

OCI FreeTrial에 Azeroth Core 서버 구축

2024년 05월 25일 by 수상한 길과

이번에는 'Mangos story'라는 네이버 카페를 통해 알게 된 Azeroth Core라는 World Of Warcraft 서버를 OCI FreeTrial에서 제공하는 평생 무료 VM에 구성한 내용을 정리해 보려 합니다. 해당 카페에도 글을 올려 두었지만 기록을 위해 개인 블로그에 조금 더 세세한 내용을 담아 구성 & 사용기를 남깁니다.

해당 카페의 게시글은 <https://cafe.naver.com/gostorm/50817>에서 확인이 가능합니다.

우선 'Mangos story'라는 카페의 글에 차운 글 <https://cafe.naver.com/gostorm/50659>을 참조하여 NI기반의 Azeroth Core 서버와 Client를 구성하였고 생각 이상으로 좋은 결과로 구현되어 있는 것을 확인 했습니다. 실제로 10여년 전으로 돌아가 어제로스를 잠깐 탐험해 보기도 했었구요. 헌데 좀 아쉬웠습니다. 게임서버는 소켓의 강점으로 인해 NT기반으로 많이 구성하는 것이 사실이지만 제 본업(?)과 관련된 Linux & Cloud에 서버를 옮겨 놓고 주변 인들을 해보고 싶었습니다. 그래서 만들었습니다. 몇 번의 좌절과 고통의 시간이 있었지만 잘 극복하고 현재 OCI Free Trial에서 제공받은 무료 VM에 서버 구성을 완료 한 상태입니다. 현재는 컨테이너 이미지로 구성들을 만들어 보는 중인데 사양 문제로 OCI Free Trial에서는 구현이 불가능 할 것 같습니다. 다만 컨테이너 작업은 마무리 할 예정입니다.

사족이 좀 길었는데 OCI Free Trial VM에 서버를 구성하려면 우선적으로 Swap 모드 구성이 필요합니다.

최초 VM이 표준이 되면 Swap 영역 없이 1GB의 메모리를 가지고 있는 상태인데 이 상태에서는 프로그램이 OS의 OOM Killer에게 접히 죽는 관계로 컴퓨터에 물을 정상적인 서버 구동도 불가능합니다.

Swap 모드는 root 유저에서 다음 명령으로 구성하였습니다.

1. Swap File 작성

```
1. dd if=/dev/zero of=/swap.img bs=1024 count=8000000
```

(8GB의 용량을 가지는 Swap 파일을 /swap.img 경로에 설정하는 과정입니다.)

2. 생성된 파일을 swap으로 사용하도록 지정

```
1. mkswap /swap.img
```

3. swap 파일 활성화

```
1. swapon /swap.img
```

4. 부팅 시 설정 적용을 위해 /etc/fstab 파일에 아래내용 추가

```
1. vi /etc/fstab
2. /swap.img swap swap defaults 0 0
```

추가 후 /etc/fstab 파일 내용입니다.

```
1. ubuntu@kami-vm-3:~$ cat /etc/fstab
2. # /etc/fstab: the canonical source for root's fs configuration
3. LABEL=WEFI /boot/efi vfat umask=0077 0 1
4. # /dev/sda1 This file was created/modifed by the Cloud Image build process
5. ##
6. # Oracle Infrastructure Customers
7. #
8. ## If you are adding an iSCSI remote block volume to this file you MUST
9. ## include the '_netdev' mount option or your instance will become
10. ## unavailable after the next reboot.
11. ## unavailability is not stable across reboots; please use the device UUID
12. ## instead of /dev path.
13. #
14. ## Examples:
15. ## UUID='94c5aae-b8b1-4d5c-a0dc-389bb8a716a' /data ext4 defaults,noatime,_netdev 0 2
16. ## More information:
17. ## https://docs.oracle.com/Content/Block/Tasks/connectingtovolume.htm
18. #
19. #/swap.img swap swap defaults 0 0
```

여기까지 swap 설정은 완료되었습니다.

다음으로 서버 개발환경 구성이 필요합니다.

