06

GS Neotek

AWS GETTING STARTED

www.wisen.co.kr

WiseN

Wisely Combine the Network platforms

AWS GETTING STARTED

Contents

1. EC2 Getting Started

EC2(Elastic Compute Cloud)는 컴퓨팅 파워를 자유롭게 변경할 수 있는 일종의 가상 서버입니다. 용도에 따라 CPU/메모리사양을 선택할 수 있고, 생성시 사용하는 이미지에 따라 OS/미들웨어/어플리케이션 또한 자유롭게 선택하여 사용할 수 있습니다.

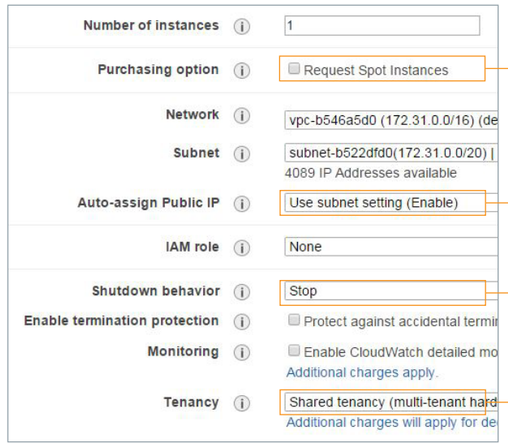
1.1. EC2 인스턴스 생성

STEP 0 **Launch Instance**
인스턴스를 생성할 Region 선택 후,
[Console] > [EC2] > [Launch Instance]

STEP 1 **Choose an Amazon Machine Image (AMI)**
인스턴스를 실행하는데 필요한 소프트웨어 구성(운영체제, 서버 어플리케이션 및 기타 비즈니스 솔루션)이 포함된 AMI를 선택합니다.

STEP 2 **Choose an Instance Type**
다양한 하드웨어(CPU, 메모리, 스토리지 등)로 조합된 인스턴스를 선택합니다.

STEP 3 **Configure Instance Details**
인스턴스 세부항목을 선택합니다.



체크시
스팟 인스턴스 구매옵션이 활성화되어 최대가격, 요청시간 등을 설정 가능

활성화시
인스턴스에 공인IP가 할당됨

인스턴스 내부에서 shutdown 했을때
Stop 선택시 OS만 종료
Terminate 선택시
종료 후 인스턴스가 삭제됨

공유 인스턴스(Shared)와
전용 인스턴스(Dedicated) 중 선택

STEP 4 **Add Storage**
인스턴스에 장착될 스토리지를 설정합니다.
필요시 추가적인 EBS볼륨을 생성/장착할 수 있습니다.

Tag Instance
Key-Value 형식으로 인스턴스의 태그를 생성합니다.

STEP 6 **Configure Security Group**
Security Group을 설정합니다. 기존에 설정된 Security Group을 선택할 수 있으며, 새로운 규칙(Inbound/Outbound Rules)을 추가할 수도 있습니다.

STEP 7 **Review Instance Launch**
앞에서 설정한 항목들을 확인하고 수정할 수 있습니다. [Launch] 버튼을 클릭하여 인스턴스 접속에 사용할 키 쌍(Key Pair)을 설정합니다. 기존에 생성한 키 쌍 또는 새로운 키 쌍을 선택한 후 다운로드가 완료되면 [Launch Instances] 버튼이 활성화되며, 해당 버튼을 클릭하여 인스턴스를 생성할 수 있습니다.

1.2. EC2 Linux 인스턴스 접속

Linux 계열의 인스턴스에 접속하기 위해서는 다음 3가지 사항을 확인해야 합니다.

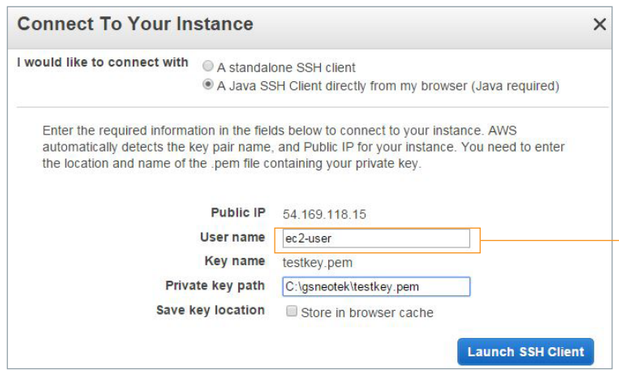
- 인스턴스의 Public DNS
- 인스턴스에 등록된 키 쌍(또는 키 쌍의 정보를 담은 (pem파일)
- 접속할 시스템의 IP로부터 SSH접속 허용 (Security Group 설정 등)

[브라우저에서 접속]

브라우저에서 인스턴스에 바로 접속하기 위해서는 Java가 설치되어 있어야 합니다.

