



docker

5장 보충 자료

Azure Storage를 Data Volume
활용하기

4-Azure Storage를 Data Volume으로 사용하기

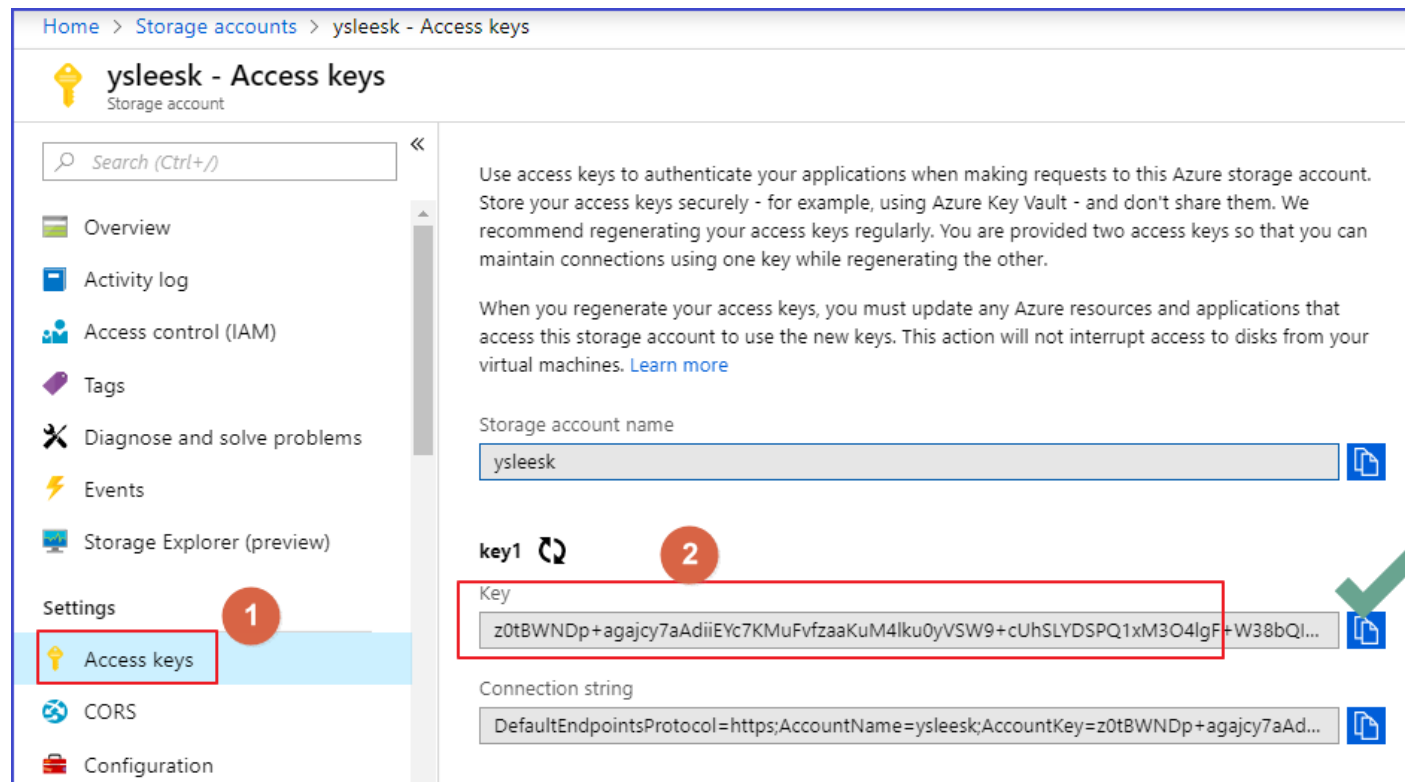
- Container Data Volume를 Azure File Storage로 사용하기
 - Docker Image의 Contents 파일이 저장되는 곳
 - **Docker Image 내**
 - **Docker Host의 특정한 디렉터리 및 volume**
 - **GitHub.com**
 - **Azure Storage Account**
 - Docker Hub에 올려 둔 docker image의 Contents를 GitHub.com에서 받을 수 있다. 그러면 GitHub에서 수정한 내용이 docker Image로 전달되고 그 Image의 Content에 접속하기 위해서는 Container를 실행하면 된다

