



ABLESTACK Online Docs
ABLESTACK-V4.0-4.0.15

WAS 구성

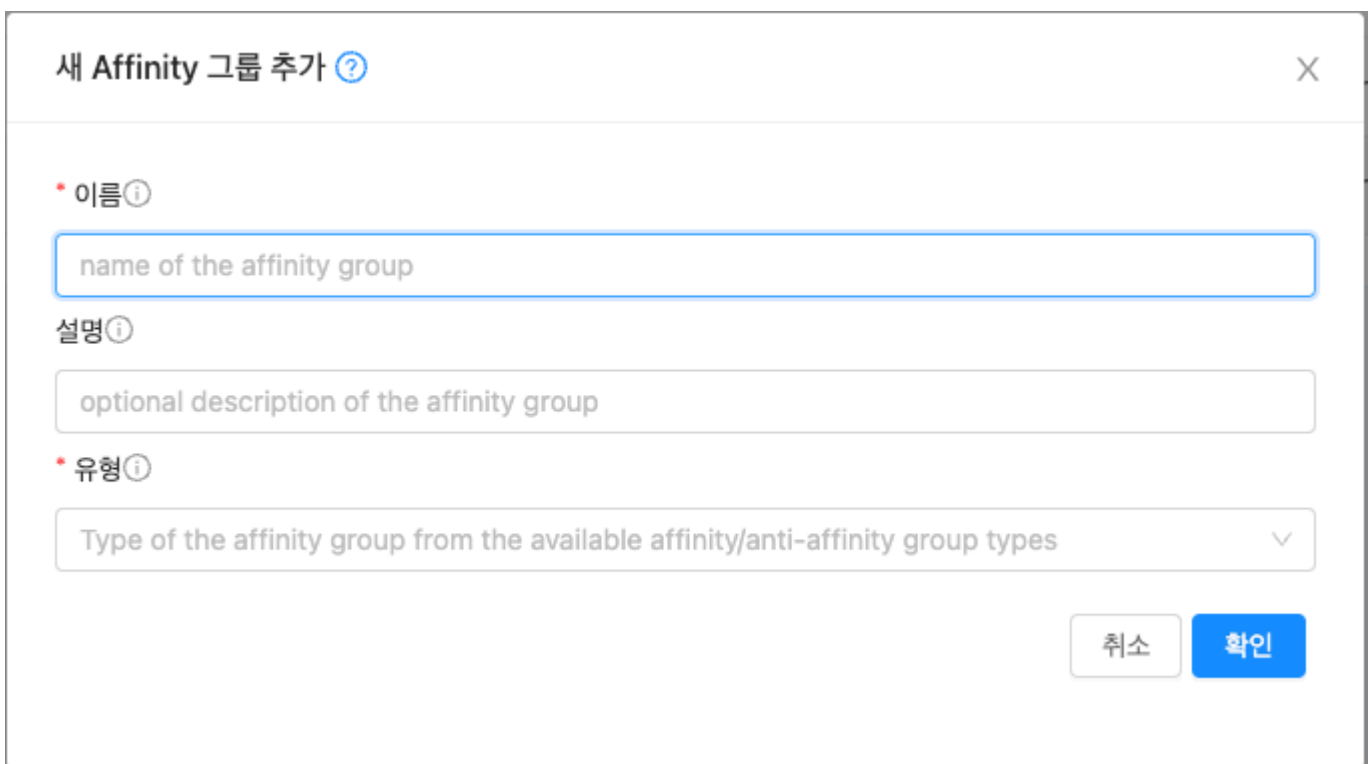
ABLESTACK Mold를 이용한 이중화를 통한 고가용성 기능을 제공하는 리눅스 기반의 3계층 구조의 구성 단계 중, 네 번째 단계인 WAS 구성에 대한 문서입니다.

가상머신에 도커 컨테이너를 이용하여 NodeJS WAS 2개를 구성하고 1 개의 가상머신에 Samba 스토리지를 생성하여 WAS에서 구동시킬 웹소스를 저장, 공유합니다. 이를 하나의 클러스터로 구성하는 방법은 다음과 같은 절차로 수행됩니다.