1. apt update

```
1. sudo apt-get update
```

2. apt install

```
1. sudo apt full-upgrade -y && sudo apt install git cmake make gcc clang libssl-dev libbz2-dev libreadline-dev libcurl4-dev libboost-all-dev mariadb-server
mariadb-client libaio-dev libmariadb-dev-compat build-essential -y
```

여기까지 서버 개발환경 구성이 완료 되었습니다.

3. 소스 다운로드

다음으로 'Azeroth-Core' 소스코드를 Clone 해 줍니다. (저는 nproc bot이 추가된 trickerer님의 branch를 사용했습니다.)

https://github.com/trickerer/AzerothCore-wotlk-with-NPCBots/tree/npcbots_3.3.5

nproc bot이 추가되지 않은 'Azeroth-Core' 서버를 구성하시려면 아래 주소를 사용하시면 됩니다.

<https://github.com/azerothcore/azerothcore-wotlk>

```
1. git clone https://github.com/azerothcore/azerothcore-wotlk.git --branch master --single-branch azerothcore
```

4. 소스 컴파일 및 빌드

소스 코드 Clone이 완료되면 해당 디렉토리로 이동 후 build 디렉토리를 생성하고 이동 해 줍니다.

```
1. cd ~/trickerer/
2. mkdir build
3. cd build
```

소스 컴파일 및 빌드를 실행합니다. (OCI Free Trial VM은 서버 사양이 좋지 않아 이 과정에서 많은 시간이 소요됩니다. – 몇 시간 이상 소요될 수도 있습니다.)

```
1. cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/AzerothCore-wotlk-server/ -DCMAKE_C_COMPILER=/usr/bin/clang -DCMAKE_CXX_COMPILER=/usr/bin/clang++ -DMITH_WARNINGS=1 -DCCOLLECT_BUILDDATE -DSOURCES="realist"
```

컴파일 완료 후 빌드(install)을 진행합니다.