STEP 1 [Console] > [EC2] > [Instances] > [Connect]

STEP 2 **“A Java SSH Client directly from my browser”를 선택 후 Username 입력, pem파일 지정**



선택한 AMI의 OS에 따라 Username이 결정됨
Amazon : ec2-user
Red Hat : ec2-user
Ubuntu : ubuntu
Fedora : Fedora
SUSE : root

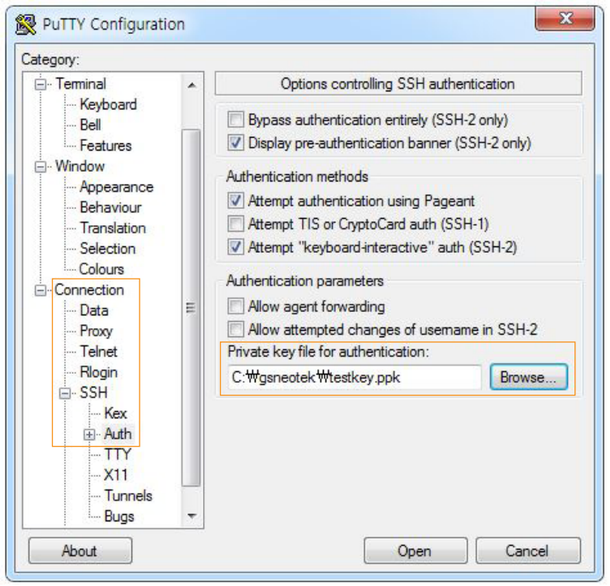
STEP 3 **[Launch SSH Client] 버튼을 클릭**
MindTerm 클라이언트 창이 실행되면서 인스턴스에 접속됩니다.

[PuTTY를 사용하여 접속]

PuTTY를 사용하여 인스턴스에 접속하기 위해서는 사전에 PuTTY Key Generator를 사용하여 pem파일을 ppk파일로 변경해야 합니다.

STEP 1 **PuTTY를 실행하고 Host Name에 접속할 EC2의 Public DNS를 입력**

STEP 2 **[Connection] > [SSH] > [Auth] > [Browse…]을 선택하여 생성한 ppk파일의 경로를 지정**



STEP 3 **[Open] 버튼을 클릭하여 인스턴스에 접속**
접속 후 AMI의 OS에 맞는 Username을 입력하여 로그인합니다.

[Mac OS, Linux에서 접속]

STEP 1 **Mac OS나 Linux에서 인스턴스에 최초 접속시 pem파일의 권한을 변경**
$ chmod 400 /gsneotek/testkey.pem
(pem파일경로)

STEP 2 **다음 명령어로 인스턴스에 접속**
$ ssh –i /gsneotek/testkey.pem ec2-user@54.169.118.15
(pem파일경로) (Username) (Public IP)

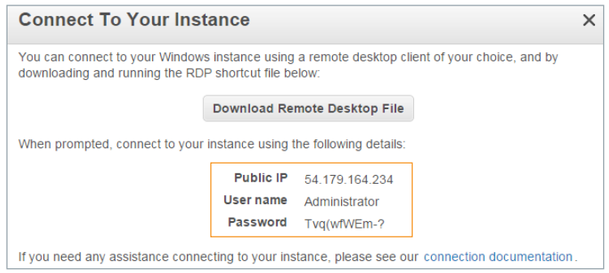
1.3. EC2 Windows 인스턴스 접속

Windows 계열의 인스턴스에 접속하기 위해서는 다음 2가지 사항을 확인해야 합니다.

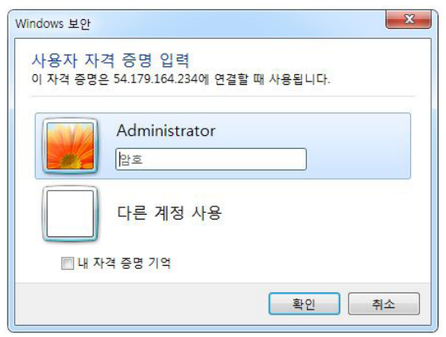
- 인스턴스에 등록된 키 쌍의 정보를 담은 pem파일
- 접속할 시스템의 IP로부터 RDP접속 허용

STEP 1 **[Console] > [EC2] > [Instances] > [Connect]**

STEP 2 **[Get Password] 버튼을 클릭하여 저장된 pem파일을 지정한 후, [Decrypt Password] 버튼을 클릭하여 패스워드를 발급**



STEP 3 **확인된 계정 정보를 이용하여 RDP접속**

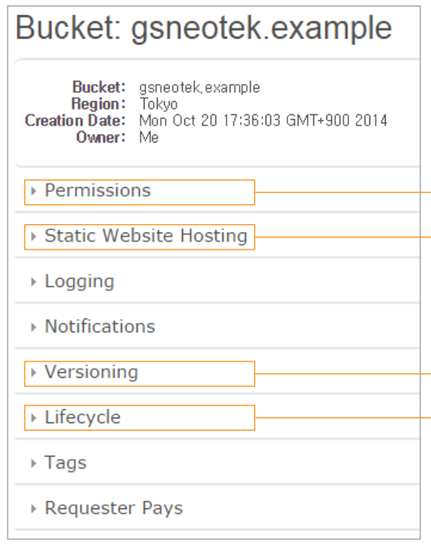


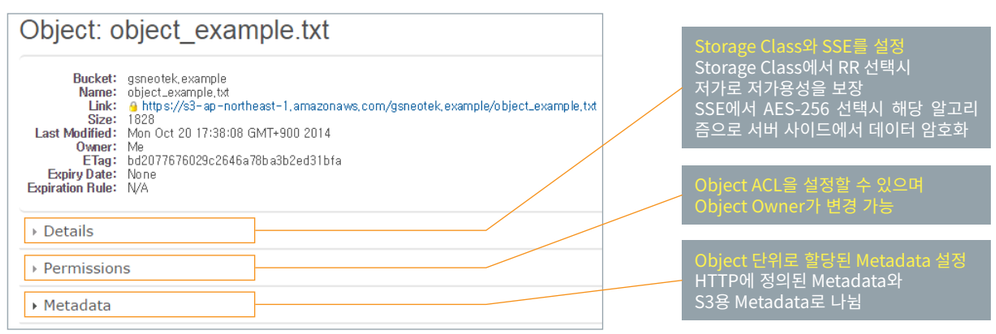
2. S3 Getting Started

S3(Simple Storage Service)는 AWS에서 제공하는 인터넷 스토리지 서비스입니다. 용량제한이 없으며 높은 내구성과 가용성을 보증합니다. 정적 웹사이트 호스팅 기능을 사용하여 정적 웹서비스가 가능하며, Versioning/Lifecycle 등의 부가 기능을 제공합니다.

