Module 2 : 가장 간단한 아키텍처

---------------------------------------

1. AWS Well-Architected Tool : 최신 AWS 모범 사례에 대한 온디맨드 액세스를 제공하는 셀프 서비스 도구

2. S3 는 객체기반 스토리지, 99.999999999% 내구성, 이벤트 트리거

3. S3 는 정적 웹 콘텐츠 , 데이터 분석용 대규모 저장소 , 백업 저장소 등 사용

4. S3 액세스 제어 : 기본적으로 소유자만 접근, Public , Bucket 정책 , CORS 정책

5. S3 액세스 포인트 : 추가 Endpoint와 전용 액세스 정책을 별도로 구성

6. S3 버전 관리 : 데이터가 삭제 되거나 변경이 되어도 이전 데이터에 버전을 부여하여 관리

7. S3 멀티파트 업로드 : 단일 데이터를 여러개의 별도 파티션으로 병렬 전송

8. S3 Transfer Acceleration : CloudFront 엣지 로케이션을 이용하여 데이터 전송성능 향상, 별도의 엔드포인트 사용, 비용 추가

9. AWS Snowball : 페타바이트 규모 데이터 전송

10. AWS Snowmobile : 엑사 바이트 규모 전송

11. S3 비용 : 데이터 전송비용, PUT, COPY, POST, LIST 및 GET 요청비용, 저장 비용등

12. S3 스토리지 클래스 : Standard, Sandard IA, One Zone IA , Glacier/Deep Archive

13. S3 수명 주기 정책 사용

14. S3 Glacier : 장기데이터 저장소, S3 Standard 보다 매우 저렴. / 데이터를 가져올때는 시간이 소요 (신속검색, 표준검색, 대량검색)

15. 리전 선택 이유 : 데이터 주권및 규정준수, 사용자와 데이터 간 근접성, 서비스 및 기능 가용성, 비용 효율성

Module3: 컴퓨팅 계층 추가

--------------------------------

1. Amazon 머신 이미지(AMI)는 클라우드의 가상 서버인 인스턴스를 시작하는 데 필요한 정보 제공(OS,APP등)

2. AMI 는 사전 구축, 직접생성, AWS Marketplace에서 가져올 수 있음.

3. EC2 Image Builder 서비스를 사용하여 AMI 구성을 자동화

4. EC2 사용자 데이터(User Data) : EC2 인스턴스를 생성할때 사용자 데이터를 인스턴스에 전달할 수 있는 옵션

5. 인스턴스 메타데이터 : 인스턴스 내에서 인스턴스의 메터데이터 ( http://169.254.169.254/latest/meta-data/public-hostname)

6. EBS : EC2 인스턴스를 위해 안정적이고 분리 가능한 블록 수준 스토리지, 인스턴스 수명주기동안 영구적, 인스턴스 종료후에도 데이터는 유지

7. 인스턴스 스토어: 휘발성 , 인스턴스 중단시 데이터 손실, 중요 데이터는 항상 백업

8. EBS 볼륨 유형 : 범용 SSD , 프로비저닝된 IOPS SSD , 처리량 최적화 HDD , 콜드 HDD

9. EBS 최적화 인스턴스 구성으로 EBS I/O를 위한 추가 전용 용량 제공

10. EC2의 공유 파일 시스템 으로 S3 , EFS/FSX 사용가능

11. EFS 는 Linux 워크로드를 지원, NFSv4 지원

12. FSX 는 Windows 워크로드 (Windows 공유) 및 Lustre 워크로드

13. Instance Type : 패밀리이름 세대번호. 인스턴스 크기. (t2.large) , 다양한 워크로드를 지원 , 범용, 컴퓨팅 최적화, 메모리 최적화, 가속화 컴퓨팅, 스토리지 최적화

14. