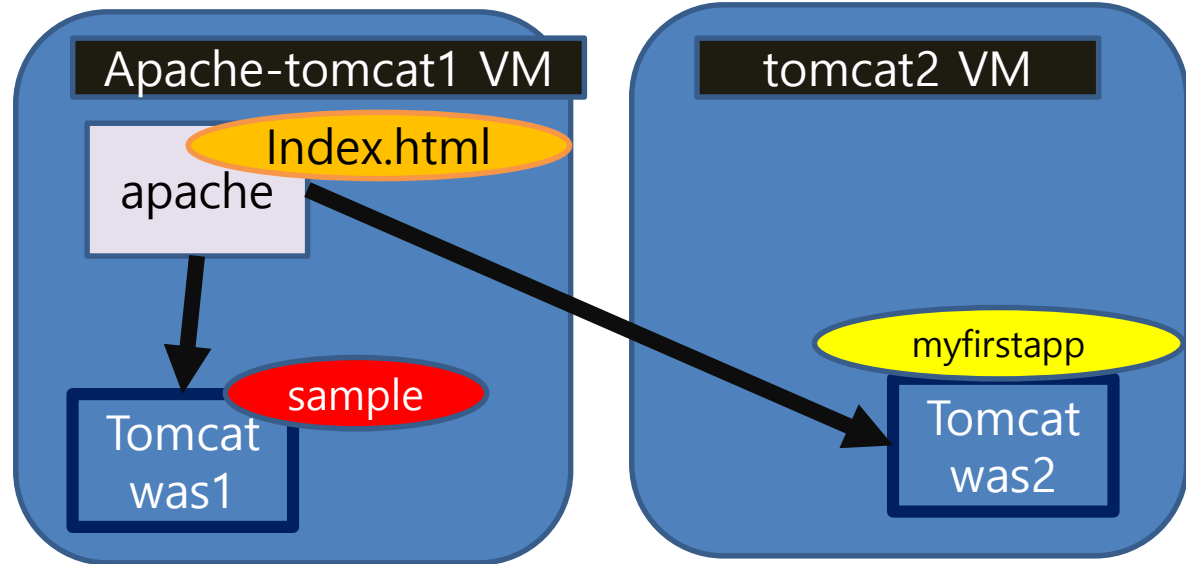
DocumentRoot가 필요하다. 이유는?

- app1, app2, app3 이외의 것들은 DocumentRoot로 진행
- 안쓰면 기본값으로 /var/www/html로 default 설정

