

# JIRA Service 등록(CentOS 7)

OS CentOS 7 이하는 Atlassian 제품을 설치시 자동으로 서비스 등록을 시켜 주지만 CentOS 7 부터는 수동으로 Service 등록해 줘야 한다.

서비스 등록 하는 절차

```
vi /etc/systemd/system/jira.service

[Unit]
Description=Atlassian JIRA Server Service
After=network.target mysqld.service

[Service]
Type=forking
User=jira
Environment=JAVA_HOME=/data/java/jdk
ExecStart=/opt/atlassian/jira/bin/start-jira.sh
ExecStop=/opt/atlassian/jira/bin/stop-jira.sh

[Install]
WantedBy=multi-user.target
```

Bin 이나 exe 로 설치하면 jdk를 따로 설치할 필요가 없고, 또한 대부분 database를 다른 서버에 설치해주므로 아래와 같이 설정할 수 있다.

```
vi /etc/systemd/system/jira.service

[Unit]
Description=JIRA Service
After=network.target firewalld.service httpd.service

[Service]
Type=forking
User=jira
ExecStart=/opt/atlassian/jira/bin/start-jira.sh
ExecStop=/opt/atlassian/jira/bin/stop-jira.sh
ExecReload=/opt/atlassian/jira/bin/stop-jira.sh | sleep 60 | /opt/atlassian/jira/bin/start-jira.sh

[Install]
WantedBy=multi-user.target
```

Description

해당 유닛에 대한 상세한 설명을 포함한다.

After=network.target mysqld.service

해당 유닛이 시작된 이후 나열된 유닛이 실행 한다.(network와 mysqld 서비스가 실행후 작동하게 된다.)

Type=forking

"simple" (기본값) 유닛이 시작된 경우 즉시 systemd 는 유닛의 시작이 완료되었다고 판단한다. 다른 유닛과 통신하기 위해 소켓을 사용하는 경우 이러한 설정을 사용하면 안된다.

"forking" 자식 프로세스를 생성이 완료되는 단계까지를 systemd 가 시작이 완료되었다고 판단하게 된다. 부모 프로세스를 추적할 수 있도록 PIDFile= 필드에 PID 파일을 선언해 주어야 한다.

"oneshot" 은 "simple" 과 다소 유사하지만 단일 작업을 수행하는데 적합한 타입이다. 또한 실행 이후 해당 실행이 종료되더라도 RemainAfterExit=yes 옵션을 통해 유닛이 활성화 상태로 간주할 수 있다.

"notify" 은 "simple" 과 동일하다. 다만 유닛이 구동되면 systemd 에 시그널을 보낸다. 이때 시그널에 대한 내용은 [libsystemd-daemon.so](http://libsystemd-daemon.so) 에 선언 되어 있다.

"dbus" DBUS 에 지정된 BusName 이 준비될때까지 대기한다. 즉 DBUS 준비가 완료된 이후 유닛이 시작되었다고 간주한다.

User=JIRA

유닛의 프로세스를 수행할 사용자명, 그룹명 등을 지정한다.

Environment=

해당 유닛에서 사용할 환경 변수를 선언한다. 또한 반드시 "Exec\*=" 옵션보다 상단에 위치해야 한다. 예제는 아래와 같다.  
Environment="ONE=one" 'TWO=two two'

EnvironmentFile=

해당 유닛에서 사용할 환경 변수 파일을 선언한다. 환경 변수 파일에서 '#' 와 ';' 로 시작되는 라인은 주석으로 처리된다. "Environment=" 와 같이 사용하는 경우 "Environment=" 옵션값이 먹게 된다. 또한 반드시 "Exec\*=" 옵션보다 상단에 위치해야 한다.

ExecStart=

구동 명령어(스크립트)를 선언한다. 실행 명령어는 반드시 절대 경로 또는 변수(\${STRINGS} 따위) 로 시작해야 한다. 다중 명령어를 지원한다. 예제는 아래와 같다.  
ExecStart="commnad 1; command 2; command 3" 또는  
ExecStart="commnad 1"  
ExecStart="command 2"

ExecStop=

중지 명령어(스크립트)를 선언한다. "ExecStart=" 동일하게 사용하면 된다. 중지 방식은 "KillMode=" 로 지정된다.  
WantedBy=, RequiredBy=

"systemctl enable" 명령어로 유닛을 등록할때 등록에 필요한 유닛을 지정한다. 해당 유닛을 등록하기위한 종속성 검사 단계로 보면 된다.

생성된 서비스를 아래 명령어로 등록 및 시작

```
systemctl enable jira.service
systemctl daemon-reload
systemctl start jira.service
systemctl status jira.service
```