2.1. S3 Bucket & Object

S3에서 사용되는 용어로는 버킷(bucket)과 객체(Object)가 있습니다. 버킷은 일종의 루트 폴더와 같은 개념의 저장공간 구분단위로 고유한 이름을 가지며 먼저 버킷을 생성한 후 하위에 객체를 업로드하거나 다운로드하게 됩니다. 버킷과 객체 모두 단위별로 속성(Property)값을 갖고 있으며, 아래에서는 버킷과 객체별 속성에 대해 설명합니다.

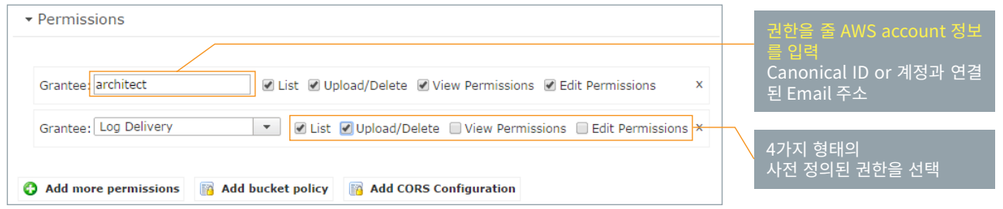




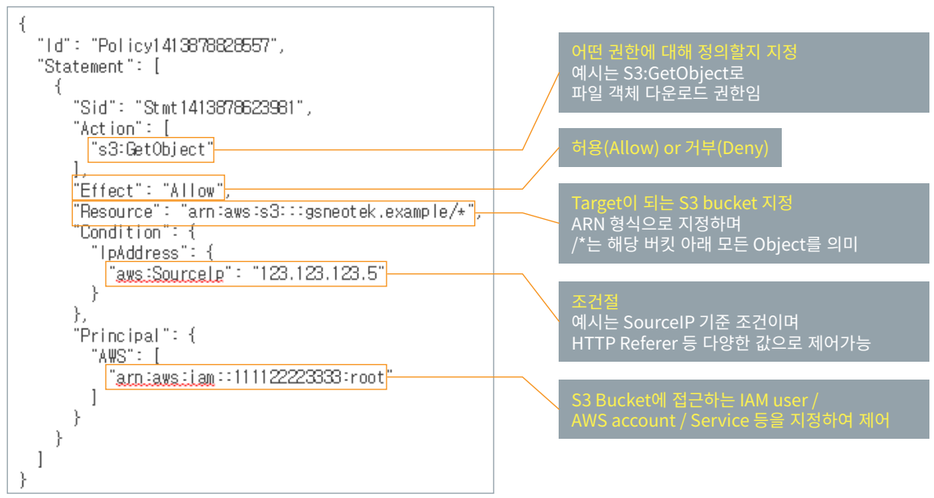
2.2. S3 Bucket Access Policy

버킷(Bucket) 단위의 접근 정책(Access Policy)에는 두가지가 있는데 하나는 Bucket ACL이고 다른 하나는 Bucket Policy입니다. 반면 객체(Object)의 경우에도 별도의 Object ACL이 존재하나, Bucket Owner와 Object Owner가 동일할 경우엔 Bucket Policy의 문맥을 따릅니다. 여기서는 Bucket ACL과 Bucket Policy에 대해 설명합니다.

Bucket ACL은 IAM 개념이 나오기 전에 사용되었던 옛 방식입니다. 따라서 Action의 종류나 Condition등을 세밀하게 정의할 수 없으며 정책의 개수도 상대적으로 제한적입니다. 따라서 보통 버킷 단위의 접근 정책은 Bucket Policy를 사용하여 제어합니다.



Bucket Policy는 아래와 같이 JSON 형식으로 쓰여집니다. JSON 형식을 잘 모르는 경우, AWS Policy Generator (http://awspolicygen.s3.amazonaws.com/policygen.html)페이지에서 원하는 조건을 입력하면 JSON 형식의 Statement를 생성할 수 있습니다. 각 항목의 쓰임새는 아래와 같습니다.



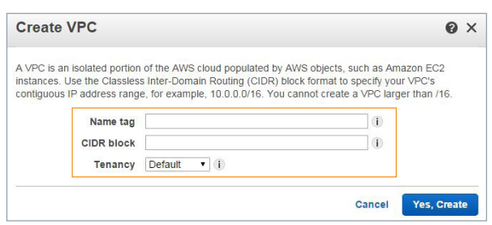
3. VPC Getting Started

VPC(Virtual Private Cloud)는 AWS에서 제공하는 가상 네트워킹 서비스입니다. 가상으로 설정한 네트워크 IP주소 정책에 따라 서브넷/라우팅/네트워크ACL 등을 자유롭게 구성할 수 있으며, VPN을 이용하여 회사 네트워크와 연결할 수도 있습니다.

3.1. VPC 생성

STEP 1

VPC를 생성할 Region 선택 후,
[Console] > [VPC] > [Your VPCs] > [Create VPC]



- Name tag : VPC의 이름을 설정합니다.
- CIDR block : 사용할 IP대역을 CIDR 표기법으로 설정합니다.
- Tenancy : VPC에서 EC2 생성 시 전용 하드웨어를 사용하려면 Dedicated로 설정합니다.