[실습 6] was가 여러 개인데, 각각 다른 app이 있는경우

192.168.128.001

192.168.128.002



JkMount /sample/* was1
JkMount /myfirstapp/* was2

<http://localhost>

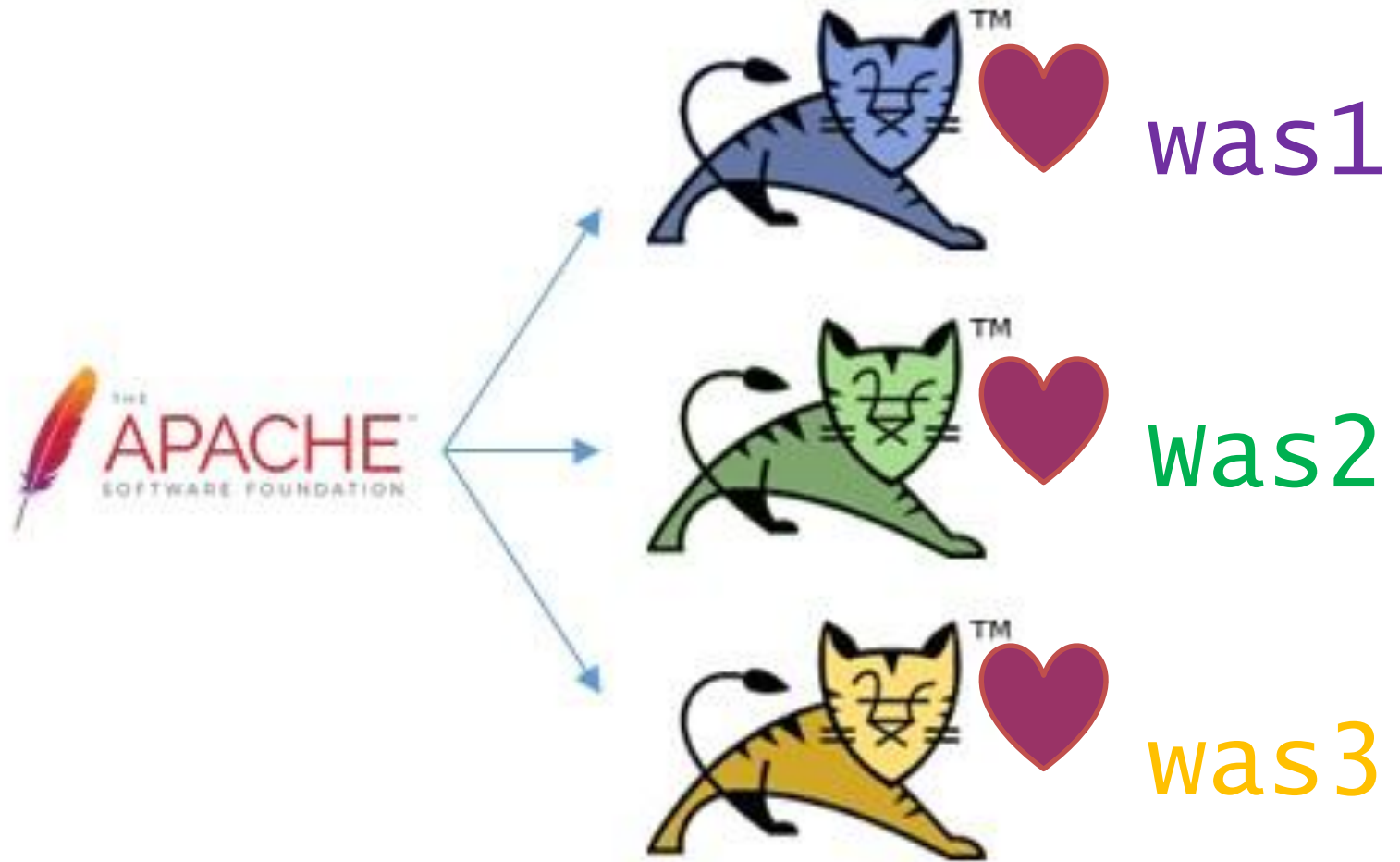
<http://localhost/sample/>

<http://localhost/myfirstapp/>

<http://192.168.128.001/sample/>

<http://192.168.128.001/myfirstapp/>

[실습 7] WAS를 여러 개 해보자. – 서로 같은 app



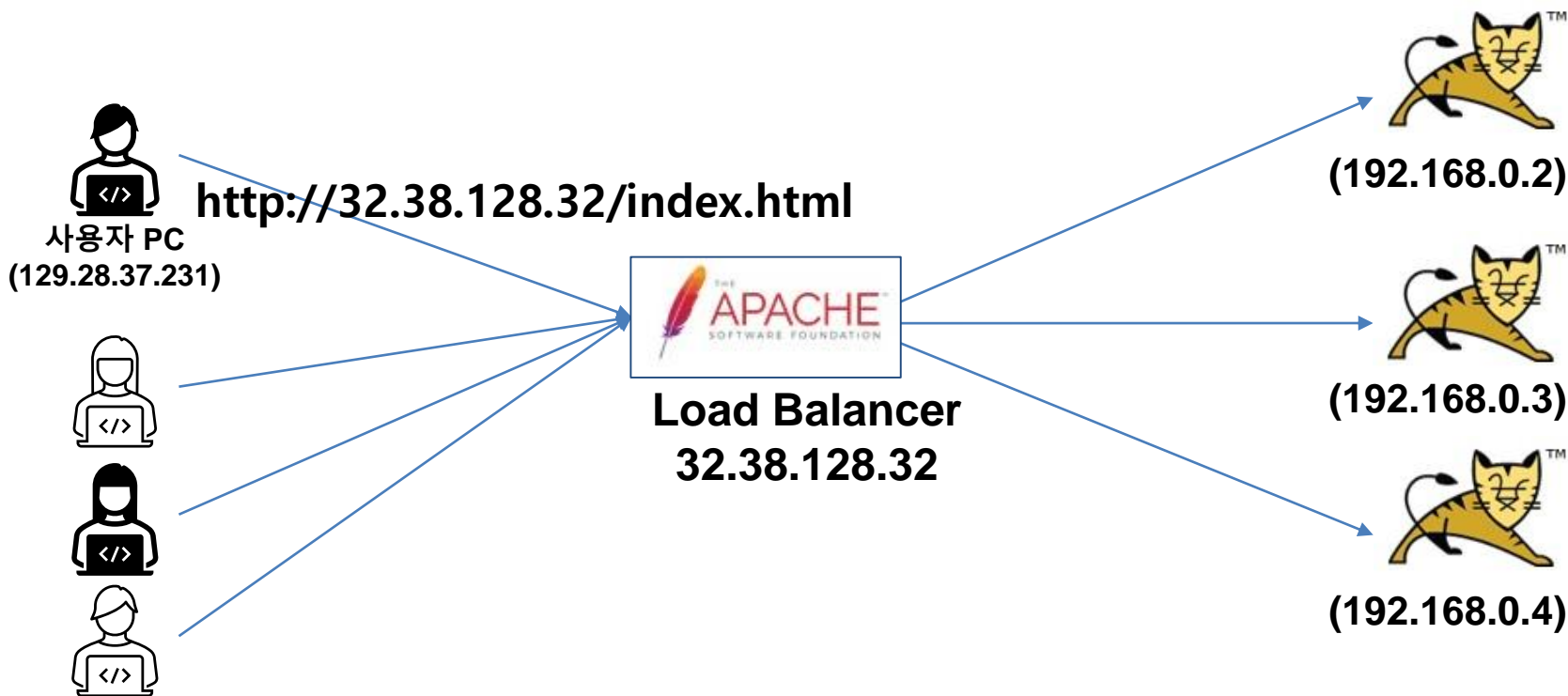
학습 목표

