Route 53에 도메인 등록하기 I

1. **[서비스]** > **[네트워킹 및 콘텐츠 전송]** > **[Route 53]**를 선택한다.

2. **Route 53 대시보드** 페이지에 들어왔다. 좌측 메뉴 중 **[도메인]** > **[등록된 도메인]**을 클릭한다.

3. **[등록된 도메인]** 페이지이다. **[도메인 등록]**은 새로운 도메인을 등록하는 데 사용한다. **[도메인 등록]**을 클릭한다.

4. 먼저 등록하고 싶은 도메인을 검색부터 해야 한다. 누군가 이미 사용하고 있다면 당연히 등록할 수 없다. 아래 그림과 같아 **myaws.com**으로 검색하면 사용할 수 없고, AWS가 추천하는 다른 도메인 목록이 나온다.

5. 이번에는 **myaws.world** 로 검색했더니 사용 가능이라고 확인할 수 있다. **[장바구니에 추가]**를 클릭한다. **[장바구니]** 섹션에 1**년에 12달러**의 금액이 필요하다고 나온다. 확인 후 **[결제 진행]**을 클릭하여 다음 단계로 이동한다.

6. 다음 단계로 **[1단계:요금]** 페이지이다. 가격을 확인하고 **[다음]** 버튼을 클릭한다. [자동 갱신]이 기본적으로 체크되어있지만, 본 예제에서는 체크 해제하기로 한다.

7. **[2단계:연락처 정보]** 페이지이다. 각각의 값을 입력한다.

8. 각 항목의 값을 입력 후 페이지 하단의 **[계속]**을 클릭한다.

9. 다음 단계는 **[3단계:검토 및 제출]** 페이지이다. **[이용 약관]** 항목을 체크하고 **[제출]**를 클릭한다.

10. **[다음 도메인에 대한 등록 요청이 진행 중입니다. 각 도메인의 등록 상태 상태에 관한 이메일을 받게 됩니다.]** 메시지를 확인할 수 있다.

11. **[상태 확인]**버튼을 클릭하면 다음과 같이 **진행 중**으로 확인된다.

12. 등록한 이메일을 확인해 보면 다음과 같이 등록이 성공했음을 확인할 수 있다.

13. 도메인 등록이 완료되었다.

14. 등록 완료된 도메인의 **[도메인 이름]**을 클릭해서 상세 페이지로 들어간다.

Freenom에 무료 도메인 생성하기

현재 freenom 사이트에서 도메인 발급은 기술적 이유로 사용할 수 없다고 사이트에서 공지하고 있다. 따라서 향후 기술적 문제가 해결되면 다시 서비스를 받을 수 있을 것으로 기대한다. 지금은 단지 참고만 한다.

1. 무료 도메인을 발급 받기 위해 freenom 사이트(https://www.freenom.com/en/index.html?lang=en)를 방문한다. 검색창에 도메인을 입력하고, 검색하기 위해 **[Check Availability]** 파란색 원을 클릭한다. 검색 결과에서 입력한 도메인을 사용하기 위해 해당 도메인의 **[Get it now!]**를 선택한 후, **[Selected]** 초록색 버튼을 확인하고, 상단의 **[Checkout]** 초록색 버튼을 클릭한다.

2. 기본이 3개월 무료이고, 최대 12개월까지 무료로 사용할 수 있다. 12개월을 선택하고 **[Continue]** 파란색 버튼을 클릭한다.

3. 금액은 무료이기 때문에 **0$**이다. 만일 로그인을 하지 않았다면 로그인한다. 구글, Facebook 가능하다. **[Complete Order]**를 클릭하여 주문을 완성한다.

4. 주문이 마무리되었다. 등록한 메일로 인증 메일이 올 수 있다. 페이지 아래의 **[Click here to go to your Client Area]** 파란색 버튼을 클릭한다.

5. 방금 등록 완료된 도메인 정보를 확인할 수 있다. 혹은 로그인 후, 상단의 메뉴 **[Services]** > **[My Domains]**를 선택해도 된다.

Route 53을 이용한 도메인 네임 서버 생성

이 부분도 freenon 사이트가 정상화되면 가능할 것이다.

