AWS 서버 구축

1. AWS란

Amazon Web Services의 약자. 아마존에서 제공하는 클라우드 서비스로 EC2같은 IaaS모델부터 머신 러닝, 데이터 레이크 구축 및 분석, IoT등을 위한 서비스까지 제공하는 PaaS모델까지 다양한 클라우드 서비스를 제공한다. 최대규모의 클라우드 서비스 답게 뛰어난 보안과 안정적인 서비스를 제공하는 점과 유연하고 다양한 서비스들을 제공하는 점이 가장 큰 장점이다.

1. AWS 웹 서버 구조
2. Availability Zone (AZ)

AWS의 데이터센터가 위치하는 가용 영역이다. 각 영역은 하나 이상의 데이터 센터로 구성된다. 하지만 하나의 데이터 센터는 여러 개의 가용영역에 포함되어 있을 수 없다. AZ는 실제로 물리적으로 격리되어 있어서 물리적인 장애에 독립적입니다.

서비스 사용자는 AZ를 선택할 수 있고 여러 AZ에 걸쳐 서비스를 확장할 수 있다. 이러한 방식을 통해 장애 발생시 장애영역을 최대한 줄일 수 있고 복원에 드는 비용을 최소화할 수 있다.

1. Region

가용영역은 다시 Region이라는 큰 그룹으로 묶인다. Region은 2개 이상의 AZ로 구성되며 데이터는 이 Region안에서 유지되기 때문에 다른 Region으로 데이터를 복제하는 일은 사용자의 몫이 된다.

Region이 위치한 국가, 지역에 정보와 Network지연시간, 규정 등을 보고 사용자는 Region을 결정해야 하며, Region간의 통신은 암호화 방식을 요구한다.

1. Virtual Private Cloud (VPC)

AWS계정 전용 가상 네트워크로 다른 가상 네트워크와 논리적으로 분리되어 있다. EC2와 같은 AWS리소스를 실행할 수 있는 사용자 별로 제공되는 클라우드 네트워크 환경이라고 할 수 있다.

각 VPC는 Region별로 제공된다. VPC는 Region내의 리소스에만 접근할 수 있고 외부 Region에 있는 리소스에는 공공 네트워크를 이용하거나 VPC끼리 연결하는 방식을 이용하지 않는 이상 접근할 수 없다.

1. Subnet

VPC의 IP주소 범위. 서브넷에 존재하는 IP를 이용해서 AWS 리소스에 접근할 수 있다. 인터넷에 연결이 필요한 리소스에는 Public, 그렇지 않은 경우 Private 서브넷을 이용한다.

1. Elastic Compute Cloud (EC2)

AWS 클라우드 내에서 확장 가능한 컴퓨팅 리소스를 제공하는 서비스. 변동사항에 따라 유연하게 확장하거나 축소가 가능하기 때문에 트래픽 예측에 대한 필요성을 줄일 수 있다.

대표적으로 아래와 같은 기능들을 제공한다

* 가상 컴퓨팅 환경: 컴퓨팅 및 메모리 기능에 따라 목적에 맞는 인스턴스 생성 가능.
* Amazon Machine Image (AMI): 서버에 필요한 OS및 여러 SW가 구성된 템플릿. 인스턴스 생성이 쉬워지는 장점이 있다.
* 인스턴스를 위함 CPU, 메모리, 스토리지, 네트워크 용량 등의 구성.
* 임시 스토리지 볼륨: 임시 데이터를 저장하는 스토리지로 인스턴스 종료시에 삭제된다.
* Amazon Elastic Block Store (EBS): 영구 스토리지 볼륨처럼 사용할 수 있는 공간.
* Security Group: 인스턴스에 연결할 수 있는 프로토콜, 포트, IP 범위를 지정할 수 있는 방화벽 기능을 제공해 준다.
* Elastic IP (EIP): 동적 클라우드 컴퓨팅을 위한 고정 IP주소.
* TAG: 사용자가 생성하여 EC2리소스에 할당할 수 있는 메타데이터:

1. Relational Database service (RDS)

관계형 데이터베이스르 더욱 쉽게 설정, 운영 확장할 수 있도록 하기위해 지원하는 서비스. 탄력적인 용량 조절이 가능하다는 장점이 있다.

일반적인 물리 서버를 구입할 시에는 모든 리소스들이 묶여서 제공되지만 RDS를 이용하면 모든 컴퓨팅 리소스들이 분할 제공되므로 독립적으로 확장할 수 있다.

또한 백업, 소프트웨어 패치, 자동 장애 감지 및 복구기능 또한 포함하고 있을 뿐 안이라 원하는 시점에 수동으로 백업포인트를 만들 수도 있다.

MySQL, Oracle등 여러 DB제품들을 사용할 수 있기 때문에 번거롭게 새로 공부하거나 준비할 필요가 없다는 장점 또한 존재한다.

1. Elastic Load Balancing (ELB)

Elastic Load Balancing이란 애플리케이션 트래픽을 EC2 인스턴스, 컨테이너, IP주소등으로 자동으로 분산처리해주는 서비스다.

ELB는 여러 네트워크 계층의 부하를 분산해 준다 자세한 기능은 아래와 같다.

* Application Load Balancer: 어플리케이션 레이어(layer7)에서 작동하는 Balancer로써 VPC내의 트래픽을 라우팅 해준다. HTTP/HTTPS 어플리케이션을 Load Balancing 해줄 뿐 아니라 암호 프로토콜을 사용하여 어플리케이션 보안까지 개선한다.
* Network Load Balancer: 네트워크 레이어(layer4)에서 작동하는 Balancer다. IP프로토콜 데이터를 기반으로 VPC내의 연결을 라우팅 한다. TCP트래픽의 Load Balancing에 적합하며 AZ당 하나의 정적 IP를 사용하여 트래픽 패턴을 처리한다.
* Classic Load Balancer: EC2인스턴스에서 기본적인 로드 Balancing을 제공하며 layer4, layer7에서 동작한다. VPC에서도 사용이 가능하고 네트워크 내에 구축된 어플리케이션 용으로도 사용이 가능하다. 설정이 간단한 편인 것이 장점이지만 위의 두 Balancer보다 먼저 제공된 서비스인 만큼 제공되는 기능이 적은 것이 단점.

1. 아키텍처 구축

본격적인 AWS웹 서버 구축에 앞서 위에 기술된 아키텍처 구축과정을 거쳐야 한다. 그 순서는 아래와 같다.

1. AWS 계정 생성
2. IAM 유저 생성: IAM 그룹 생성, IAM 유저 생성, IAM 로그인
3. Cloud Trail생성: Cloud Trail생성 및 log 저장용 S3 bucket생성
4. 네트워크 구성: VPC와 subnet생성 및 Route table과 Gateway설정
5. NAT 서버 구축: Key pair, Security group, 인스턴스 생성 및 Apache web server 설치
6. ELB 생성: ELB생성, 인스턴스 연결
7. 무려 도메인으로 ROUTE53 등록 및 ELB 연결
8. DB 서버 구축