- 웹 서비스에 과도한 부하로 정상적인 서비스가 어려울 수 있으므로, 이를 회피하기 위해 여러 대의 서버에 동일한 서비스를 운영할 수 있도록 도와주는 Load Balancing를 이용해 부하를 분산해서 많은 사용자가 동시에 서비스를 사용할 수 있게 된다.

Load Balancing

■ 로드 밸런싱(Load Balancing)

- 네트워크 트래픽을 하나 이상의 서버나 장비로 분산하기 위해 사용되는 기술
- S/W나 H/W를 통해 로드 밸런싱을 수행할 수 있음
- 서비스 사용자의 요청을 서버로 분산해서 전달

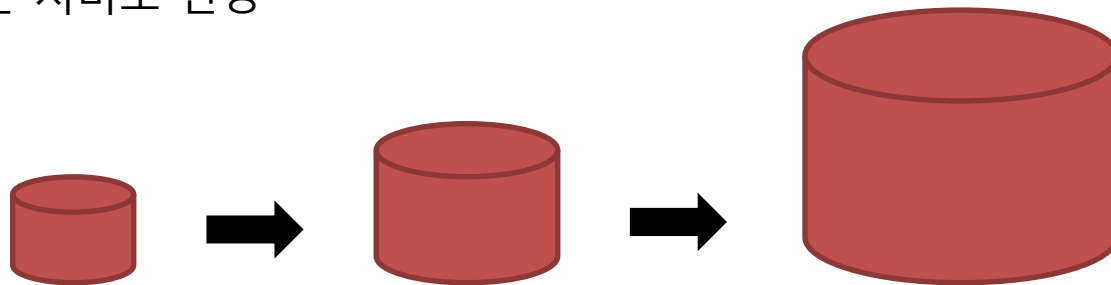


Load Balancing

■ 웹 트래픽 증가에 따른 대응 방법

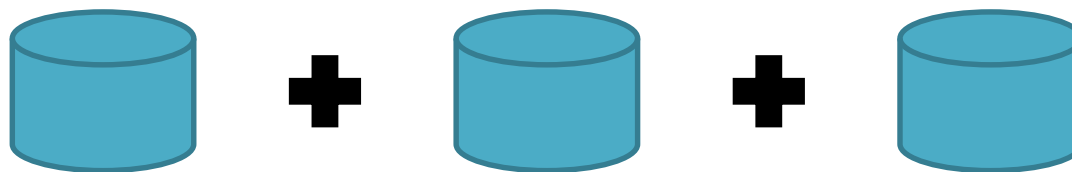
■ 1. Scale Up

- CPU/RAM/Disk 성능/Network 대역폭 등 증가
- 비싸고 성능 좋은 서버로 변경



■ 2. Scale Out (로드 밸런싱과 함께 활용)