1. 이번 실습은 **Route 53**에 도메인을 가지고 있는 상태에서 **Route 53**을 네임 서버로 이용해서 어떤 특정 도메인과 연결하는 방법을 살펴보자. 좌측 메뉴 중 **[호스팅 영역]**을 클릭한다. 이미 위 실습에서 등록했던 도메인이 보인다. 선택하여 **[삭제]** 버튼을 클릭하여 **Freenom**에서 등록한 도메인으로 실습을 계속 진행하기로 한다.

2. **[호스팅 영역...을 삭제하시겠습니까?]** 창에서 **삭제**라고 입력하고 **[삭제]** 버튼을 클릭한다.

3. **[호스팅 영역 생성]**을 클릭한다.

4. **[호스팅 영역 생성]** 페이지에서 **[도메인 이름]**은 **Freenom.com**에서 생성한 도메인을 넣는다.

5. 페이지를 아래로 스크롤다운하여 **[태그]** 섹션에 **[키]**는 **Name**으로, **[값]**은 **lab-hosting**이라고 입력하고 [**호스팅 영역 생성]**을 클릭한다.

6. 다음과 같이 네임서버가 생성되었다.

7. 이번에는 네임 서버를 등록해 보자. 다시 **Freenom.com**의 **My Domain** 페이지로 이동한다. **[Manage Domain]** 옆의 톱니바퀴 아이콘을 클릭한다.

8. 또는 등록한 도메인의 **[Management Tools]** > **[Nameservers]** 메뉴를 선택한다.

9. 다음 그림과 같은 페이지로 이동한다. 현재 **[Use default...]**로 선택되어 있는데, 방금 생성한 도메인의 관리를 **Freenom.com**이 아닌 **AWS Route 53** 서비스가 대행할 예정이기 때문에 **[Use custom nameservers (enter below)]**를 선택한다.

10. **AWS Route 53** 페이지에서 등록한 도메인의 **NS** 유형의 값 4개를 복사해서 **Freedon.com**에 붙여넣는다.

11. 여기서 주의할 점은 반드시 **[유형]**이 **NS**의 값만 복사해서 넣어야 한다. 또한 마지막 **.**은 붙여넣지 않는다. 입력을 마치면 **[Change Nameservers]** 파란색 버튼을 클릭한다.

12. 새로 등록한 네임서버 정보가 전체 DNS 서버들을 통해 업데이트하는데는 24시간 정도의 시간이 필요하다.

DNS 디버그

1. **Dig**이라는 도구를 통해 DNS 설정을 확인하는 실습을 한다. **Dig**은 Windows를 제외한 다른 OS에서 사용할 수 있는 명령어이다. Windows에서는 다음과 같은 과정을 통해 사용할 수 있다. 웹 브라우저를 열고 다음의 URL을 입력하여 이동한다. **[Hostnames or IP addresses:]** 항목에 등록 테스트할 도메인을 입력하고, **[Options]**에서는 **Colorize output**, **Trace**를 체크하고 **[Dig]** 버튼을 클릭한다.

   http://www.digwebinterface.com

2. 다음과 같은 결과를 얻을 수 있다. 만일 OS가 macOS이라면 **Terminal**에서 다음과 같은 명령을 사용한다.

