AWS_210803

***오토 스케일링**

오토 스케일링(Auto Scaling) : 미리 정해놓은 규칙에 따라 워크로드를 자동으로 확대 또는 축소할 수 있는 기술.클라우드가 제공하는 탄력성에 의 해 만들어지고, 사용자의 요구 수준을 세심하게 반영할 수 있는 혁신적인 기술. 오토 스케일링을 이용하면 처리 요구량이 급등하는 시기인, 피크 워크로드(peak workloads)에 맞춰 과잉 프로비전을 할 필요성이 사라짐. 새 리소스를 자동으로 추가 및 환경 설정하며, 처리 요구량이 줄어들면 해당 소스를 감소시킴.

***오토 스케일링의 필요성**

기존의 온프레미스 중심의 인프라를 운영하는경우 미리 피크워크로드를 예상하여 서비스 하여야함. 이러한 경우, 투자대비 효율성이 떨어지거나 비용의 낭비를 할 가능성이 있는데 이때 AWS의 오토 스케일링과 일래스틱 로드 밸런싱을 이용하면 다수의 EC2 서버의 처리 능력을 적절히 분배해, 워크로드 상승과 하락에 맞춰 서버의 처리 성능을 높이고, 워크로드 하락에 맞춰 효율적으로 서버처리가 가능함.

***오토 스케일링의 장점**

-동적 스케일링(Dynamic Scaling) : 오토 스케일링의 가장 큰 장점으로, 사용자의 요구 수준에 따라 리소스를 동적으로 스케일가능.

-헬스 체크(Health check)와 서버 플릿 관리(server fleet management) : 오토 스케일링을 이용해서 헬스 체크를 하는 과정에서 특정 인스턴스의 문제가 감지되면 자동으로 다른 인스턴스로 교체함.

-일레스틱 로드 밸런싱(ELB, Elastic load Balancing)으로 오토 스케일링을 설정하는 경우, ELB의 헬스 체크 기능도 사용가능. ELB의 헬스 체크 기능은 다양하며, 하드웨어 실패, 시스템 성능 약화 등 다양한 상태 확인 기능 사용가능

-로드 밸런싱(Load balancing) : 오토 스케일링은 리소스를 동적으로 스케일업 또는 스케일다운하므로, ELB와 함께 사용하면 다수의 EC2 인스턴스에게 워크로드를 효과적으로 분배할 수 있고 다수의 AZ에 분포된 EC2 인스턴스에 대 한 워크로드도 자동으로 분배하도록 설정가능. 오토 스 케일링은 CPU 활용률에 맞춰 EC2 인스턴스의 수를 자동으로 조정함.

***실행 환경 설정**

실행 환경설정(Launch Configuration) : 오토 스케일링으로 인스턴스를 확장 또는 축소하려할때 어떤 서버를 사용할 지 결정해야할때 이용. 일종의 템플릿으

로 AMI 상세정보, 인스턴스 타입, 키 페어, 시 큐리티 그룹, IAM 인스턴스 프로 파일, 유저 데이터, 부착 스토리지 등 인스턴스에 대한 모든 정보를 담고있음.

***오토 스케일링 그룹(Auto Scaling Group)**

스케일업 및 스케일다운 규칙의 모음, EC2 인스턴스의 시작부터 삭제까지의 모든 동작에 한 규칙과 정책을 담 고 있음. 오토 스케일링 그룹은 EC2 서버가 하나의 그룹으로 작동하는 방법과 사용자 정의에 따라 동적으로 그룹화하는 방법을 정의함.

***스케일링 유형 인스턴스 레벨 유지**

기본 스케일링 계획으로도 부르며, 항상 실행 상태를 유지하고자 하는 인스턴스의 수를 정의가능. 항상 실행하려는 최소한의 서버 수 또는 특정 서버 수를 지정할 수 있으며, 오토 스케일링 그룹은 해당 숫자만큼의 인스턴스를 항상 실행.

***수동 스케일링**

콘솔이나 API, CLI 등을 이용해서 수동으로 스케일링 작업을 수행. 수동 스케일링을 선택하면, 사용자가 직접 인스턴스를 추가 또는 삭제해야 하며, 스케일 자동화라는 오토 스케일링의 기본 취지를 생각하면 수동 스케일링 방식은 추천x

***요구별 스케일링**

시스템의 요구 수준에 맞추는 방식으로 AWS 클라우드와치(CloudWatch)가 모니터링하는 CPU, 디 스크 쓰기와 읽기, 네트워크 유입과 유출 등 주요 지표를 바탕으로 요구 수준에 맞춰 스케일링 규칙 을 정하는 방식.