- 부하를 처리할 서버 대수를 늘림
- 저렴한 서버 여러 대를 이용해 더 많은 부하를 감당



Load Balancing

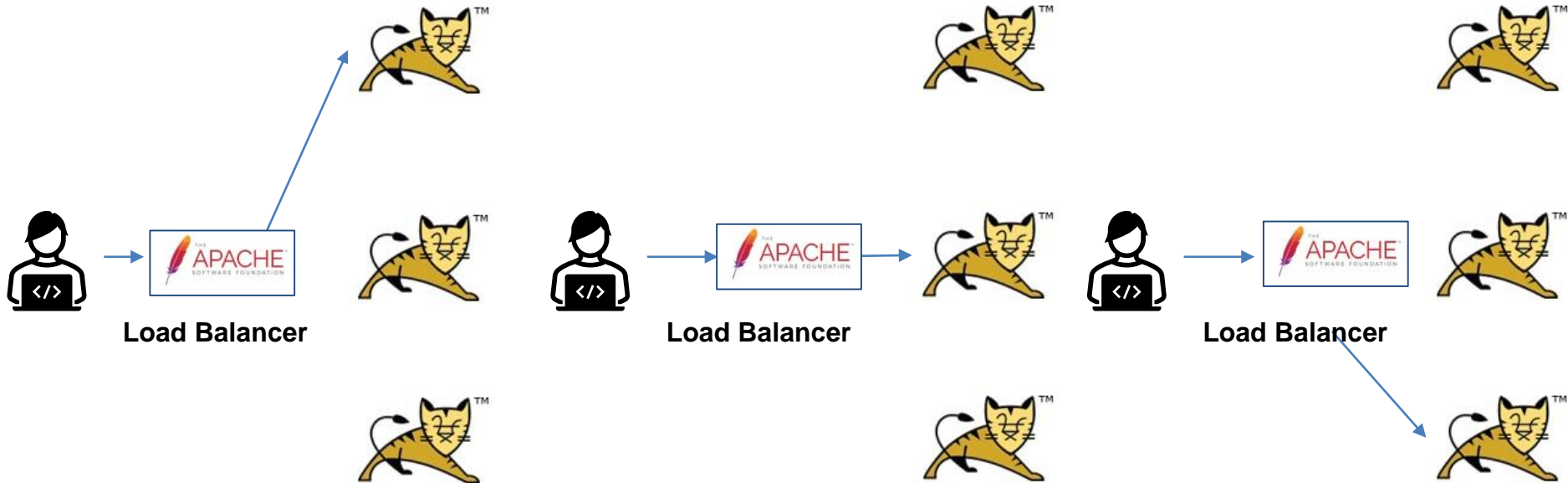
- 같은 그룹에 있는 server들에게 요청이 정책에 맞게 골고루 분배될 수 있도록 한다.
- 같은 그룹에 있는 server들은 동일한 기능을 service한다.