STEP 2

모든 설정을 마친 후
[Yes, Create] 버튼을 클릭하여 VPC를 생성

3.2. VPC 서브넷 생성

STEP 1

[Console] > [VPC] > [Subnets] > [Create Subnet]

- Name tag : 서브넷의 이름을 설정합니다.
- VPC : 서브넷을 생성할 VPC를 선택합니다.
- Availability Zone : 서브넷을 생성할 가용 영역을 선택합니다. No Preference를 선택시 임의의 AZ가 선택됩니다.
- CIDR block : 사용할 IP대역을 CIDR 표기법으로 설정합니다.

STEP 2

모든 설정을 마친 후 [Yes, Create] 버튼을 클릭하여 서브넷을 생성합니다.

3.3. 인터넷 게이트웨이 생성

STEP 1

[Console] > [VPC] > [Internet Gateways] > [Create Internet Gateway]

STEP 2

이름을 설정한 후 [Yes, Create] 버튼을 클릭하여 인터넷 게이트웨이를 생성합니다.

4. ELB Getting Started

ELB(Elastic Load Balancing)은 네트워크 트래픽을 EC2에 부하분산하는 로드 밸런싱 서비스입니다. 동일 Region내의 EC2를 대상으로 생성해야 하며, Multi-AZ 구성을 통해 고가용성의 서비스를 손쉽게 구축할 수 있습니다.

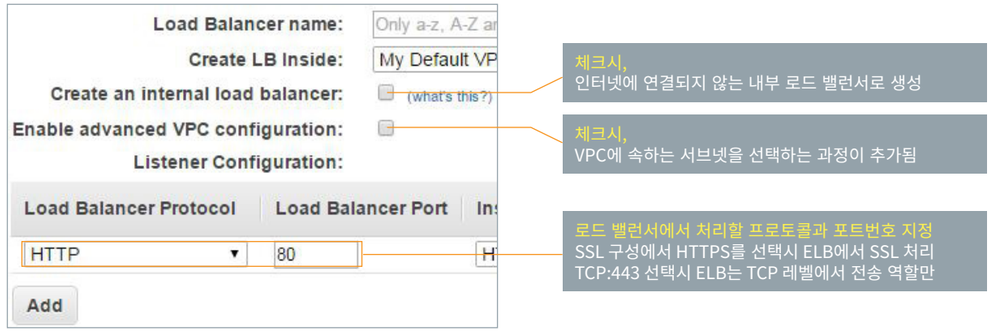
4.1. ELB 로드 밸런서 생성

STEP 0

Create Load Balancer
로드 밸런서를 생성할 Region 선택 후,
[Console] > [EC2] > [Load Balancer] >
[Create Load Balancer]

STEP 1

Define Load Balancer
로드 밸런서의 이름, 생성될 VPC, 처리할 프로토콜, 포트번호 등을 선택합니다.



STEP 2

Configure Health Check
헬스 체크에 사용할 프로토콜 및 헬스 체크 주기, 응답 대기 시간 등을 설정합니다.

STEP 3

Assign Security Group
로드 밸런서의 Security Group을 선택합니다. 기존 것을 선택하거나 신규 그룹을 생성합니다.

STEP 4

Add EC2 Instances
로드 밸런서에 연결할 EC2 인스턴스들을 선택합니다.

STEP 5

Add Tags
로드 밸런서에 원하는 태그값을 추가합니다.

STEP 6

Review
앞에서 설정한 내용을 확인 후 [Create] 버튼을 클릭하여 로드 밸런서를 생성합니다.

5. RDS Getting Started

RDS(Relational Database Service)는 AWS에서 제공하는 관계형 데이터베이스(RDBMS) 서비스입니다. 패치/백업/복구 등의 관리기능을 추가적으로 제공하며, 인스턴스성능 및 스토리지 용량도 탄력적으로 조정하여 사용할 수 있습니다.

5.1. RDS 인스턴스 생성

새로운 RDS DB 인스턴스를 생성합니다.

STEP 0

Launch a DB Instance
DB 인스턴스를 생성할 Region 선택 후,
[Console] > [RDS] > [Launch a DB Instance]

STEP 1

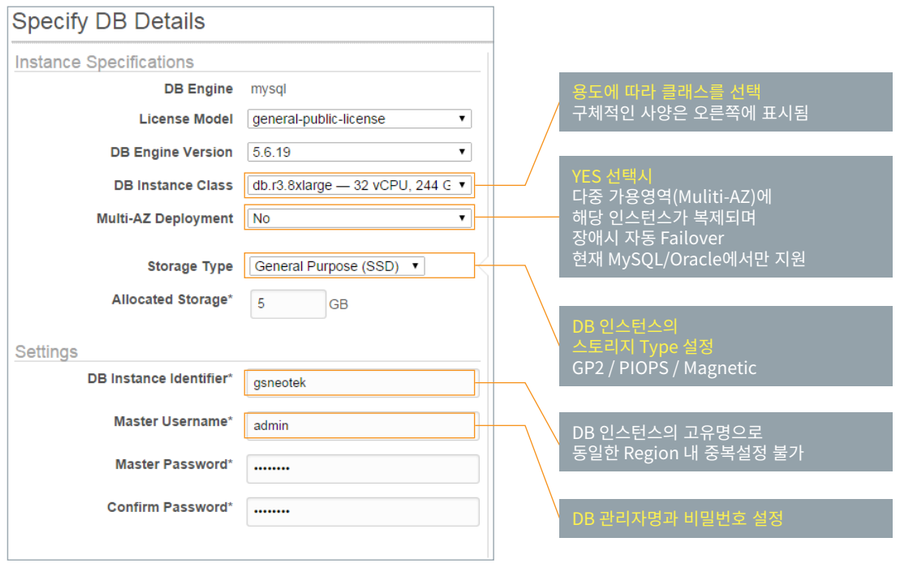
Select Engine
사용할 DB Engine(MySQL/Oracle/MSSQL/PostgreSQL)을 선택합니다.