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - Azure Storage Account를 **Container의 Data Directory**로 지정하기
 - Container가 Azure File Storage를 사용하기 위해 반드시 Docker Host로 사용할 **Azure VM** 2대를 준비한다
 - ubuntu 14.04 이상 또는 CentOS 7.5도 사용 가능하다
 - 각 VM의 공인 IP Address를 확인해 둔다
 - Azure File Storage는 SMB3.0 기술을 이용하고 있다.
 - **국내 ISP에서 인터넷으로 나가는 445번 포트를 차단**하고 있기 때문에 **한국에서는 접속이 안된다**
 - 그래서 실습을 할 때는 **반드시 Azure VM 상에서** 진행해야 한다
 - 다른 Docker Host에서 실행중인 Container들이 공유된 동일한 Data volume을 이용할 수 있는 것을 말한다
 - 2대의 Linux docker host를 준비하여 nginx라는 도커 이미지를 사용하여 Azure File Storage에 Container가 이용하는 Volume을 생성한다
 - **Azure File Storage의 ID와 암호(Access Keys)를 기억해 둔다**

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - **standard** Storage Account를 생성한다
 - 이 File Storage에 접속할 Access Key를 복사해 둔다



- AZURE_STORAGE_ACCOUNT=**ysleesk**
AZURE_STORAGE_ACCOUNT_KEY=**z0tBWNDp+agajc******

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - Docker host로 사용한 Linux VM을 2대 생성한다
 - GUI로 VM을 생성한다
 - ssh로 Azure Linux VM들에 접속한다
 - **ssh adminuser@ip_address**
 - **sudo passwd root** (##root 계정의 암호 재설정)
 - **su root** (##root로 전환)
 - 각 VM에 docker를 설치한다
 - **curl -sSL http://get.docker.com | sh**
 - **systemctl start docker**
 - **systemctl enable docker**
 - **docker version**

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - 첫 번째 Linux docker host에서 작업하기
 - azurefile plugin 설치하면서 Azure File Storage에 로그인하기
 - docker plugin install [jmaitrehenry/azurefile](#)
AZURE_STORAGE_ACCOUNT=[ysleesk](#)
AZURE_STORAGE_ACCOUNT_KEY=[z0tBWNDp+agajc==](#)
 - docker plugin ls
 - docker container가 사용할 Azure File Storage의 volume 생성하기
 - docker volume ls
 - docker volume create --name [vol1](#) -d [jmaitrehenry/azurefile](#) -o [share=myvol](#)
 - docker volume ls

```
root@ubuntu1:~# docker volume ls
DRIVER                                VOLUME NAME
jmaitrehenry/azurefile:latest        vol1
root@ubuntu1:~#
```

NAME	MODIFIED	QUOTA
myvol	2018년 7월 7일 오후 10:04:38	5 TiB

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - 두 대의 Linux Docker host에서 nginx 도커 이미지를 실행하여 동일한 Web Contents를 사용하도록 한다
 - nginx 웹 서버의 root director는 /usr/share/nginx/html/ 이다
 - 첫 번째 Docker host에서 작업하기
 - 이미 Azure File Storage용 Docker Plugin이 설치되어 있고, Azure File Storage에 로그인되어 있다
 - **docker plugin ls**

```
root@ubuntu1:~# docker plugin ls
```

ID	NAME	DESCRIPTION	ENABLED
43d1496d4335	jmaitrehenry/azurefile:latest	Volume plugin for Azure File Storage	true

```
root@ubuntu1:~#
```

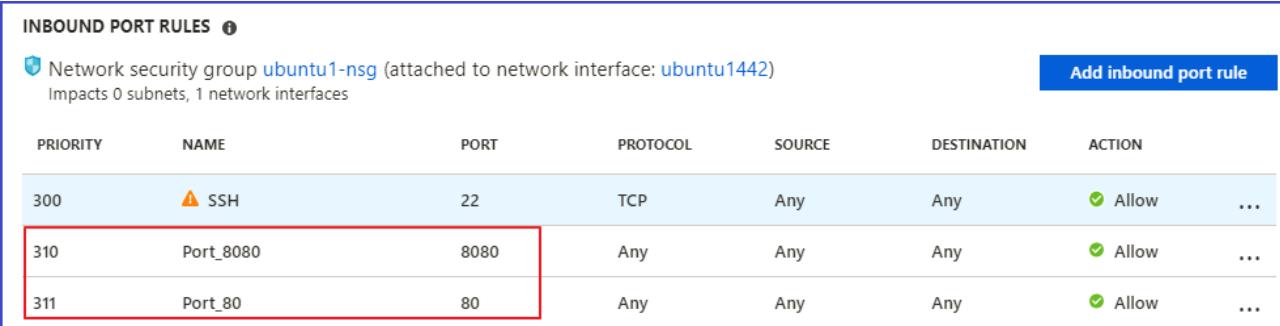
- nginx web service의 root directory로 사용할 Azure File Storage 생성한 후 연결하기
 - **docker volume create --name nginxweb -d jmaitrehenry/azurefile -o share=nginxrootdir**
 - **docker volume ls**

```
root@ubuntu1:~# docker volume ls
```