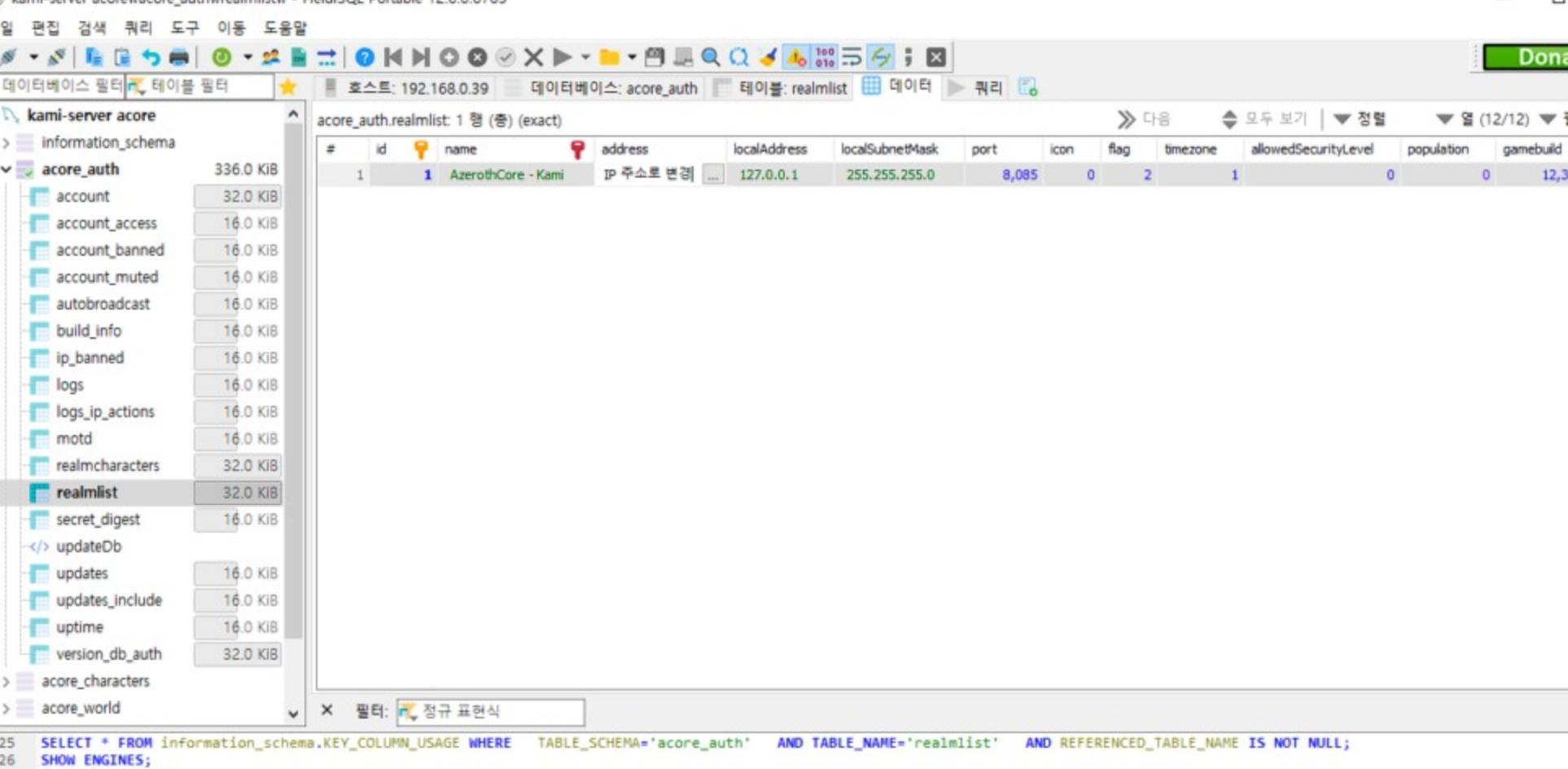
```
1. make -j 4 (nproc) install
```

(저는 빌드 실행을 위해 OCI Free Trial VM과 동일한 Ubuntu 22.04 서버를 별도의 고사양 VM으로 설치 후 swap 설정하고 해당 파일들을 OCI Free Trial VM으로 복사하였습니다. 리눅스에 대한 지식이 부족하신 분들은 시간이 걸리더라도 본 서버에서 컴파일 및 빌드 하는 것을 강력 추천드립니다. 경로, 권한 등의 문제가 있으므로 별도의 VM에서 작업하시는 경우 OCI Free Trial VM과 동일한 유저와 디렉토리 경로에서 수행해 주셔야 별도의 작업없이 파일 복사만으로 작업이 가능합니다.)

빌드가 마무리되면 \$HOME/home/ubuntu/.ubuntu 경로(OCI에 Ubuntu VM을 만드시면 ubuntu가 기본 유저입니다.) 아래에 azeroth-server라는 디렉토리가 생성되고 해당 디렉토리에 azeroth-core 서버가 설치 된 디렉토리입니다.

다음 과정으로 data 파일을 client에서 출처해야 하는데 저는 로아나님이 작성해 주신 글에 나와 있는 원도우 서버(C:\AzerothCore\data)의 data 디렉토리와 파일을 복사하였습니다. (참조 URL : <https://cafe.naver.com/gostorm/50659>)

아래 경로를 압축(data.tar.gz)하여 ftp 등의 방식으로 OCI VM으로 옮긴 후 /home/ubuntu/azeroth-server/data 경로에 압축을 풀어 주시면 됩니다.