Load Balancing

■ 로드 밸런싱(Load Balancing) 방식

■ 1. Round Robin

- 로드 밸런서에서 서버 선택 시, 순차적으로 서버를 선택하는 방법

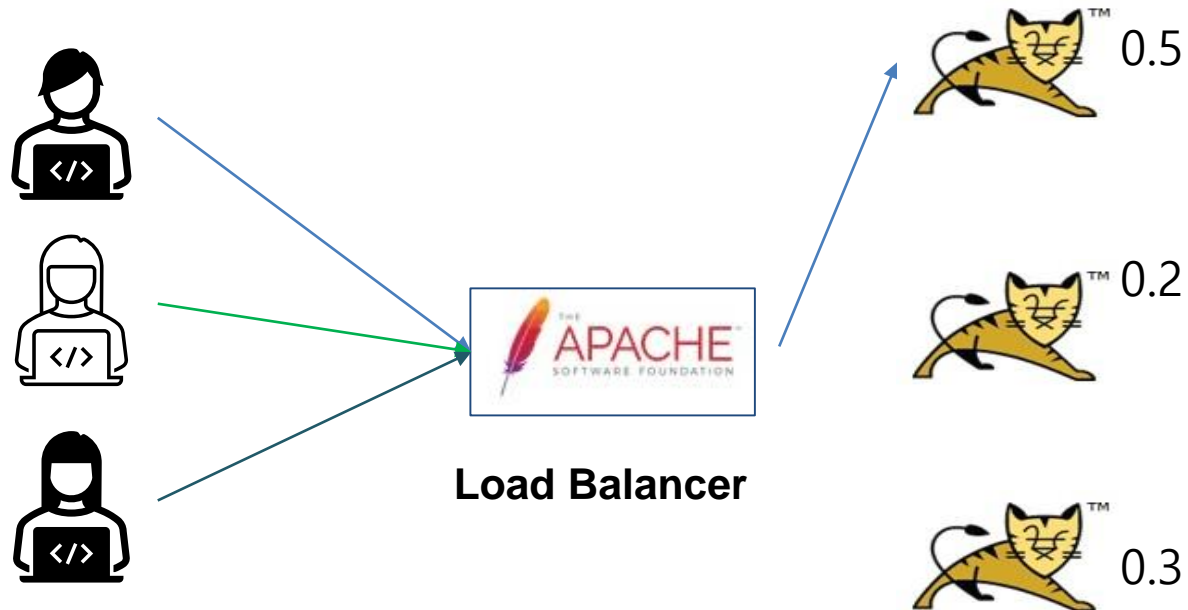


Load Balancing

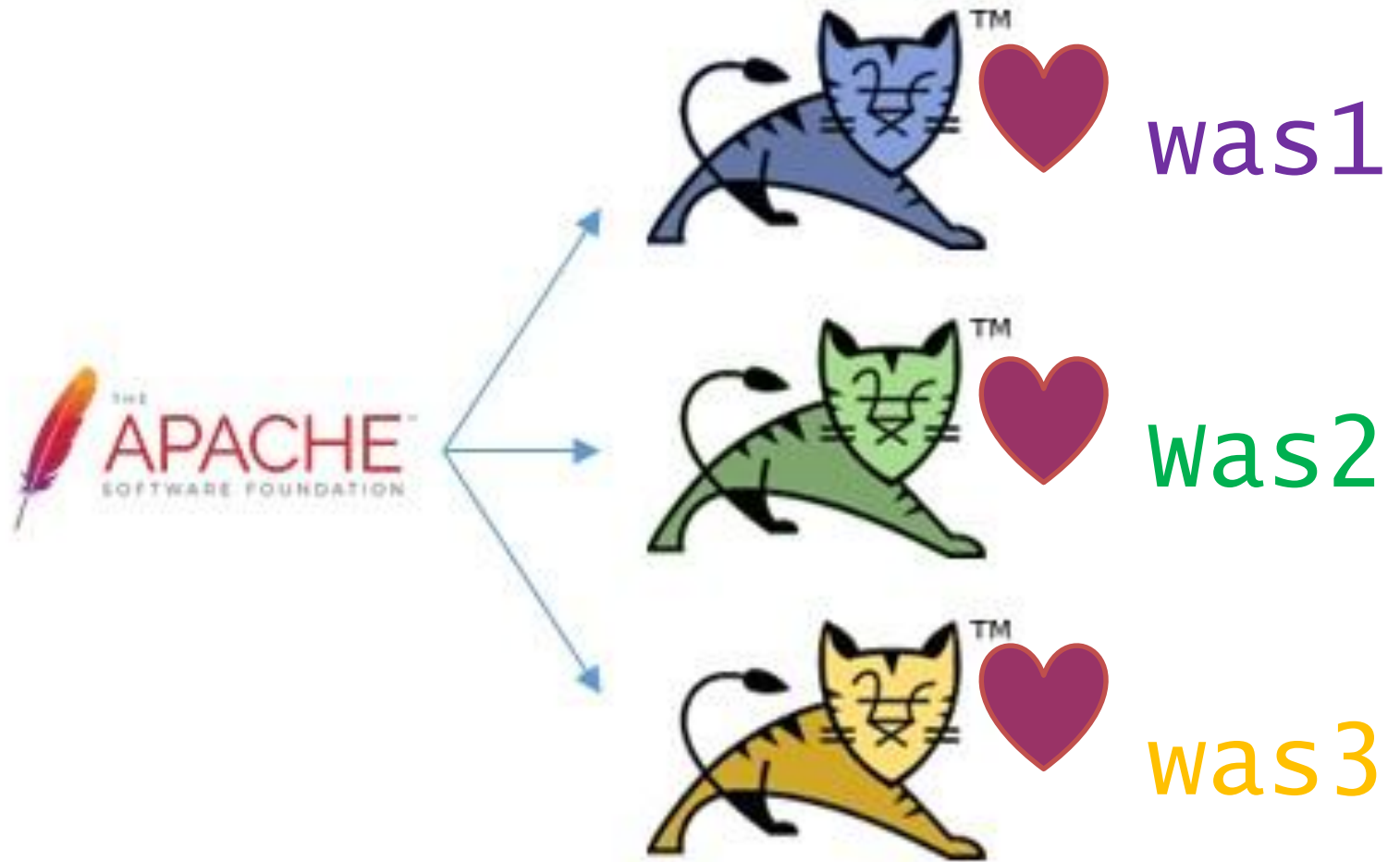
■ 로드 밸런싱(Load Balancing) 방식

■ 2. Weighted Round Robin

- 로드 밸런서에서 서버 선택 시, 비중에 따라서 서버를 선택하는 방법



[실습 7] WAS를 여러 개 해보자. – 서로 같은 app



[실습 7] was가 여러 개인데, 같은 app인 경우

- 1. apache-tomcat connector 설정
 - /etc/libapache2-mod-jk/workers.properties

```
#/etc/libapache2-mod-jk/workers.properties
```

```
worker.list=loadbalancer
```

```
worker.loadbalancer.type=lb
```

```
worker.loadbalancer.balance_workers=was1,was2,was3
```

```
worker.was1.port=port1(해당 port)
```

```
worker.was1.host=ip1(해당 ip)
```

```
worker.was1.type=ajp13
```

```
worker.was1.lbfactor=1
```

```
worker.was2.port=port2(해당 port)
```

```
worker.was2.host=ip2 (해당 ip)
```

```
worker.was2.type=ajp13
```

```
worker.was2.lbfactor=1
```

```
...
```

[실습 7] was가 여러 개인데, 같은 app인 경우

```
#/etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