***일정별 스케일링**

트래픽의 변화가 예측 가능하고, 특정 시간대에 어느 정도의 트래픽이 증가하는지에 대한 패턴을 파악하고 있을시, 일정별 스케일링을 사용에 적합.

***정확한 용량(Exact capacity)**

– 스케일링 정책 작성 시 증가 또는 감소시킬 정확한 용량을 정의할 수 있다. – 예를 들어, 그룹의 현재 용량이 2개의 인스턴스이고 조정 용량이 4개의 인스턴스인 경우 오토 스케일링 정책이 실 행되면 4개의 인스턴스로 변경된다. • 숫자 지정 용량 변경(Chance in capacity) – 숫자에 따라 현재의 용량을 증가 또는 감소시킬 수 있다. – 예를 들어, 그룹의 현재 용량이 2개의 인스턴스이고 조정 용량이 4개인 경우 오토 스케일링 정책이 실행되면 6개 의 인스턴스로 변경된다. 3. 오토 스케일링 그룹

***단계 조정**

심플 스케일링 또는 단계별 심플 스케일링을 사용하면 다음과 같이 처리 용량을 변경할 수 있다. • 퍼센트 단위의 용량 변경(Percentage change in capacity) – 퍼센트 단위로 현재의 용량을 증가 또는 감소 시킬 수 있다. – 예를 들어, 그룹의 현재 용량이 10개의 인스턴스이고 조정 용량이 20%인 경우, 정책이 실행되면 10개의 20%인 2 개가 추가, 총 12개의 인스턴스가 된다. – 이 때 주의 사항은 입력은 퍼센트 단위이지만, 결과값은 정수형 숫자로 나오게 되며, 소수점 자릿수가 있는 경우 반올림 또는 반내림하게 된다. – 예를 들어, 1보다 작은 소수점 값은 반내림해 13.5의 경우 13이 된다. – 0에서 1 사이의 수는 1이 되며, 0.77은 1이 된다. – 0에서 - 1 사이의 수는 -1이 되고, -0.72는 -1이 된다. – 마지막으로 -1보다 작은 수는 반올림해서, -8.87은 -8이 된다. 3. 오토 스케일링 그룹

*** 대상 추적 조정**

타깃 트래킹 스케일링 정책(Target Tracking Scaling Polcies)
동적환경설정 가능, 이때 미리 정의된 성능지표를 이용하거 커스텀 성능지표를 만들어서 타깃 값으로 설정할 수 있음.

***인스턴스 삭제 정책**

오토 스케일링은 스케일업 정책은 물론, 스케일다운 정책도 반영. 스케일다운 정책이 반영되면, EC2 인스턴스가 삭제되며, 서버를 셧다운하는 것은 좀 더 확실한 리소스 관리를 위해서도 필요함. 스케일다운 정책에서 정확하게 몇 개의 인스턴스를 삭제할 인지 정의가능. 인스턴스 삭제 정책을 작성하는 방법은 매우 다양한데, 그중 하나는 가장 오랫동안 실행된 서버를 삭제하는 것. 그리고 시간 단위 과금이 임박한 서버를 삭제하는 것이 있음. 또한 실행 환경설정 기간이 가장 긴 인스턴스를 삭제하는 것도 가능.

*** 일래스틱 로드 밸런싱(ELB, Elastic Load Balancing)**

로드 밸런서(load balancer)는 다수의 서버에 유입되는 트래픽을 분산시켜 워크로드의 균형을 잡기 위한 하드웨어로 현대적인 다수의 애플리케이션은 웹 서버에 로드 밸런서를 배치해 트래픽을 분산시키고 워크로드의 균형을 맞추며 탄력성을 증대시킴.

*** AWS ELB의 주요 장점**

-탄력성(Elastic) : ELB의 최대 장점은 자동적 확장성이다. 서버 다운시에도 탄력적으로 대응가능.
-통합성(Integrated) : ELB는 다양한 AWS 서비스와 통합해 사용가능.
-안전성(Secured) : ELB는 통합 인증 관리, SSL 복호화, 포트 포워딩 등 다수의 보안 성능을 제공함.