   $ dig +trace {domain name}

```
myaws.world@8.8.4.4 (Default):

.                       34014   IN      NS      a.root-servers.net.
.                       34014   IN      NS      b.root-servers.net.
.                       34014   IN      NS      c.root-servers.net.
.                       34014   IN      NS      d.root-servers.net.
.                       34014   IN      NS      e.root-servers.net.
.                       34014   IN      NS      f.root-servers.net.
.                       34014   IN      NS      g.root-servers.net.
.                       34014   IN      NS      h.root-servers.net.
.                       34014   IN      NS      i.root-servers.net.
.                       34014   IN      NS      j.root-servers.net.
.                       34014   IN      NS      k.root-servers.net.
.                       34014   IN      NS      l.root-servers.net.
.                       34014   IN      NS      m.root-servers.net.
;; Received 228 bytes from 8.8.4.4#53(8.8.4.4) in 24 ms

world.                  172800  IN      NS      v0n2.nic.world.
world.                  172800  IN      NS      v2n0.nic.world.
world.                  172800  IN      NS      v0n1.nic.world.
world.                  172800  IN      NS      v2n1.nic.world.
world.                  172800  IN      NS      v0n0.nic.world.
world.                  172800  IN      NS      v0n3.nic.world.
;; Received 416 bytes from 202.12.27.33#53(202.12.27.33) in 674 ms

myaws.world.            3600    IN      NS      ns-181.awsdns-22.com.
myaws.world.            3600    IN      NS      ns-1432.awsdns-51.org.
myaws.world.            3600    IN      NS      ns-580.awsdns-08.net.
myaws.world.            3600    IN      NS      ns-1638.awsdns-12.co.uk.
;; Received 169 bytes from 65.22.25.1#53(65.22.25.1) in 28 ms

myaws.world.            900     IN      SOA     ns-580.awsdns-08.net. awsdns-hostmaster.amazon.com. 1 7200
;; Received 113 bytes from 205.251.192.181#53(205.251.192.181) in 2 ms
```

Route 53 실습을 위한 VPC 설정하기

1. 다음과 같이 **VPC**를 생성한다.

① **[생성할 리소스]** : VPC 등

② **[이름 태그 자동 생성]** : 자동 생성 체크, lab

③ **[IPv4 CIDR 블록]** : 10.0.0.0/16

④ **[가용 영역]** : 2 / ap-northeast-2a, ap-northeast-2c

⑤ **[퍼블릭 서스넷 수/프라이빗 서브넷 수]** : 각 2개

i. **[ap-northeast-2a 퍼블릭 서브넷 CIDR]** : 10.0.10.0/24

ii. **[ap-northeast-2c 퍼블릭 서브넷 CIDR]** : 10.0.20.0/24

iii. **[ap-northeast-2a 프라이빗 서브넷 CIDR]** : 10.0.30.0/24

iv. **[ap-northeast-2c 프라이빗 서브넷 CIDR]** : 10.0.40.0/24

⑥ **[NAT 게이트웨이]** : AZ당 1개

⑦ **[DNS 옵션]** : **[DNS 호스트 이름 활성화]** 체크/ **[DNS 확인 활성화]** 체크

2. 다음과 같이 보안 그룹을 생성한다.

① **[보안 그룹 이름]** : lab-webserver-sg

② **[설명]** : Security group for Webserver

③ **[VPC]** : lab-vpc

④ **[인바운드 규칙]**

i. SSH, TCP, 22, Anywhere-IPv4, 0.0.0.0/0

ii. HTTP, TCP, 80, Anywhere-IPv4, 0.0.0.0/0

⑤ **[태그]** > **[새로운 태그 추가]**

i. **[키]** : Name

ii. **[값]** : lab-webserver-sg

3. 다음과 같이 **WebServer(Based on Ubuntu)** 인스턴스를 생성한다.

① **[이름]** : lab-ubuntu-ec2

② **[AMI]** : Ubuntu Server 22.04 LTS (HVM), SSD Volume Type, 64비트(x86)

③ **[인스턴스 유형 선택]** : t2.micro

④ **[키 페어(로그인)]** > **[새 키 페어 생성]** : lab-ubuntu-ec2-key.pem

⑤ **[네트워크 설정]** > **[편집]**

① **[네트워크]** : lab-vpc

② **[서브넷]** : lab-subnet-public1-ap-northeast-2a

③ **[퍼블릭 IP 자동 할당]** : 활성화

⑥ **[방화벽(보안그룹)]** > **[기존 보안 그룹 선택]**

⑦ **[일반 보안 그룹]** : lab-webserver-sg

⑧ **[스토리지 구성]** : 8 GiB, gp2

4. 생성한 EC2 Instance를 Tabby 같은 SSH Client Tool을 통해 접속한다.

5. 다음과 같이 Ubuntu 기반 Apache Webserver를 설치한다.