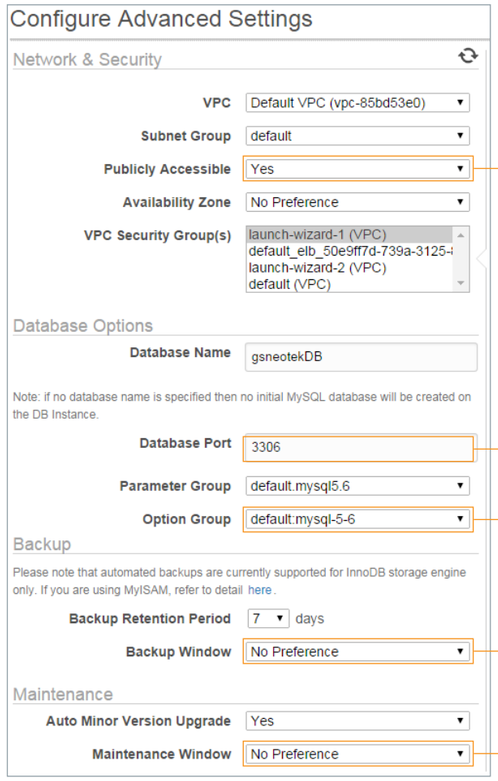
STEP 2

Production
Multi-AZ(다중 가용영역)와 Provisioned IOPS 스토리지 사용 여부를 선택합니다.
다음 단계에서 두 항목을 재설정할 수 있으므로 큰 의미는 없습니다.

STEP 3

Specify DB Details
DB 인스턴스의 세부 항목을 설정합니다.





STEP 4

Configure Advanced Settings

YES 선택시
인스턴스에 별도의 Public IP가 부여
NO 선택시
설정된 VPC 내부에서만 접근 가능

사용할 DB 포트를 설정
초기 설정값은 이후에 변경 불가

Parameter Group/Option Group 선택
이후 RDS Dashboard에서
Parameter/Option Groups 탭을 통해
관련 설정을 수정할 수 있음

자동 백업 옵션으로
백업 데이터 유지 기간을 설정
백업 수행 시간도 수동 설정 가능

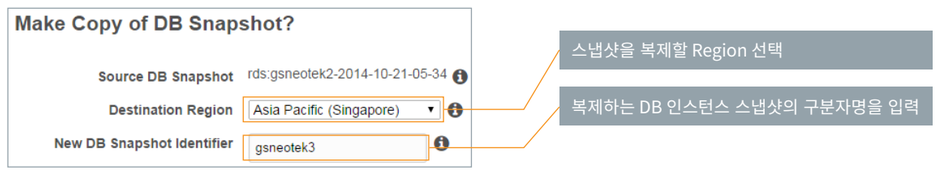
YES 선택시
버전 업데이트가 자동으로 수행
업데이트 수행 시간을 정할 수 있으며
업데이트 수행시 인스턴스는 중지됨

5.2. RDS 인스턴스를 다른 Region으로 복사

위와 같이 생성한 DB 인스턴스는 생성시 지정한 Region 내에서만 각종 설정 변경 및 운영이 가능하며, 다른 Region에 복사하여 운영하고자 할 경우엔 [DB 인스턴스 스냅샷 생성] > [Region간 스냅샷 복사] > [복사한 스냅샷으로 새로운 DB 인스턴스 생성] 순으로 복제본을 생성해야 합니다. 아래 그림은 생성한 DB 인스턴스 스냅샷을 다른 Region으로 복사하는 단계를 설명합니다.

STEP 1

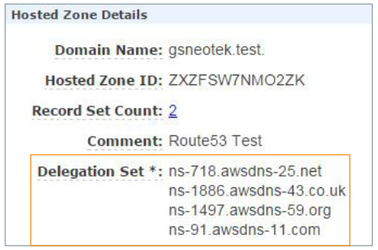
[Console] > [RDS] > [Snapshots] > [Copy Snapshot]



6. Route53 Getting Started

Route53은 AWS에서 제공하는 DNS 서비스 입니다. 일반 DNS서버와 달리 다양한 라우팅 알고리즘을 제공하며, EC2/ELB/S3/Cloudfront 등의 AWS 리소스와 손쉽게 연동할 수 있습니다.

6.1. Route53 Hosted Zone 생성



STEP 1 [Console] > [Route53] > [Hosted Zones] > [Create Hosted Zone]

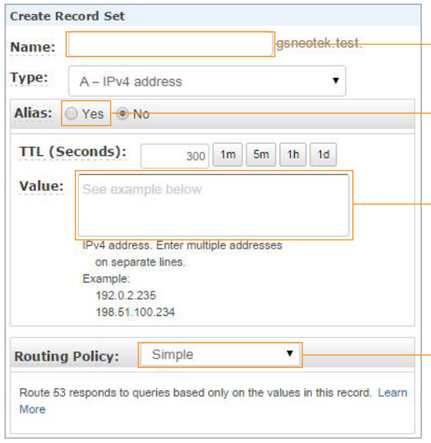
STEP 2 구입한 도메인을 입력 후 [Create] 버튼을 클릭하여 Hosted Zone을 생성합니다.