    ServerName 127.0.0.1
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    JkMount /* loadbalancer
</VirtualHost>
```

JkMount url_pattern worker

=> url_pattern은 worker에게 보냄

JkMount /* loadbalancer => 모든 요청은 loadbalancer에게 보냄

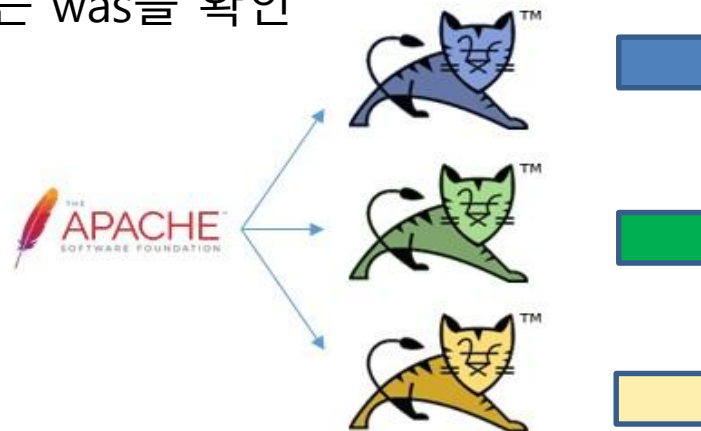
[실습 7] was가 여러 개인데, 같은 app인 경우

```
# /etc/apache2/mods-available/jk.conf
<IfModule jk_module>
JkWorkersFile /etc/libapache2-mod-jk/workers.properties

JkMount /* loadbalancer
```

[실습 7] was가 여러 개인데, 같은 app인 경우

- Load balancing 이 잘 되는지 확인하자
 - 각 tomcat 에 index.html을 내용을 다르게 작성
 - App이름은 같지만, 다른 내용으로 서비스 하는 was를 확인



- Lbfactor는 의미가 뭐지?
 - 가중치
 - Lbfactor를 변경해 보고, 정말 가중치대로 나오는지 확인해보자.
 - lbfactor>0

[실습 7] was가 여러 개인데, 같은 app인 경우

- Apache log를 통하여 확인하자.

```
# /etc/apache2/mods-available/jk.conf

LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so
JkWorkersFile /etc/libapache2-mod-jk/workers.properties
JkLogFile /var/log/apache2/mod_jk.log
JkLogLevel info
JkShmFile /var/log/apache2/jk-runtime-status
JkRequestLogFormat "%w %R %V %T %U %q"
```