① $ sudo apt update

② $ sudo apt install apache2

③ $ sudo systemctl status apache2

```
ubuntu@ip-10-0-10-205:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-10-21 01:04:09 UTC; 16s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2214 (apache2)
      Tasks: 55 (limit: 1121)
     Memory: 4.9M
        CPU: 32ms
     CGroup: /system.slice/apache2.service
             ├─2214 /usr/sbin/apache2 -k start
             ├─2216 /usr/sbin/apache2 -k start
             └─2217 /usr/sbin/apache2 -k start

Oct 21 01:04:09 ip-10-0-10-205 systemd[1]: Starting The Apache HTTP Server...
Oct 21 01:04:09 ip-10-0-10-205 systemd[1]: Started The Apache HTTP Server.
```

6. Webserver의 IP로 브라우저에서 연결해서 확인한다.

7. 다음과 같이 index.html을 수정한다.

① $ sudo su

② # nano /var/www/html/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome to My Homepage.</title>
</head>
<body>
    <center>
        <h1>Hello, Ubuntu Webserver!!!</h1>
        <img src='./apples.jpg' />
    </center>
</body>
</html>
```

8. 웹 페이지에 이미지를 렌더링하기 위해 다음의 경로에서 apples.jpg를 /var/www/html 디렉토리에 다운로드 한다.

① # cd /var/www/hrtml

② # wget https://raw.githubusercontent.com/swacademy/AWS-reStart/main/30-DAY/apples.jpg

```
root@ip-10-0-10-205:/var/www/html# wget https://raw.githubusercontent.com/swacademy/AWS-reStart/main/30-DAY/apples.jpg
--2023-10-21 01:15:35--  https://raw.githubusercontent.com/swacademy/AWS-reStart/main/30-DAY/apples.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 49808 (49K) [image/jpeg]
Saving to: ‘apples.jpg’

apples.jpg                      100%[===================================================>]  48.64K  --.-KB/s    in 0.002s

2023-10-21 01:15:35 (29.7 MB/s) - ‘apples.jpg’ saved [49808/49808]
```

9. 수정된 웹페이지를 브라우저에서 확인한다.

도메인 이름과 IP 주소 연결 – A record

1. AWS에서 **[서비스]** > **[네트워킹 및 콘텐츠 전송]** > **[Route 53]**으로 이동한다.

2. 좌측 메뉴의 **[호스팅 영역]**을 클릭하여 페이지를 이동한다.

3. 이번 실습에서는 필자가 미리 도메인 신청한 myaws.world로 진행하기로 한다. 해당 도메인을 클릭한다. 여기서 **[레코드 생성]** 버튼을 클릭한다.

4. 다음과 같이 **[빠른 레코드 생성]** 페이지에서, **A record**를 생성한다.

   ① **[레코드 이름]** : 그냥 myaws.world 그대로 사용한다.

   ② **[레코드 유형]** : **A-IPv4 주소 및 일부 AWS 리소스로 트래픽 라우팅**

   ③ **[값]** : EC2 Instance(현재 Ubuntu)의 퍼블릭 IP

   ④ **[레코드 생성]** 버튼 클릭

5. **myaws.world에 대한 레코드가 생성되었습니다.** 메시지박스안에 있는 **[상태 보기]** 버튼을 클릭해보자.

6. 약 1분 동안 전 세계의 DNS 서버의 복제작업 때문에 바로 서비스를 이용할 수 없다. 현재 **[상태]**가 PENDING 이기 때문이다.

7. 일정시간이 지나면 **[상태]**가 INSYNC임을 확인할 수 있다.

8. 등록된 도메인 이름으로 다시 브라우저에서 그리고 터미널에서 확인한다.