-가용성(Highly available) : ELB는 최고 수준의 고가용성 아키텍처를 구현하는 데 도움을 줌. 다수의 AZ에 배포된 EC2 인스턴스에 애플리케이션을 배포해 트래픽을 여러 AZ로 분산시킬 수 있기 때문에 하나의 AZ가 모두 다운돼도 사용자의 애플리케이션은 문제 없이 실행 상태를 유지가능.
-저렴함(Cheap) : ELB는 저렴하며 비용효율적. 온프레미스 환경에서 로드 밸런서를 이용하는것에 비해 ELB는 로드 밸런싱 작업을 자동화해서 많은 시간과 비용을 절약가능.

***로드 밸런서의 유형**

네트워크 로드 밸런서 • 네트워크 로드 밸런서(NLB, Network Load Balancer)는 TCP 로드 벨런서로도 부르며, OSI 모델의 레이어에서 작동. NLB는 기본적으로 연결 기반 로드 밸런서이며 EC2 인스턴스, 컨테이너, IP 주소 등의 연결을 관리. 이들이 생성하는 모든 요청은 로드 밸런서를 통해 흘러가며, 로드 밸런서는 전달되는 패킷을 관리하고, 이를 백엔드로 전달하는 역할을 수행. TCP와 SSL 모두 지원.

***애플리케이션 로드 밸런서**

애플리케이션 로드밸런서(ALB, Application Load Balancer) 는 OSI 모델의 레이어에 해당. HTTP 와 HTTPS를 지원. 애플리케이션으로부터 패키지가 되면, 헤더로 받은 뒤 어디로 전송할지 결정하는데 이 연결은 로드 밸런서에서 종료되고 전달 내용은 연결 풀 형태로 모여있다가 로드 밸런서가 요청 을 받으면, 연결(connection pool)을 이용해 필요한 곳으로 전달한다. ALB의 호스트 기반 라우팅을 통해 HTTP 헤더의 Host 필드에 따라 클라이언트 요청을 라우팅 할 수 있고, 경로 기반 라우팅을 통해 HTTP 헤더의 URL 경로에 따라 클라이언트 요청을 라우팅가능.

***클래식 로드밸런서**

클래식 EC2 인스턴스를 지원하며, 네트워크 로드 밸런 싱 및 애플리케이션 로드 밸런싱 모두를 지원. 클래식 EC2 인스턴스를 사용하는 경우가 아니라면 필요에 따라 네트워크 로드 밸런싱 또는 애플리 케이션 로드 밸런싱을 사용해야함.

*** 로드 밸런서의 유형**

로드 밸런서는 외부형(external facing) 또는 내부형(internal facing)으로 설정할 수 있으며, 인터넷을 통한 접근이 가능한 로드 밸런서를 외부 로드 밸런서라 부름.

***로드 밸런서의 핵심 개념 및 용어**

로드 밸런서는 고확장성, 고가용성의 완전 관리형 서비스로, 애플리케이션 로드 밸런서는 콘텐츠 기반 라우팅을 지원하므로, 하나의 로드 밸런서에 다수의 애 플리케이션을 호스팅가능.

***로드 밸런서의 핵심 구성 요소**

-리스너 : 리스너(Listener)는 트래픽이 유입되는 로드 밸런서 리스너의 연결 부분에 대한 프로토콜과 포트를 의미.

-타깃 그룹과 타깃 : 타깃 그룹을 논리적으로 그룹화한 것.

-규칙 : 로드 밸런서에서 규칙(Rules)은 리스너와 타깃 그룹을 연결, 조건(conditions)과 동작(act)으로 구성.

-헬스 체크 : 로드 밸런서는 기본적으로 트래픽의 효율적인 분배를 돕는 장치이지만, 애플리케이션의 고가용성을 구현하는 도구, 고가용성을 구현하기 위해서는 헬스 체크(Health Check)가 필요하며, 이는 타깃과 타깃 그룹이 항상 문제 없이 가동될 수 있도록 미리 정의된 주기별로 타깃과 타깃 그룹의 상태를 확인하는 동작.

-다중 AZ의 활용 : 오토 스케일링과 ELB를 활용해 애플리케이션을 구현할 때는 가능한 한 다중 AZ 기반으로 할 것을 권장하는데, 이는 다중 AZ가 고 가용성을 구현하기 위한 기본 구조.

***실습하기 1: 직접 오토 스케일링 준비하기**