- Affinity 그룹 생성
- 가상머신 생성
- 데이터 디스크 설정
- 보안 설정
- Samba 스토리지 가상머신 구성
- WAS 가상머신 구성
- 로드 밸런서(부하 분산) 설정

Affinity 그룹 생성

가상머신을 생성하기 전, Anti Affinity 그룹을 생성하여 어느하나의 서브넷에 속한 가상머신들이 특정 호스트 한 곳에 몰려 실행하도록 하거나 반대로 몰려 실행되지 않도록 합니다. 이중화를 위해 Affinity 그룹을 anti-affinity 유형으로 WEB, WAS, DB 각 추가해야 합니다. 이를 위해 **컴퓨터 > Affinity 그룹** 화면으로 이동하여 **새 Affinity 그룹 추가** 버튼을 클릭합니다. 클릭하게 되면 다음과 같은 입력항목을 확인할 수 있습니다.



The screenshot shows a modal window titled "새 Affinity 그룹 추가" (Add New Affinity Group). It contains three required input fields: "이름" (Name) with a placeholder "name of the affinity group", "설명" (Description) with a placeholder "optional description of the affinity group", and "유형" (Type) with a dropdown menu showing "Type of the affinity group from the available affinity/anti-affinity group types". At the bottom right, there are two buttons: "취소" (Cancel) and "확인" (Confirm).

- 이름 : 서브넷을 분별할 수 있는 Affinity 그룹 이름을 입력합니다.
- 설명 : Affinity 그룹에 대한 설명을 입력합니다.
- 유형 : Affinity 그룹에 대한 유형을 선택합니다. Anti 여부를 선택할 수 있습니다.

새 Affinity 그룹 추가 대화상자에서의 입력 항목 예제는 다음과 같습니다.

- 이름 : **ablecloud-3tier-linux-was**
- 설명 : **3tiers-linux**의 was 구성 시 사용되는 Affinity 그룹입니다.
- 유형 : **host anti-affinity (Strict)**

i Affinity 그룹 유형

host anti-affinity: 가능한 한 서로 다른 호스트에 인스턴스를 배포합니다.

host affinity: 가능한 한 동일한 호스트에 인스턴스를 배포합니다.

- Non-Strict 옵션은 마지막으로 해당 가상머신을 실행했던 호스트를 고려하지 않고 가상머신을 시작합니다.

가상머신 생성

ABLESTACK Mold는 기본적으로 템플릿을 이용해 가상머신을 생성하고 사용하는 것을 권장합니다. 따라서 가상머신을 생성하기 전에 먼저 "**가상머신 사용 준비**" 단계를 통해 CentOS 기반의 가상머신 템플릿 이미지를 생성하여 등록하는 절차를 수행한 후 가상머신을 생성해야 합니다.

가상머신을 추가하기 위해 **컴퓨터 > 가상머신** 화면으로 이동하여 **가상머신 추가** 버튼을 클릭합니다. **새 가상머신** 마법사 페이지가 표시됩니다. 해당 페이지에서는 **템플릿을 이용한 가상머신 생성** 문서를 참고하여 가상머신을 생성합니다.

i 템플릿을 이용한 가상머신 생성

템플릿을 이용한 가상머신 추가를 위해 **템플릿을 이용한 가상머신 생성** 문서를 참고하십시오.

입력 항목 예시는 다음과 같습니다.

- WAS 가상머신 1
 - 배포 인프라 선택 : **Zone**
 - 템플릿/ISO : **Rocky Linux 9.0 기본 이미지 템플릿** * Rocky Linux release 9.0 (Blue Onyx)
 - 컴퓨터 오퍼링 : **1C-2GB-RBD-HA**
 - 데이터 디스크 : * 디폴트로 생성합니다.
 - 네트워크 : **was** * VPC명이 일치하는지 확인합니다.
 - IP: **192.168.2.11**
 - SSH 키 쌍 : **3tier_linux_keypair**
 - 확장 모드 :
 - Affinity 그룹 : **ablecloud-3tier-linux-was**
 - 이름 : **ablecloud-3tier-linux-was-01**
- WAS 가상머신 2
 - 배포 인프라 선택 : **Zone**
 - 템플릿/ISO : **Rocky Linux 9.0 기본 이미지 템플릿** * Rocky Linux release 9.0 (Blue Onyx)