DRIVER	VOLUME NAME
jmaitrehenry/azurefile:latest	nginxweb
jmaitrehenry/azurefile:latest	vol1

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - 첫 번째 Docker host에서 작업하기
 - Azure VM의 Networking에서 80,8080을 접속하도록 허용한다



INBOUND PORT RULES ⓘ



Network security group **ubuntu1-nsg** (attached to network interface: **ubuntu1442**)
Impacts 0 subnets, 1 network interfaces

[Add inbound port rule](#)

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	
300	⚠ SSH	22	TCP	Any	Any	✔ Allow	...
310	Port_8080	8080	Any	Any	Any	✔ Allow	...
311	Port_80	80	Any	Any	Any	✔ Allow	...

- Docker Host로 실행 중인 CentOS Linux VM에서 SELinux 기능을 끈다
 - **vi /etc/sysconfig/selinux**
 - 여기서 enforcing을 **disabled**로 변경한 후 저장하고 빠져 나온다
 - **VM을 재시작하여** Selinux 기능을 켜지 않도록 한다
 - **reboot**
 - 다시 SSH로 Linux VM에 접속한다

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - 첫 번째 Docker host에서 작업하기
 - nginx라는 이미지를 실행하여 webapp8080이라는 container를 실행하되, container의 **nginx 홈디렉터리**를 **azure file storage**로 연결한다
 - nginxweb이라는 docker volume을 이미 생성해 두었다. 그 볼륨은 로컬 볼륨이 아닌 Azure File Storage에 있는 Volume이다
 - **docker run -d --name webapp8080 -p 8080:80 -v nginxweb:/usr/share/nginx/html nginx**
 - **docker exec webapp8080 ls -l /usr/share/nginx/html**
 - Azure File Storage에 nginxrootdir이라는 디렉터리 안에 2개의 파일이 생성되었는지 확인하기
- | NAME | TYPE | SIZE |
|--|------|-------|
|  50x.html | File | 537 B |
|  index.html | File | 612 B |
- 만약에 index.html 파일이 없으면 index.html 파일을 만들어서 Azure Storage에 업로드한다

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - 첫 번째 Docker host에서 작업하기
 - Web Browser를 사용하여 실행된 Container의 Web Site에 접속한다
 - `http://13.76.43.26:8080`



- 새로운 index.html 파일을 만들어서 기존의 Azure File Storage에 있는 index.html 파일을 대체한다
 - 그 결과가 변하는지 확인한다 (성공)



4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - 두 번째 Docker host에서 작업하기
 - **azurefile plugin 설치하면서 Azure File Storage에 로그인하기**
 - **docker plugin install jmaitrehenry/azurefile**
AZURE_STORAGE_ACCOUNT=ysleesk
AZURE_STORAGE_ACCOUNT_KEY=z0tBWNDp+agajc==
 - **docker container가 사용할 Azure File Storage의 volume 생성하기**
 - **docker volume create --name nginxweb -d jmaitrehenry/azurefile -o share=nginxrootdir**
 - Azure VM의 Networking에서 80,8080을 접속하도록 허용한다
 - CentOS Linux VM에서 **SELinux** 기능을 끈 후 재부팅한다. 그리고 다시 로그인한다
 - nginx라는 이미지를 실행하여 webapp8080이라는 container를 실행하되, container의 nginx 홈디렉토리를 azure file storage로 연결한다
 - **docker run -d --name webapp8080 -p 8080:80 -v nginxweb:/usr/share/nginx/html nginx**

4-Azure Storage를 Data Volume으로 사용하기

- Container Data Volume를 Azure File Storage로 사용하기
 - **두 번째** Docker host에서 작업하기
 - 로컬 노트북에서 Web Browser를 사용하여 실행된 Container의 Web Site에 접속한다
 - <http://13.76.43.26:8080> (##접속 성공하고, 첫번째 내용과 동일)
 - **결론:** Container의 Data Volume을 Azure File Storage로 사용하여 운영할 수 있다