STEP 3 생성한 Hosted Zone의 정보에서 등록한 도메인의 네임서버를 확인합니다.

STEP 4 도메인을 구입한 사이트로 접속하여 위의 네임서버를 등록합니다.

6.2. Route53 A레코드 생성하기

STEP 1 [Console] > [Route53] > [Hosted Zones] > [Go to Record Sets] > [Create Record Set]



생성할 서브 도메인명 입력

Yes 선택시,
IP주소 대신 S3, ELB, CloudFront 리소스 선택 가능
A레코드에서만 활성화

도메인 네임을 쿼리했을 때
해당 레코드에 설정할 값(여기선IP) 입력

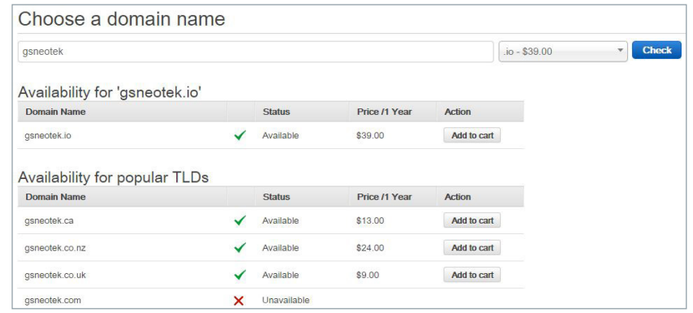
Simple 선택시,
부가기능 없이 IP주소만 전달
Weighted 선택시,
Weighted Round Robin 사용
Latency 선택시,
Latency Based Routing 사용
Failover 선택시,
DNS Failover 사용

STEP 2 [Create] 버튼을 클릭하여 A레코드를 생성합니다.

6.3. Route53에서 도메인 구입

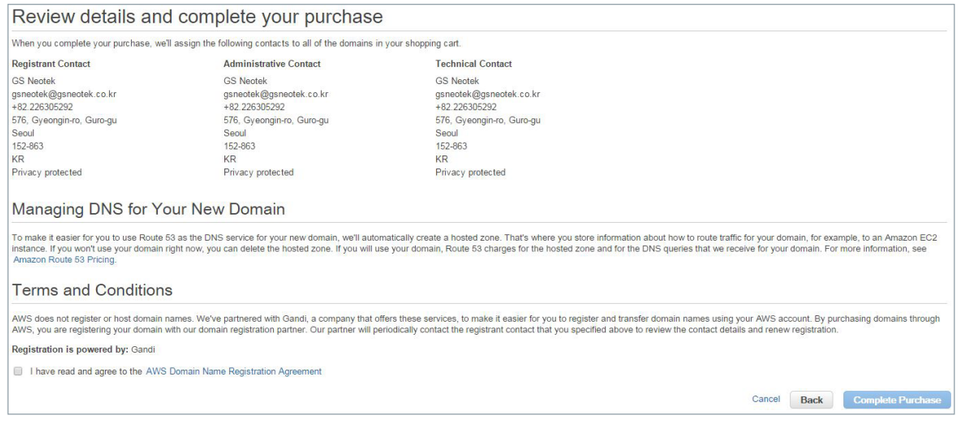
STEP 0 **Register Domain**
[Console] > [Route53] > [Domains] > [Register Domain]

STEP 1 **Domain Search**
원하는 도메인을 입력 후, [Check] 버튼을 클릭하여 사용 가능한지 확인합니다. 구입 가능한 도메인이면 [Add to cart] 버튼을 클릭하여 장바구니에 담습니다.

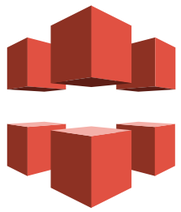


STEP 2 **Contact Details**
사용자의 개인정보를 입력한 뒤, [Continue] 버튼을 클릭합니다.

STEP 3 **Review details and complete purchase**
등록된 정보 확인 및 약관 동의 후 활성화 된 [Complete purchase] 버튼을 클릭하여 구매합니다.



STEP 4 **도메인 네임서버 등록**
구매가 완료된 도메인은 [Domains]화면에서 확인할 수 있습니다.
해당 도메인으로 Hosted Zone을 생성하면 네임서버가 자동으로 등록됩니다.



7. Cloudfront & Cloudwatch Getting Started

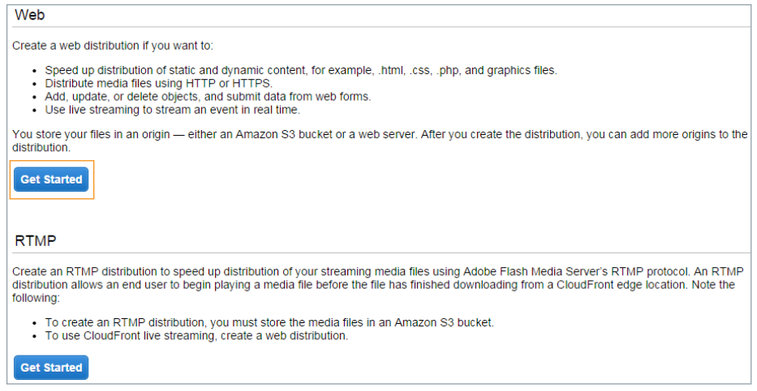
Cloudfront는 캐시를 사용하여 전세계에 파일을 빠르게 배포하는 CDN 서비스입니다. 현재 51개의 엣지 서버가 구축되어 있으며, 사용자의 위치에 따라 가장 가까운 엣지 서버에서 콘텐츠를 배포합니다. Cloudwatch는 각종 AWS 컴포넌트를 모니터링하는 서비스입니다. 컴포넌트별로 모니터링 항목이 상이하며 Cloudfront의 경우 총 6개의 Metric을 갖고 있습니다.