- 컴퓨터 오퍼링 : **1C-2GB-RBD-HA**
- 데이터 디스크 : * 디폴트로 생성합니다.
- 네트워크 : **was** * VPC명이 일치하는지 확인합니다.
 - IP: **192.168.2.12**
- SSH 키 쌍 : **3tier_linux_keypair**
- 확장 모드 :
 - Affinity 그룹 : **ablecloud-3tier-linux-was**
- 이름 : **ablecloud-3tier-linux-was-02**
- SAMBA 스토리지 가상머신
 - 배포 인프라 선택 : **Zone**
 - 템플릿/ISO : **Rocky Linux 9.0 기본 이미지 템플릿** * Rocky Linux release 9.0 (Blue Onyx)
 - 컴퓨터 오퍼링 : **1C-2GB-RBD-HA**
 - 데이터 디스크 : **100GB-WB-RBD**
 - 네트워크 : **was** * VPC명이 일치하는지 확인합니다.
 - IP: **192.168.2.13**
 - SSH 키 쌍 : **3tier_linux_keypair**
 - 확장 모드 : * 디폴트로 생성합니다.
 - 이름 : **ablecloud-3tier-linux-was-storage**

⚠ 스토리지 가상머신의 Affinity 그룹 적용

스토리지 가상머신의 Anti-Affinity 그룹 적용은 권장되지 않습니다. 어떠한 이유로 스토리지 가상머신을 구동하던 호스트가 중단될 경우, 스토리지 가상머신은 다른 호스트로 이관을 시도하게 되는데 이 때 Anti-Affinity의 영향으로 스토리지 가상머신이 재기동되지 않을 수 있습니다.

데이터 디스크 설정

안정적인 운영을 위해 기본 RootDisk가 아닌 고용량의 스펙을 가진 디스크로의 데이터 저장이 필요합니다. 이를위한 사전작업으로 가상머신 생성 시 추가했던 데이터 디스크에 대한 사용을 위해 "[데이터 디스크 설정](#)" 문서를 참고하여 수행합니다.

Note

SAMBA 스토리지 가상머신에만 데이터 디스크 설정을 수행합니다.

보안 설정

생성한 가상머신에 대해 보안 설정을 하여 허용되지 않은 접근을 차단하고 필요한 서비스만 운영할 수 있도록 설정합니다.

Note

WAS 가상머신 1, 2 및 스토리지 가상머신에 대해 실행 및 설정을 적용합니다.

네트워크 방화벽 해제

방화벽은 들어오고 나가는 네트워크 트래픽을 모니터링하고 필터링하는 방법입니다. 특정 트래픽을 허용할지 차단할지 결정하는 일련의 보안 규칙을 정의하여 작동합니다. CentOS 운영체제에서는 firewalld라는 이름의 방화벽 데몬과 함께 해당 기능이 제공됩니다.

`firewall-cmd` 명령어를 이용하여 samba 서비스에 대한 방화벽을 해제하고 `--permanent` 옵션을 사용하여 영구적으로 적용합니다.

```
firewall-cmd --zone=public --permanent --add-service=samba
firewall-cmd --reload
```

Info

samba 서비스는 TCP 포트 139, 445와 UDP 포트 137, 138을 사용합니다.

Samba 스토리지 가상머신 구성

WAS 가상머신 1, 2와 데이터를 공유할 Samba 스토리지 가상머신 구성을 위해 아래 절차를 수행합니다.

패키지 관리 명령어인 **dnf**를 사용하여 Samba 패키지를 설치합니다.

```
1 dnf install samba
```

WAS와 파일을 공유할 SAMBA 스토리지의 공유폴더를 생성하고 적절한 권한을 부여합니다. 스토리지 공유폴더 경로 예시는 `/mnt/data/shared_folder`입니다.

```
mkdir -p /mnt/data/shared_folder
chmod -R 777 /mnt/data/shared_folder
```

Samba 사용자 생성을 위해 먼저 리눅스 **user1** 계정을 생성하고 비밀번호를 적절한 부여합니다.

```
useradd user1
passwd user1
```

리눅스 계정과 동일한 이름으로 samba 계정을 생성합니다.

```
smbpasswd -a user1
```

WAS에서 구동할 샘플 소스를 Samba 스토리지의 공유폴더로 다운로드하기 위해 먼저 git 패키지를 설치한 후 생성한 폴더에 Git 샘플 소스를 다운로드하고 경로를 수정합니다.

```
dnf install git
git clone https://github.com/ablecloud-team/ablestack-docs-sample.git /mnt/data/shared_folder/

cd /mnt/data/shared_folder/
rm -rfv !("linux-3tier-example")
mv linux-3tier-example/* ./
rm -rf linux-3tier-example/
```

i 다른 웹소스를 NodeJS 서버에 구동하기

위 예시에서 제시된 Git 소스가 아닌 다른 샘플소스를 사용하려면 Docker 컨테이너 이미지에서 해당 소스를 구동할 수 있도록 빌드된 이미지여야 합니다. 즉 공식 Docker NodeJS 이미지에 Dockerfile을 구성하여 새로운 NodeJS 이미지로 빌드하여야 합니다.