5. 서버 설정

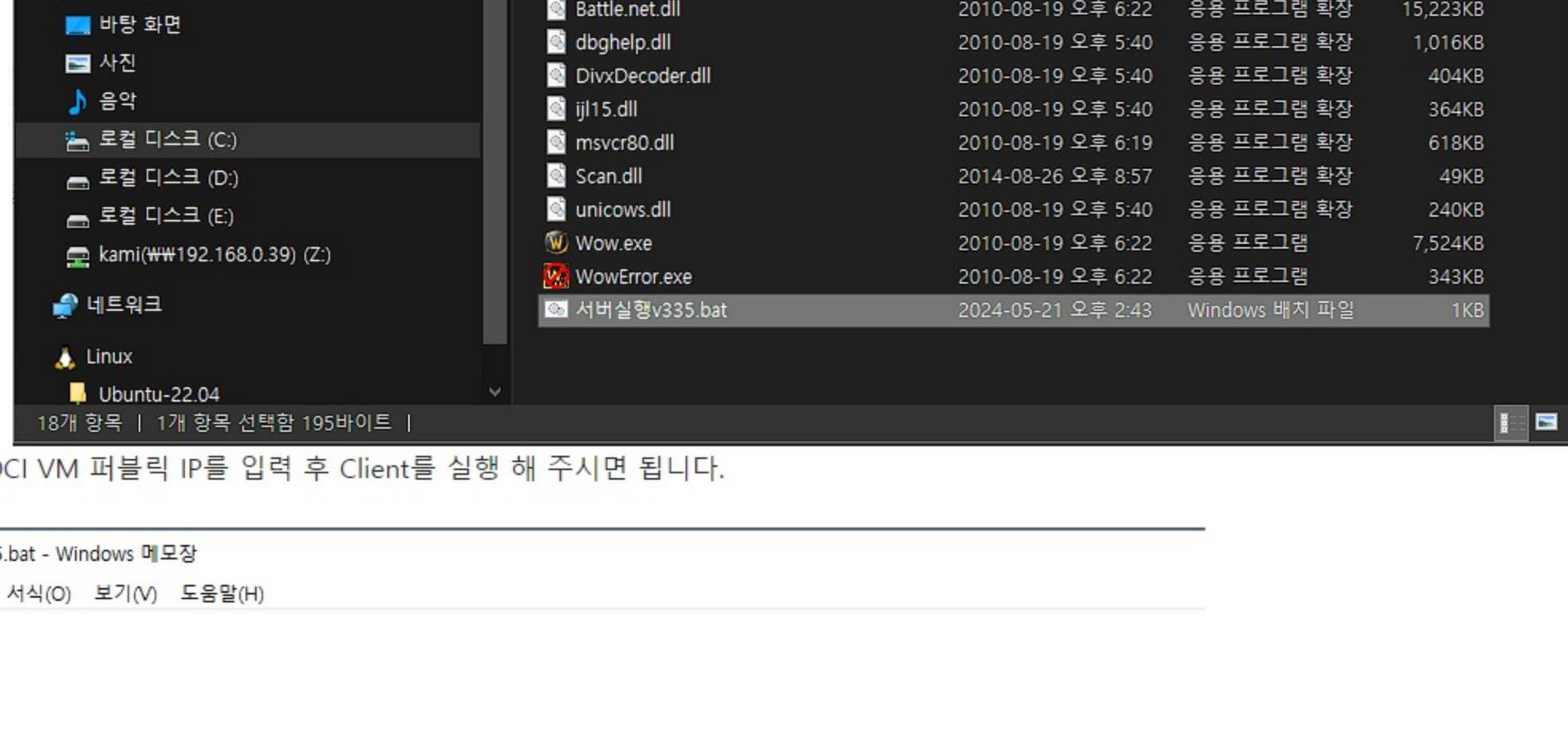
서버는 준비가 완료 되었고 conf 파일을 수정해 줍니다. (서버가 설치될 때 confdist의 확장자로 설치된 파일명을 conf로 변경 해 주고 위에서 작업한 data 파일의 경로를 설정하는 작업입니다.)

```
1. mv /home/ubuntu/azeroth-server/etc/worldserver.conf.dist /home/ubuntu/azeroth-server/etc/worldserver.conf
2. mv /home/ubuntu/azeroth-server/etc/worldserver.conf /home/ubuntu/azeroth-server/etc/worldserver.conf
```

확장자 변경 후 worldserver.conf 파일의 DataDir 경로를 설정해 줍니다. (vi, nano 등 익숙한 에디터로 worldserver.conf 파일을 열어 편집 후 저장합니다.)

```
1. vi /home/ubuntu/azeroth-server/etc/worldserver.conf
```

저와 동일한 과정을 진행하셨다면 worldserver.conf 파일의 DataDir 경로를 설정해 줍니다. 부분은 DataDir = '/home/ubuntu/azeroth-server/data' 과 같이 변경 해주시면 됩니다.



6. MariaDB 설정

서버 구성이 완료 되었으니 아래 명령을 입력하여 MariaDB 서버를 설정해 줍니다.

```
1. sudo mysql_secure_installation
```

화면을 보며 다음 명령을 차례로 입력하여 MariaDB를 초기화 해 줍니다.

```
1. [First option leave blank and hit enter]
2. [To switch to unix_socket authentication]
3. [Y to change the root password]
4. [Y to remove anonymous users]
5. [Y to remove test database]
6. [Y to remove test DB]
7. [Y to reload privilege tables now]
```

초기화 후 MariaDB에 접속하여 acore 계정을 설정합니다.

```
1. sudo mysql -u root -p
```

acore 계정과 데이터베이스를 생성하고 권한을 설정합니다.

```
1. DROP USER IF EXISTS 'acore'@'localhost';
2. CREATE USER 'acore'@'localhost' IDENTIFIED BY 'acore';
3. GRANT ALL PRIVILEGES ON *.* TO 'acore'@'localhost' WITH GRANT OPTION;
4. CREATE DATABASE `score_world` DEFAULT CHARACTER SET UTF8MB4 COLLATE utf8mb4_general_ci;
5. GRANT ALL PRIVILEGES ON `score_world`.* TO 'acore'@'localhost' WITH GRANT OPTION;
6. GRANT ALL PRIVILEGES ON `acore_characters`.* TO 'acore'@'localhost' WITH GRANT OPTION;
7. GRANT ALL PRIVILEGES ON `acore_auth`.* TO 'acore'@'localhost' WITH GRANT OPTION;
8. GRANT ALL PRIVILEGES ON `acore_auth_characters`.* TO 'acore'@'localhost' WITH GRANT OPTION;
9. GRANT ALL PRIVILEGES ON `acore_auth_realist`.* TO 'acore'@'localhost' WITH GRANT OPTION;
10. GRANT ALL PRIVILEGES ON `realist`.* TO 'acore'@'localhost' WITH GRANT OPTION;
11. FLUSH PRIVILEGES;
12. exit;
```

서버 구성이 완료 되었습니다. 데이터베이스 작성을 위해 다음 명령으로 서버를 실행해 줍니다.

```
1. sudo /home/ubuntu/azeroth-server/bin/authserver
2. sudo /home/ubuntu/azeroth-server/bin/worldserver
```

각각 서버가 포어 커맨드에 플로트 입력으로 실행되므로 authserver 실행 후 별도의 세션으로 접속하여 worldserver를 실행 해 주어야 정상적으로 게임 접속이 가능합니다.

(원도우 azerothcore 서버에서도 authserver worldserver 클릭 시 생성되는 CMD창을 닫으면 서버가 깨지는 것과 동일한 원리입니다.)

nonstop ./authserver & 등과 같은 방식으로 백그라운드 데몬으로 프로세스를 실행할 수도 있지만 관련 내용은 이번 포스팅에서는 다루지 않도록 하겠습니다.

서버가 실행된 후 데이터베이스 작성성이 되면 acore_auth 데이터베이스의 realist 테이블의 값을 OCI VM의 퍼블릭 IP로 변경 해 주어야 정상적으로 접속이 가능합니다.

autoserver와 worldserver를 종료한 후 hediSQL 등을 활용하여 아래와 같이 해당 값을 변경해 줍니다.

7. 서버실행 & 클라이언트 접속

```
1. sudo /home/ubuntu/azeroth-server/bin/authserver
2. sudo /home/ubuntu/azeroth-server/bin/worldserver
```

마지막으로 클라이언트에서 서버 접속 정보 수정 후 게임을 즐기시면 됩니다.

여기까지 OCI Free Trial을 활용하여 Azeroth-Core 서버를 구성하는 방법을 설명 드렸습니다.

주후 다른 버전의 WOW 서버도 구성 해 보면 재미있을 것 같네요. 컨테이너 버전도 조만간 구성 후 포스팅 진행 하겠습니다.

고맙습니다.

aci@kami-vm-8: ~\$./worldserver

aci@kami-vm-8: ~\$./authserver

aci@kami-vm-8: ~\$./realist

aci@kami-vm-8: ~\$./score

aci@kami-vm-8: ~\$./characters

aci@kami-vm-8: ~\$./logins

aci@kami-vm-8: ~\$./accounts

aci@kami-vm-8: ~\$./realist

aci@kami-vm-8: ~\$./realist