```
C:\Users\MZC01-HENRY>nslookup
기본 서버:  kns.kornet.net
Address:  168.126.63.1

> myaws.world
서버:    kns.kornet.net
Address:  168.126.63.1

권한 없는 응답:
이름:    myaws.world
Address:  3.36.123.70
```

9. **정리하면, A 레코드는 도메인 이름을 IP와 바인딩하는 레코드 유형이다. 해당 도메인 이름으로 DNS Query를 하면 결과는 연결된 Webserver IP이다.**

S3 Bucket의 정적 웹사이트 호스팅을 Route 53에 등록하기

1. 이번에는 Route 53의 Alias 즉 별칭을 이용한 레코드 등록을 한다. 대상은 S3 Bucket 정적 웹사이트 호스팅이다. 자세한 설정은 **Amazon S3 버킷의 정적 웹 사이트에 도메인 사용.pdf**를 참고한다.

2. 다음과 같이 버킷을 생성한다.

   ① **[버킷 이름]** : blog.myaws.world ← *반드시 등록하려는 도메인 이름과 동일해야 한다.*

   ② **[AWS 리전]** : 아시아 태평양(서울) ap-northeast-2

   ③ **[객체 소유권]** : **[ACL 활성화됨]**

   ④ **[객체 소유권]** : 버킷 소유자 선호

   ⑤ **[이 버킷의 퍼블릭 액세스 차단 설정]** > **[모든 퍼블릭 액세스 차단]** 모두 해제 후, [**새 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단]**과 **[임의의 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 및 교차 계정 액세스 차단]**만 체크

   ⑥ **[현재 설정으로 인해 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있음을 알고 있습니다.]** 체크

⑦ 나머지 설정은 기본값으로 한다. 생성된 버킷 상세페이지에서 **[권한]** 탭으로 이동한다. **[ACL(액세스 제어 목록)]** 섹션에서 **[편집]** 버튼을 클릭하여 **[모든 사람(퍼블릭 액세스)** 그룹의 권한을 **읽기** 권한을 부여한다. 체크박스에 체크하고 **[변경 사항 저장]**을 클릭한다.

⑧ **[속성]** 탭으로 이동하여 페이지 제일 아래에 위치한 **[정적 웹 사이트 호스팅]** 섹션에서 **[편집]**을 클릭한다.

⑨ **[정적 웹 사이트 호스팅 편집]** 페이지에서 다음과 같이 설정하고 **[변경 사항 저장]** 버튼을 클릭한다.

i. **[정적 웹 사이트 호스팅]** : 활성화

ii. **[호스팅 유형]** : 정적 웹 사이트 호스팅

iii. **[인덱스 문서]** : index.html

3. 다음과 같이 index.html 파일을 준비한다.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome to My Homepage.</title>
</head>
<body>
    <center>
        <h1>Hello, Amazon Simple Storage Service Homepage!!!</h1>
        <img src='./lemon.jpg' />
    </center>
</body>
</html>
```

4. 그리고 필요한 이미지도 준비한다. 버킷의 **[객체]** 탭으로 이동한다. **[업로드]** 버튼을 클릭한다. 그리고 **[파일 추가]** 버튼을 클릭하여, 위에서 생성한 index.html과 이미지를 선택한다. **[권한]** > **[ACL(액세스 제어 목록)]** 섹션에서 **[미리 정의된 ACL]**을 **퍼블릭 읽기 액세스 권한 부여**를 선택한다. 그리고 체크 박스에 체크한 후, **[업로드]** 버튼을 클릭한다.

5. 정상적으로 파일이 업로드됐다.

6. Route 53 페이지로 이동하여, **[호스팅 영역]**에서 등록한 도메인 이름을 클릭하여 **[레코드 생성]** 버튼을 클릭하여 새 레코드를 등록한다.

7. 다음과 같이 별칭으로 새 레코드를 저장한다.

① **[레코드 이름]** : blog.{도메인 이름}

② **[레코드 유형]** : **A-IPv4 주소 및 일부 AWS 리소스로 트래픽 라우팅**

③ **[별칭]** 사용

④ **[트래픽 라우팅 대상]** : **[S3 웹 사이트 엔드포인트에 대한 별칭]**

⑤ **[리전 선택]** : 아시아 태평양(서울)

⑥ **[S3 엔드포인트 입력]** > 마우스 선택 > 위에서 생성한 버킷 선택

8. 아래와 같이 새 레코드가 잘 등록되었다. 상단의 메시지의 **[상태 보기]** 버튼을 클릭하여 현재 **[상태]**를 확인한다.

9. **[상태]**가 **INSYNC**가 되면 서비스가 가능하다는 것이다. 브라우저에서 확인해보자.