다운로드 받은 샘플 소스에는 DB와 통신하는 NodeJS 모듈이 포함되어 있습니다. 이 파일들을 편집하여 사전에 구성된 DB 가상머신의 구성정보로 변경합니다. 두 경로에 위치한 `index.js` 파일을 편집합니다.

```
vi /mnt/data/shared_folder/router/signUp/index.js
vi /mnt/data/shared_folder/router/login/index.js
```

Host, User, Password, Port, Database 정보를 사전에 구성된 DB 정보로 변경합니다. **host** 의 값은 DB 가상머신의 내부 로드 밸런서 주소로 입력합니다.

index.js

```
1 // DATABASE setting
2 var connection = mysql.createPool({
3     host: '192.168.3.26',
4     user: 'root',
5     password: 'PASSWORD!',
6     port: '3306',
7     database: 'galeradb'
8 });
```

WAS 서버 구동을 위한 NodeJS 모듈 패키지를 설치합니다. 먼저 18.0.0 버전 이상의 NodeJs를 설치하기 위해 `make`, `git`, `gcc` 와 같은 개발 도구를 설치한 후 `nodejs`를 설치합니다.

```
dnf groupinstall "Development Tools"
dnf module install nodejs:18
```

`npm install` 명령어를 실행하여 **package.json** 파일에 포함된 의존성 패키지들을 일괄적으로 설치합니다.

```
cd /mnt/data/shared_folder/
npm install
```

Samba 설정 파일을 열어 Samba 사용자 계정 "user1" 의 정보를 입력합니다.

vi /etc/samba/smb.conf

```
1          [user1]
2          path = /mnt/data/shared_folder
3          # 사용 가능한 공유 목록에 디렉토리를 보여줄지 여부를 설정합니다.
4          browseable = yes
5          read only = no
6          writable = yes
7          #user1에 쓰기 권한 부여합니다.
8          write list = user1
9          # 이 공유에서 새로 만든 파일에 대한 권한을 설정합니다.
10         force create mode = 0777
11         # 이 공유에서 새로 만든 디렉토리에 대한 권한을 설정합니다.
12         force directory mode = 2770
13         public = yes
```

Samba 스토리지의 공유폴더에 대한 Selinux 보안설정을 합니다.

```
setsebool -P samba_enable_home_dirs on          # 삼바 홈 디렉토리 읽기/쓰기 권한 부여
setsebool -P samba_export_all_rw on            # (읽기, 쓰기) 또는 setsebool -P
samba_export_all_ro on (읽기만)
chcon -R -t samba_share_t /mnt/data/shared_folder # 하위디렉토리 포함 특정디렉토리 삼바권한부여
```

Samba Storage 가상머신의 smb 서비스를 시작합니다.

```
systemctl enable smb
systemctl start smb
```

WAS 가상머신 구성

WAS 가상머신에서 Samba 패키지를 설치합니다.

```
dnf install samba samba-client cifs-utils
```

Samba 스토리지 가상머신과 파일을 공유할 폴더를 생성합니다.

```
mkdir -p /mnt/data/shared_folder
chmod -R 777 /mnt/data/shared_folder
```

Samba 사용자 생성을 위해 먼저 리눅스 **user1** 계정을 생성하고 비밀번호를 적절한 부여합니다.

```
useradd user1
passwd user1
```

리눅스 계정과 동일한 이름으로 samba 계정을 생성합니다.

```
smbpasswd -a user1
```

Samba 스토리지 마운트 시, 명령줄에 계정 및 패스워드를 노출하는 대신 Samba Storage Node에서 설정한 내용으로 `.smb.cred` 계정정보 파일을 생성합니다.

/root/.smb.cred

```
1          username=user1
2
3          # 패스워드를 입력하세요.
4          password=PASSWORD
```

Samba Storage 가상머신의 smb 서비스를 시작합니다.

```
systemctl enable smb
systemctl start smb
```

Samba 스토리지를 마운트합니다.

```
mount -t cifs -o credentials=/root/.smb.cred,vers=3.0 //192.168.2.13/user1
/mnt/data/shared_folder

# cifs: 프로토콜
# credentials: samba 계정정보
# user1: samba 스토리지 정보 (경로 정보 포함된 /etc/samba/smb.conf 내용)
# /mnt/data/shared_folder: 마운트할 위치
```

추가적으로 재부팅 시 자동으로 마운트가 적용되도록 합니다.