7.1. Cloudfront 배포 생성

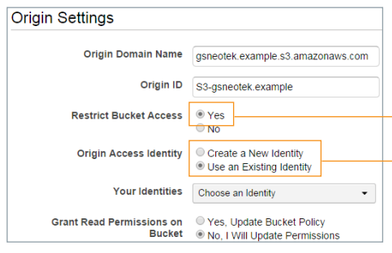
Cloudfront에서 배포를 신규 생성합니다. AWS 리소스(S3, EC2 인스턴스, ELB)뿐만 아니라 외부 웹서버도 캐시 원본(Origin)으로 설정 가능합니다. 여기서는 S3 Bucket을 Origin으로 설정합니다.

STEP 0 [Console] > [Cloudfront] > [Create Distribution]

STEP 1 **Select delivery method**
배포 방식 유형에는 WEB과 RTMP가 있습니다. 여기서는 WEB을 선택합니다.

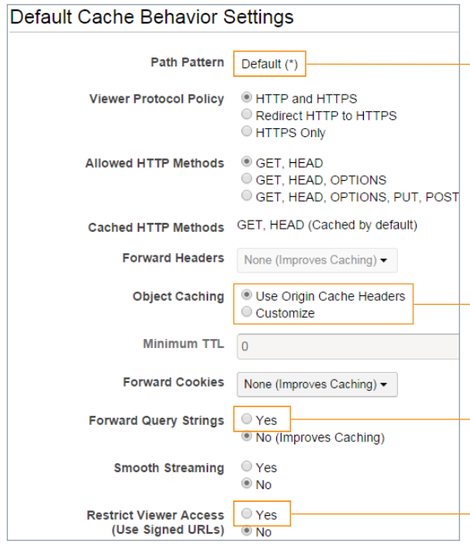


STEP 2 **Create distribution**



YES 선택시
Cloudfront를 거쳐서만 지정한 S3 Bucket(Origin)에 접근 가능하도록 설정함
해당 제어는 S3 Bucket policy를 통해 구현되므로, S3 Bucket policy 수정에 따라 다른 자원의 접근을 허용할 수 있음

지정한 S3 Bucket(Origin)에 접근시 사용될 고유 식별자(OAI) 설정



특정 경로 또는 특정 확장자의 파일만 캐시하도록 설정 가능, 기본 설정은 변경할 수 없으며 배포 생성 후에 추가 패턴 설정이 가능

파일 캐시 유지 시간을 설정

Use Origin Cache Headers 선택시
Origin HTTP 헤더에 캐시 설정값이 있으면 해당 값, 설정값이 없으면 24시간으로 동작

Customize 선택시
Minimum TTL값과 Origin HTTP 헤더의 캐시 설정값 중 긴 시간으로 동작

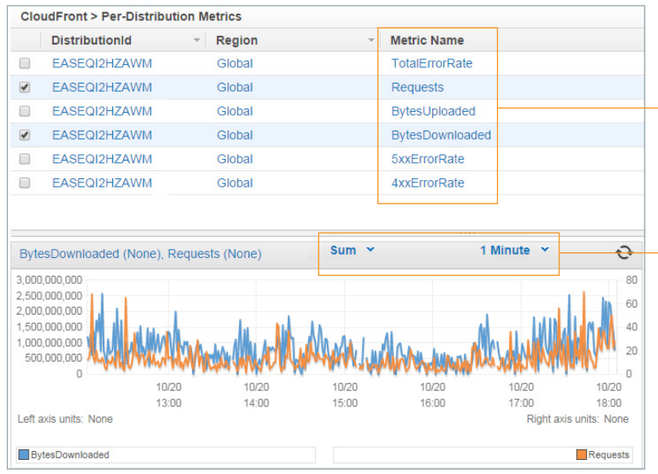
YES 선택시
URL의 Query String 값에 따라 각기 다른 버전의 object를 반환 가능

YES 선택시
Signed URL을 사용하여 Cloudfront 사용을 제한 가능

7.2. Cloudwatch로 모니터링

Cloudwatch를 사용하여 EC2, EBS, ELB, RDS 등 AWS 리소스와 관련된 다양한 값을 모니터링할 수 있습니다. 여기서 수집된 값을 기반으로 사용자는 Auto scaling 임계치를 지정하거나 Alarm을 설정할 수 있습니다. 아래 그림은 Cloudfront와 관련된 Metric입니다. DistributionID별로 Metric이 생성되며 Cloudfront 관련 Metric은 Region을 US East로 지정해야 확인할 수 있습니다.