- 로그 확인 - /var/log/apache2/mod_jk.log

- `worker.loadbalancer.balance_workers=was1,was2`

```
heejin@heejin-registry:/var/log/apache2$ tail /var/log/apache2/mod_jk.log
[Mon Mar 27 02:07:45.867 2023] loadbalancer was1 localhost 0.004555 /
[Mon Mar 27 02:07:46.254 2023] loadbalancer was2 localhost 0.014872 /favicon.ico
[Mon Mar 27 02:07:47.663 2023] loadbalancer was1 localhost 0.005559 /
[Mon Mar 27 02:07:49.639 2023] loadbalancer was2 localhost 0.011049 /
[Mon Mar 27 02:07:50.989 2023] loadbalancer was1 localhost 0.008506 /
[Mon Mar 27 02:07:53.142 2023] loadbalancer was2 localhost 0.010372 /
[Mon Mar 27 02:07:54.658 2023] loadbalancer was1 localhost 0.004809 /
[Mon Mar 27 02:07:55.794 2023] loadbalancer was2 localhost 0.003841 /
[Mon Mar 27 02:07:56.759 2023] loadbalancer was1 localhost 0.005512 /
[Mon Mar 27 02:07:58.322 2023] loadbalancer was2 localhost 0.004472 /
```

[실습 7] was가 여러 개인데, 같은 app인 경우

■ JkRequestLogFormat

Attribute	Description
%b	Bytes sent, excluding HTTP headers (CLF format)
%B	Bytes sent, excluding HTTP headers
%H	The request protocol
%m	The request method
%p	The canonical Port of the server serving the request
%q	The query string (prepended with a ? if a query string exists, otherwise an empty string)
%r	First line of request
%s	Request HTTP status code
%T	Request duration, elapsed time to handle request in seconds '.' micro seconds
%U	The URL path requested, not including any query string.
%v	The canonical ServerName of the server serving the request
%V	The server name according to the UseCanonicalName setting
%w	Tomcat worker name
%R	Real worker name

[실습 8] WAR 직접 만들어서 deploy 해보자

- Eclipse에서 war 제작하기
- Tomcat에서 war deploy하기

[실습 9] apache-tomcat 연동

- Apache는 아무 page 없이 tomcat으로 바로 연동하는 경우
- Apache에서 static resource를 처리하고, servlet만 tomcat으로 연동하는 경우

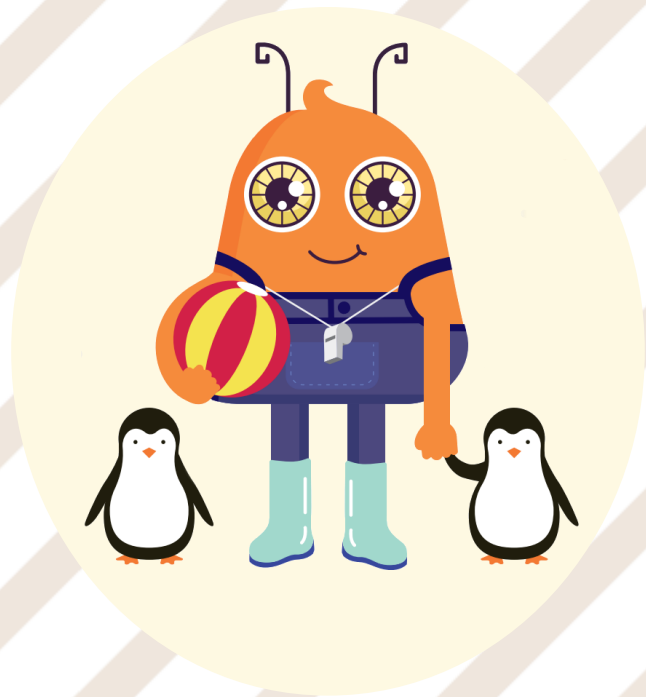
[실습 9] WAS 를 더 설치해 보자

- Apache/Tomcat 조합을 한번 더 설치해 본다.
 - 1. 기본설치(같은 VM) webserver1 - was1
 - 2. webserver (ip1), was (ip2)
- Apache1개, Tomcat 여러 개 조합
 - 같은 app을 사용하는 경우 – load balancing setting
 - 다른 app을 사용하는 경우 – app별로 부하분산
- Oracle Weblogic
- Glassfish
- Tmax JEUS

[실습 9] WAS 를 더 설치해 보자

- Apache-tomcat1 VM 에 설치
 - Apache
 - Tomcat
 - 두개 연결
 - <http://localhost:80> 으로 들어갔을때, 탐캣의 홈페이지가 보이면 성공.

