|  |
| --- |
|  |
| 쉬운 것들을 합성해서 뒤의 어렵고 복잡한 것들이 파생된다.  원천적인 것들.  처음에 배우는 것이 제일 중요한 것이다.  재미있는 부분까지만 해라.  지루해지면 배운 것들을 이용해서 만들어봐라.  힘들고 지치면 그만하고 하고싶은거 해라. |
| 남의 컴퓨터를 빌려서 원격 제어를 통해 사용하는 것  클라우드 컴퓨팅.  호스팅.  호스팅 비즈니스.  컴퓨팅 자원을 빌려주는 임대 사업.  Elastic Compute Service  컴퓨터를 빌려주는 서비스.  컴퓨터에 소프트웨어를 설치해주고 그것을 대신 운영해주는 방향으로 클라우드 컴퓨팅이 확장되고 있다.  이것이 유용한 이유는 서버 컴퓨터 쪽에서 동작하는 소프트웨어들을 설치하고 운영하고 백업하는 것이 상당히 까다롭고 위험한 일이기 때문. |
| 꿈과 기능이 필요하다.  무엇인가 하고자 했을 때 필요한 것.  꿈꾸는게 없는데 기능을 배우고 있을 때 초라해진다.  원격제어를 통해 필요한 때만 쓸 수 있다.  클라우드 컴퓨팅. |
| aws는 무료가 아니다.  요금을 살펴봐야 한다.  온디맨드는 쓰는만큼 돈을 내는 것이다.  예약 인스턴스는,  일종의 할인 쿠폰을 구입해서 일년 동안 쓸 수 있는 양을 한꺼번에 사는것 비슷.  요금을 선택하면  EC2-이용하고자 하는 컴퓨팅 자원, 즉 운영체제를 선택해야 한다.  AWS는 전세계에 컴퓨터를 분산해서 보관하고 있다.  리전.  요금 정책을 잘 알아야  다른 서비스들과 비교할 수 있고  좀 더 효율적인 서비스 이용을 할 수 있게 된다. |
| AWS같은 클라우드 서비스들은  테스트 기간을 제공한다. 프리티어  무료로 사용할 수 있는 기간과 용량을 잘 살펴보자  AWS 클라우드 콘솔.  이곳을 통해 아마존을 제어할 수 있다 |
| 클라우드 콘솔을 이용해서 EC2에서 컴퓨터를 만드는 방법.  우리는 ec2 수업이 아니라  aws의 기본적인 사용 방법을 보는 수업입니다.  ec2에서는 컴퓨터 한 대를 ec2라고 부른다.  키페어 생성.  굉장히 복잡한 비밀번호 파일이 다운로드 된다.  유출에 주의한다.  다운로드는 한 번만 가능한 것 같다.  잃어버리면 컴퓨터를 복제하고 새로운 키 페어를 발급받아야 한다.  인스턴스 상태pending > running  이제 원격 제어를 통한 이 과정으로 만든 컴퓨터를 사용할 수 있다. |
| 중지는 요금이 안나가고 데이터는 남아 있는 상태  종료는 요금도 안나가고 데이터도 사라진다.  컴퓨터를 다시 쓸거라면 중지를 해야한다.  데이터 유지를 위해 약간의 비용이 발생한다.  조심해야 할 것은,  컴퓨터를 끄면 ip가 해제되고  다시 키면 다른 ip가 할당된다.  그래서 접속하는 방법을 다시 재설정 해주어야 한다.  예전에 세팅해 놓은 방법 그대로 접속할 수 없다.  윈도우 컴퓨터 기준-실행 파일을 다시 다운받아서 실행시켜야 함.  데이터가 삭제되면-종료하면 복구할 수 없다.  클라우드 컴퓨팅의 본질.  임대한다.  필요할 때.  큰 자유를 준다. |
| 돈 관리를 잘 해야 한다.  인생이 달라질 수 있단다. |
| <보안>  데이터를 누군가가 가져간다거나  우리 인프라를 장악하고 사용해서 요금 폭탄을 맞게 한다거나    MFA – 멀티 펙터 인증  원 타임 패스워드 – 시간이 지나면 달라지는 비밀변호  otp에 설정된? 비밀번호를 모르면 접속할 수 없다.  멀티 펙터 – 아이디/비밀번호 인증과 원 타임 패스워드를 합친것  스마트폰이랑 연결하네? google authentication  컴:qr코드-폰:번호  연속적으로 입력하면 된다. 번호.  문제없으면 mfa 등록된다.  mfa 설정을 하면 우리가 훨씬 더 안전하게 클라우드 콘솔에 접속할 수 있게 되는 것이다.  계정을 도둑맞을 가능성을 아주 떨어뜨릴 수 있다.  강력한 보안 수단 |
| <계정종료>  계정 닫기'누르면 계정이 폐쇄됩니다.  90일 이내에는 다시 열 수 있다.  지난 후에는 등록한 이메일로도 새 계정을 만들 수 없음  요금폭탄을 줄 수 있는 IT 인프라이기 때문에  사용하지 않으면 계정 폐쇄가 한 방법이 될 수 있다. |