STEP 1 [Console] > [Cloudwatch] > [Metrics] > [Cloudfront]



Cloudfront와 관련하여 총 6개의 Metric을 제공함
일례로, TotalErrorRate 는 전체 HTTP Request 중 응답코드가 4xx or 5xx인 것의 비율을 의미함

위 Metric 중

Requests
BytesUploaded
BytesDownloaded 항목은 Sum 속성이며

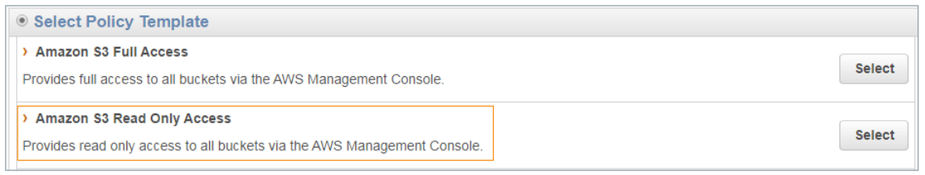
4xxErrorRate
5xxErrorRate
TotalErrorRate 항목은 Average 속성임

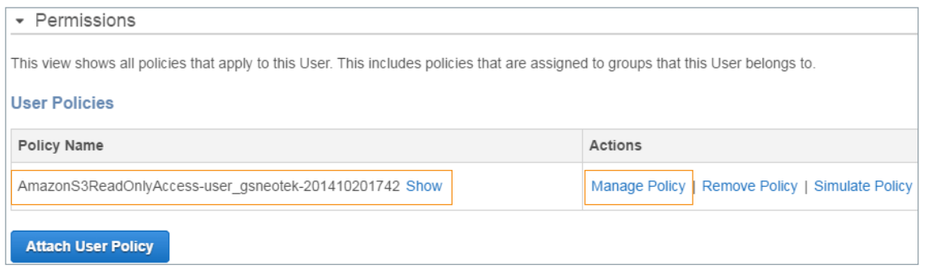
8. IAM Getting Started

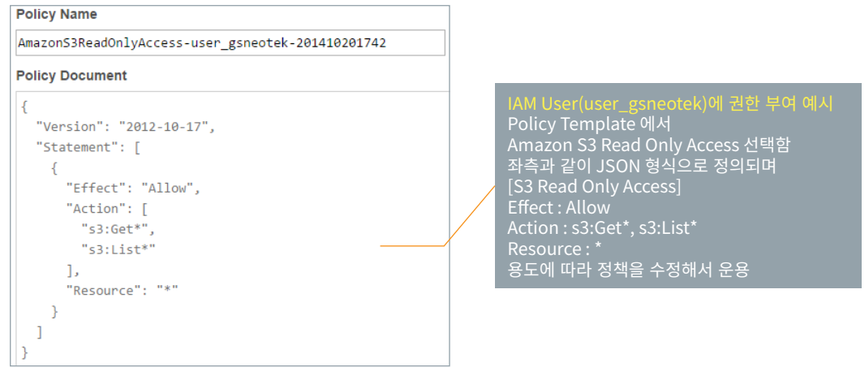
IAM은 AWS에서 제공하는 일종의 권한관리 시스템입니다. 용도에 따라 사용자/그룹/역할을 생성한 후, 각 단위별로 권한 정책을 부여할 수 있습니다. 각 AWS 리소스에 접근시, 사전에 정의된 권한 정책에 따라 접근이 허용되거나 거부됩니다.

8.1. IAM User/Group/Role

예를 들어 특정 S3 Bucket에만 접근 가능한 사용자(IAM User)를 다수 생성할 경우, 먼저 다수의 IAM User를 생성하고 각 IAM User에 모두 동일한 권한을 부여하는 방식으로 설정할 수 있습니다. 다만 이 경우, IAM Group을 생성하고 권한을 부여한 후 다수의 IAM User를 해당 IAM Group에 추가하는 형태가 더욱 이상적입니다. 또한 한 IAM User는 복수의 IAM Group에 속할 수 있으므로, 권한 정책을 되도록 세분화된 Group 단위로 운영하는 것이 좋습니다. 정책을 설정할 때는 아래와 같이 기존에 사전 정의된 Policy Template을 지정하는 방식으로 손쉽게 설정 가능하며, 부여된 정책은 JSON 형식이므로 용도에 따라 수정하여 사용할 수 있습니다. IAM Role은 보통 타 AWS 계정에서의 접근을 인증값을 노출하지 않고(임시 Credentials 사용) 구현할 때 사용됩니다.

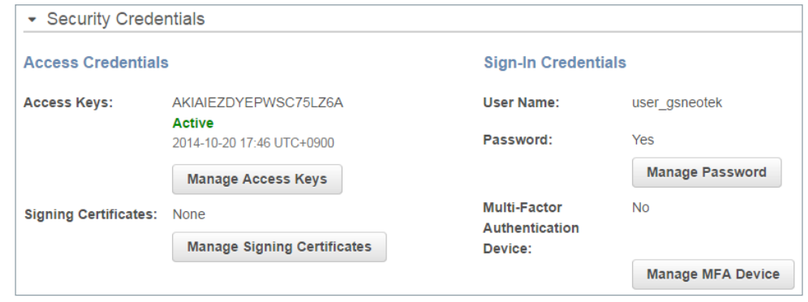






8.2. IAM Credentials

IAM User를 생성한 후에는 2가지 형태의 Credentials을 이용하여 각 AWS 리소스에 접근할 수 있습니다. Access Credentials 항목에서는 최대 2쌍의 Access Key ID와 Secret Access Key값을 생성할 수 있으며, 해당 인증값은 AWS에서 제공하는 API를 호출할 때 사용됩니다. Sign-In Credentials 항목에서는 비밀번호를 설정하여 AWS 콘솔에서 IAM User 단위로 로그인할 수 있으며, 추가적으로 MFA Device를 등록하여 2-factor 인증을 활성화할 수 있습니다.



IAM User 콘솔 로그인 페이지 주소는 IAM Dashboard에서 아래와 같이 확인할 수 있습니다.

www.wisen.co.kr

WiseN

Wisely Combine the Network platforms

AWS GETTING STARTED

GS Neotek

서울특별시 구로구 경인로 576 (구로동) [TEL] 02-2630-5795 [FAX] 02-2630-5255