/etc/fstab

```
1          //192.168.2.13/user1 /mnt/data/shared_folder cifs
2          credentials=/root/.smb.cred,vers=3.0,iocharset=utf8 0 0
3
4          # cifs: 프로토콜
5          # credentials: samba 계정정보
6          # user1: samba 스토리지 정보 (경로 정보 포함된 /etc/samba/smb.conf 내용)
          # /mnt/data/shared_folder: 마운트할 위치
```

WAS 가상머신 1,2 에서 실행할 NodeJS 컨테이너 이미지를 다운로드 받습니다. 해당 이미지는 샘플 웹소스를 구동하기 위해 사용자가 별도로 빌드한 이미지로써 컨테이너 구동 시 `server.js` 를 실행하도록 제작되었습니다.

```
podman pull docker.io/ablecloudteam/nodejs-server:linux-0.1
```

다운로드한 NodeJS 컨테이너 이미지를 실행합니다. WAS가 정상적으로 로드 벨런싱되는 지 확인하기 위해 WAS 가상머신의 이름에 따라 `--hostname` 옵션 값을 지정합니다.

```
podman run \
--privileged=true \
-d \
-p 5000:3000 \
--name nodejs-server \
--hostname was-container-1 \
--restart always \
-v /mnt/data/shared_folder:/usr/src/app \
ablecloudteam/nodejs-server:linux-0.1

# run: 컨테이너를 실행합니다.
# --privileged=true: 컨테이너 시스템 주요 자원에 접근할 수 있는 권한 취득
# -d: detached 모드 (컨테이너 백그라운드 실행)
# -p: 포트포워딩 (외부:내부)
# --name: 컨테이너 이름
# --hostname: 컨테이너 호스트네임을 지정합니다.
# --restart: 컨테이너 오류 시, 항상 재시작
# -v: 컨테이너의 특정 폴더와 로컬의 폴더를 서로 공유
# ablecloudteam/nodejs-server:linux-0.1: 다운로드한 이미지 이름
```

사용자 정의 데몬인 서비스를 생성하여 가상머신이 부팅될 때 NodeJs컨테이너가 자동으로 실행하도록 할 수 있습니다.

```
podman generate systemd nodejs-server > /etc/systemd/system/nodejs-server.service
systemctl enable nodejs-server.service
systemctl daemon-reload
```

로드 밸런서(부하 분산) 설정

Mold 사용자 또는 관리자는 서버넷에서 수신된 트래픽을 해당 서버넷 내의 여러 가상머신간에 부하 분산되도록 규칙을 만들 수 있습니다. 예를 들어 WAS 계층에 도달한 트래픽은 해당 WAS 서버넷의 다른 가상머신으로 리디렉션됩니다. 로드 밸런서를 설정하면 Health Check를 통해 장애 여부를 판단하고 노드에 이상이 발생하면 다른 정상 동작중인 노드로 트래픽을 보내주는 Fail-over가 가능합니다. 내부 로드 밸런서 규칙 생성을 위해 아래 문서를 참고합니다.

내부 로드 밸런서 규칙 생성

템플릿을 이용한 가상머신 추가를 위해 [내부 로드 밸런서 규칙 생성](#) 문서에서 [내부 LB 규칙 생성](#) 항목을 참고하십시오.

입력 항목 예시는 다음과 같습니다.

- 이름 : **was-lb**
- 설명 : **WAS의 내부 로드 밸런서입니다.**
- 소스 IP 주소 : **192.168.2.26**
- 소스 포트 : **5000**
- 가상머신 포트 : **5000**
- 알고리즘 : **최소 접속**

생성된 내부 로드 밸런서 규칙을 선택한 후, **가상머신 할당** 버튼을 클릭하여 WAS 가상머신 1,2 를 할당합니다.

가상머신 할당 ?

×

이름	상태	내부 이름	이름 표시	IP	계정	Zone 이름	선택
ablecloud-3tier-linux-was-storage	● 실행중		ablecloud-3tier-linux-was-storage		test	Zone1	<input type="checkbox"/>
ablecloud-3tier-linux-was-02 192.168.2.12 (기본) ×	● 실행중		ablecloud-3tier-linux-was-02		test	Zone1	<input checked="" type="checkbox"/>
ablecloud-3tier-linux-was-01 192.168.2.11 (기본) ×	● 실행중		ablecloud-3tier-linux-was-01		test	Zone1	<input checked="" type="checkbox"/>

총 3 항목 < 1 > 10 / 쪽 ▾

취소

확인

ABLESTACK Online Docs