| 이제 AWS의 기본 사용버을 알게 되었다.  이제 어떤 문제가 있을 때 그 문제를 aws에서 쉽게 해결할 수 있는 방법이 있는지 찾아볼 수 있게 되었다.  새롭게 출시되는 it 기술들을 살펴보면 it 트렌드가 보입니다.  소식 정도 구독해놓는 것이 유용할 수 있다.  어떤 서비스들이 있는지 알고있으면 좋다.  너무 깊게 모든것을 정복할 필요는 없다.  EC2는 컴퓨터를 통째로 빌려주는 서비스. 자유도가 높다.  다를 건 AWS가 자동으로 해준다. S3 파일서버  S3에 퍼블릭으로 업로드 하면  객체 url를 전달하기만 해도 다른 사용자들이 볼 수 있게 된다.  컴퓨터를 통째로 빌려주는 것이 아니라  컴퓨터를 활용해서 할 수 있는 여러 가지 일들을 서비스 형태로 제공해 준다.  이런 서비스들을 이용하면 생산성을 비약적으로 향상시킬 수 있게 된다.  aws를 다루는 여러 가지 방법을 살펴보자.  지금까지 우리는 클라우드 콘솔을 통해서 또 웹을 통해서 또 gui 방식으로 aws를 사용했다.  그런데 이런 클라우드 콘솔이 아니라  커맨드 라인, 프로그래밍을 통해서도 aws를 제어할 수 있다.  검색하고 뒤져보고 사용설명서를 참조하며 배우면 된다.  가설을 세우고 그것을 검증하는 과정을 통해 스스로 방법을 찾는다. |
| S3.  저장소. 저장장치.  파일을 잃어버리고 싶지 않을때.  고장안나고 잃어버리지 않는 저장장치를 바란다.  S3의 이점 중에 내구성이 언급되고,  하나의 aws 리전 내에서 수 킬로 떨어진 최소 3개의 무리적 가용 영역에 자동 분산된다고 언급.  이게 내구성. 법적으로 계약되어 있는 상태.  파일 보관 뿐만 아니라  인터넷에 연결되어 있어서  전 세계 누구나 접속하여 다운로드 받을 수 있게 할 수도 있다.  파일 서버.  아무리 많은 사용자가 들어와도 감당 가능. 죽지않는.  파일에 대한 또 다른 꿈.  5년 뒤에는 삭제되었으면 좋겠다.  1년 뒤에 저렴 저장장치로 이동했으면 좋겠다. 등등 가능.  파일 하나를 계속 해서 수정할 때, 예전 파일 백업 가능. 쉽게 찾읈도 있음.  이게 버전관리.  훨씬 더 많은 기능 제공하고 있다.  복잡한 기능 다 알 필요 없다.  정보 시스템 CRUD 할 줄 알면 자신감 가져라.  S3 구성요소 살펴보자.  3인방으로 이루어져 있다.  버킷. 하나의 프로젝트.  버킷 안에 폴더.  폴더 안에 파일이 보관된다.  파일은 오브젝트라고 부른다. 오브젝트=파일.  파일과 파일의 부가정보들을 묶어서 오브젝트라고 한다.  파일 폴더 버킷을 생성하고 읽고 수정하고 삭제하자. 목표.  파일을 안전하게 보관하고 보호하기 위해 S3를 이용하면 간편하게 해결 가능.  어떻게 이용하는지 알아보자.  고객이 있는 지역을 리전으로.  버전 관리를 체크하면 수정되는 모든 파일 저장.  로깅.  암호화 설정도 할 수 있다.  퍼블릿. 파일 업로드 하면 누구나 볼 수 있게 된다는 상태.  공개 파일이라도 비공개 상태로 되게 할 수도 있나보다.  s3는 내부적인 이유로 인해 폴더 이름 수정이 쉽지 않다.  그냥 새 폴더 만들어라.  폴더 삭제.  s3의 핵이라고 할 수 있는  실제 데이터가 담기는 그릇이라고 할 수 있는 객체를  생성해보자.  객체=파일.  파일을 업로드 해보자.  폴더도 통째로 올릴 수 있나?ㅇㅇ  s3는 가격,속도,안정성에 따라서 비용이 달라진다.  상황에 따라 파일의 중요도/쓰임새에 따라 선택지를 주고 있다.  객체를 누구나 살펴볼 수 있게 하는 방법을 살펴보자.  그러기 위해선 권한을 알아야 한다.  다른 사람에게 공개하고 싶다.  url 복사해서 주면 된다.  퍼블릭으로 설정하면 된다.  권한]으로 하는 방법,  오른쪽 마우스 퍼블릭 설정,으로 하는 방법이 있다.  권한 주는 방법과 링크 공유하는 방법을 살펴보았다. |
| 파일의 속성  스토리지 클래스  스탠다드 등등 4가지 형태가 있다. 사실은 5가지.  각각 장단점.  파일의 성격에 따라 다르게 설정하면 된다.  처음부터 정교한 스토리지 클래스들이 필요하지 않다.  그냥 스탠다드 써라.  규모가 커지면 약간의 차이만으로도 몇 억씩 달라진다.  그때 클래스 타입이란게 중요해 지는 거다.  적정 수준에서 자기의 학습 노력을 쏟을 때를 구분해라. 헐. |
