

DDNS Management : DDNS Updater

- [DDNS Updater 소개](#)
- [Cloudflare](#)

DDNS Updater 소개

가정에서 쓰는 IP는 거의 유동 IP입니다.

변경되는 IP를 DNS Provider에 자동 IP 갱신이 필요 할 경우 이 DDNS Updater를 이용하시면 됩니다.

Script가 돌때마다 IP변경 여부를 확인한 후 변경 시 DNS Server에 IP를 갱신해 줍니다.

----- Sample -----

작업 스케줄러가 예약 작업을 완료했습니다.

작업: DDNS Update

시작 시간: Sun, 25 Jun 2023 00:00:01 +0900

중지 시간: Sun, 25 Jun 2023 00:00:02 +0900

현재 상태: 0 (정상)

표준 출력/오류:

CloudFlare A Recoard Updater v1.0.9

Current IP: xxx.xxx.xxx.xxx

Previous IP: xxx.xxx.xxx.xxx

No need to update

Cloudflare

아래 스크립트로 sh 작성 후 스케줄러에 등록합니다.

계정관련 변수 값 부분만 사용자 환경에 맞게 수정해 주시면 됩니다.

```
#!/bin/bash
### CloudFlare A Recoard Updater
### A_Record : Separate the contents with commas (,)
### Proxied : true or false
### TTL : Between 120 and 2147483647 seconds, or 1 for automatic
### To force updating, run with -f
Login_Email=CLOUDFLARE_EMAIL_ID
Global_API_Key=CLOUDFLARE_GLOBAL_API_KEY
Domain=DOMAIN
A_Record=A_RECORD주소
Proxied=false
TTL=1
[ ! -f /var/tmp/ip.txt ] && touch /var/tmp/ip.txt
CIP=$(curl -s "https://ipv4.icanhazip.com/")
PIP=$(cat /var/tmp/ip.txt)
echo -e "CloudFlare A Recoard Updater v1.0.9"
echo -e "Current IP: $CIP"
echo -e "Previous IP: $PIP"
if [ "$CIP" == "$PIP" ] && [[ $1 != "-f" ]]; then
    echo "No need to update"; exit 0
elif [[ $1 == "-f" ]]; then
    echo "Force updating A recoard....."
elif [ "$CIP" != "$PIP" ]; then
    echo "Updating A recoard....."
fi
echo $CIP > /var/tmp/ip.txt
V4="https://api.cloudflare.com/client/v4/zones"
H1="-HX-Auth-Email:$Login_Email"
H2="-HX-Auth-Key:$Global_API_Key"
H3="-HContent-Type:application/json"
ZN=$(curl -s -X GET "$V4?name=$Domain" \
    $H1 $H2 $H3 | grep -Po '(?<="id:")[^"]*' | head -1)
string=$A_Record
IFS=' ' ARARY=(${string})
function AID() {
    for AREC in "${ARARY[@]}"
    do
        (curl -s -X GET "$V4/$ZN/dns_records?name=$AREC" \
            $H1 $H2 $H3 | grep -Po '(?<="id:")[^"]*' | head -1)
    done
}
```

```
IFS=$'\n' AIDARY=$(AID)
ATOTAL=${#ARARY[*]}
for ((i=0; i<$ATOTAL; i++))
do
  (curl -s -X PUT "$V4/$ZN/dns_records/${AIDARY[$i]}" $H1 $H2 $H3 \
  --data "{\"type\":\"A\",\"name\":\"${ARARY[$i]}\",\"content\":\"$CIP\",\"proxied\":\"$Proxied\",\"ttl\":\"$TTL}\" \
  | grep -Po '(?<="name:")[^"]*(?<="content:")[^"]*(?<=Z"},)[^"]*(?<="success":false,)[^"]*(?<=\s\s)[^"]*' |
xargs)
done
```