참고자료



JDK 설치

- 본인의 개발환경의 JDK 버전과 동일한것으로 설치
- JDK 버전이 다를경우, app이 정상 동작을 안함

```
$java -version  
$apt-get install openjdk-8*
```

■ JAVA_HOME 설정

- 여기있는 java를 사용하시오. 라는 의미
- Java application들이 \$JAVA_HOME 변수를 활용하여 실행 (e.g. tomcat)
- 여러 java 버전이 설치되어 있을때, 원하는 java 버전을 사용하기 위하여 설정
- /etc/profile 에 설정
 - 시스템 전역 쉘 변수 , login시 가장 먼저 실행됨
 - 설정 파일들은 리부팅, 재로그인 할때만 적용됨
 - 즉시 사용하기 위하여 - source /etc/profile

```
$source /etc/profile
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
export PATH=$PATH:$JAVA_HOME/bin
```

JDK 설치

- Java -version이 /usr/bin/java에 연결되어 있는것이 나옴. 그래서 그 link를 변경해줘야함.

```
heejinlee@ubuntu:~$ java -version
openjdk version "14.0.1" 2020-04-14
OpenJDK Runtime Environment (build 14.0.1+7-Ubuntu-1ubuntu1)
OpenJDK 64-Bit Server VM (build 14.0.1+7-Ubuntu-1ubuntu1, mixed mode, sharing)
heejinlee@ubuntu:~$ which java
/usr/bin/java
heejinlee@ubuntu:~$ ls -al /usr/bin/java
lrwxrwxrwx 1 root root 22 Jul 1 2020 /usr/bin/java -> /etc/alternatives/java
heejinlee@ubuntu:~$ ls -al /etc/alternatives/java
lrwxrwxrwx 1 root root 43 Jul 1 2020 /etc/alternatives/java -> /usr/lib/jvm/java-14-openjdk-amd64/bin/java
heejinlee@ubuntu:~$ ls -al /usr/lib/jvm/
total 32
drwxr-xr-x 6 root root 4096 Mar 16 13:18 .
drwxr-xr-x 128 root root 4096 Mar 16 11:05 ..
lrwxrwxrwx 1 root root 21 Apr 18 2020 java-1.14.0-openjdk-amd64 -> java-14-openjdk-amd64
-rw-r--r-- 1 root root 1984 Apr 18 2020 .java-1.14.0-openjdk-amd64.jinfo
drwxr-xr-x 10 root root 4096 Jul 1 2020 java-14-openjdk-amd64
lrwxrwxrwx 1 root root 20 Nov 3 22:54 java-1.8.0-openjdk-amd64 -> java-8-openjdk-amd64
-rw-r--r-- 1 root root 2764 Nov 3 22:54 .java-1.8.0-openjdk-amd64.jinfo
heejinlee@ubuntu:~$ sudo ln -s /usr/lib/jvm/java-8-openjdk-amd64/bin/java /etc/alternatives/java
heejinlee@ubuntu:~$ java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
```

JDK 설치

#현재 java version확인

\$ java -version

#java의 위치 확인

\$ which java

#어떤 형태인지 확인 - 링크로 되어 있는것을 확인

\$ ls -al /usr/bin/java

#기존링크를 삭제하고 새로 생성 - 원하는 jdk를 가리키도록 변경

\$ unlink 기존링크

\$ ln -s /usr/lib/jvm/java-8-openjdk-amd64/bin/java 기존링크

#변경된 java version확인

\$ java -version

JDK 설치

- Java -version이 동작 안하는 경우
 - PATH가 제대로 설정되어 있지 않아서

```
#/etc/profile  
  
export PATH=$PATH:/usr/bin
```

- JAVA_HOME 설정

```
#/etc/profile  
  
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
export PATH=$PATH:$JAVA_HOME/bin
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
export PATH=$PATH:$JAVA_HOME/bin
```

Java compiler

- Java compiler도 version을 맞춰준다.
- 앞 ppt 참조

Java – Hello World

- Javac HelloWorld.java
- Java HelloWorld

```
public class HelloWorld{  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

Systemd - systemctl

- systemd (system daemon)
 - 컴퓨터가 처음 실행되면 가장 처음 실행되는 daemon
 - 초기에 실행해야 하는 여러 프로세스들을 systemd에 설정해 놓는다.
 - 여러 service의 형태로 만들어서 systemd에 등록함
- Daemon
 - 컴퓨터에서 사용자가 직접 제어하지 않고, 백그라운드에서 돌면서 작업하는 프로그램
- systemctl
 - systemd를 제어하는 도구, 명령어
 - 설정 변경, 상태 확인, 구동, 중단 등의 관리기능 제공
 - enable/disable – 특정 프로그램을 최초 시스템 실행시에 자동으로 시작할지 여부 결정



Thank You