| 요금 정책 살펴보자.  스토리지 요금. 저장공간 요금.  사용 요금. 얼마나 사용했느냐. 얼마나 업로드/다운로드.  데이터 전송 요금. 얼마나 전송했냐에 따라. |
| 파일 저장소인 S3는 웹 서버 기능이 기본적으로 내장되어 있다.  정적 웹 사이트 호스팅 기능을 켜면 s3에 보관한 파일로 전세계 누구나 방문가능한 웹 사이트 만들 수 있다.  버킷을 리전별로 지정하게 되어있다.  고객이 있는 곳으로 리전으로 설정해라.  s3는 cloud front 기능을 가지고 있다.  이런 서비스를 content delivery network라고 한다.  서비스 전달하는 네트우크  CDN . cdn이 동작하는 원리는,  전세계 cdn 서버 분산하고 요청이 들어오면 가장 가까운 곳에 미리 저장하고 있었떤 콘텐츠를 보내주는 것.  훨씬 더 빠르게 서비스 할 수 있다.  cloud front  버전관리 기능.  변경된 내용을 모두 저장하여 복원가능하다.  만드는 시스템이 시스템의 이력을 남기는 것을 중요시 한다면  s3버전 관리 기능써라.  라이프 사이클 기능 키면,  시간에 따라 저장소 유형이 달라진다?  스탠다드 -> ia |
| 나의 문제를 코딩으로 해결하는 엔지니어  <https://www.advancedwebranking.com/html/>  unsplash 사이트 이미지 공공 |
| s3의 웹 호스팅 기능. 웹 서버.    캐싱에 대해 잘 모른다면 http 캐싱 수업을 먼저 봅니다.  클라우드 크론트를 즐기기 위해서는 두 가지 기능을 이해해야 한다.  캐시와 CDN.  캐시가 기본.  php 등은 웹 페이지를 동적으로 프로그래밍 적으로 생성해서 응답해주는 방식.  정적 웹 페이지보다 많은 자원을 사용한다. 더 많은 시간+비용.  사용자에게 한 번 서비스하고 그 다음에 요청했을 때는 더 이상 이 복잡한 과정을 거치지 않고 저장된 결과를 서비스 하겠다는 꿈.  저장된결과를 이용해서 응답하는 것. 캐시.  그러한 것들을 전담 해주는 서버를 캐싱 서버라고 한다.  클라우드 프론트의 첫 번째 역할은 캐싱 서버를 제공하는 것?  두 번째는 세계화.  전세계 어디에 있건 1초에 응답. CDN  클라우드 프론트는 기본적으로 캐시 서버.  전 세계의 AWS 인프라를 이용하는 캐시 서버.  그래서 CDN으로 동작하기도 한다.  웹 서버를 하나 장만?  origin domain Name에는 오로지 도메인 이름만 적어야 한다. |
| 캐시를 제어하는 방법에 대해 알아보자.  origin이 바뀌었음에도 distribution이 바뀌지 못하는 신선도 문제.  관리자 화면에서 path pattern은 클라우드 프론트로 들어오는 모든 경로의 동작 방법을 지정한다는 뜻.  default(\*)  클라우드 프론트가 얼마나 캐시를 지속할지 설정할 수 있다.  웹 브라우저가 해당 정보를 받으면 웹 브라우저도 클라우드 프론트에 요청하지 않는다.  그 만큼 웹 서버는 여유.  접속자 정보를 로깅하는 아파치.  reload를 했는데도 전혀 반응이 없다? 웹서버는 놀고있고 캐시가 작동중. 클라우드 프론트.  php로 하네.  header에 max age  max age가 0이면 캐시를 안쓰겠다는 뜻.  자기자신으로 가는 링크를 누르면 캐시 쓴다?  from cache.  기본적으로 클라우드 프론트는 캐시를 24시간동안 저장한다.  캐시를 강제로 지우는 방법이 필요해졌다.  캐시를 무효화-강제 삭제하는 데에는 돈이 든다.  다른 전략이 필요.  오리진 캐시타임과 프론트 캐시타임은 무관하지 않다.  프론트 위주로 설정된다는 거지?  7일차 캐시설정4은 다시보면 좋을것 같다  DNS 공부를 해봐라.  그리고 aws에서 제공하는 dns 전용 서비스 라우트 53을 공부해라  도메인 장착.  사용자가 정보를 작성하는 기술을 사용하려면 보안이 중요하다.  HTTPS, SSL, TLS를 살펴봐라.  동적 웹 페이지 공부.  클라우드 프론트 통계 페이지 잘 살펴보기. |
| 클라우드 프론트 생성 6일차 3 영상 보기. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |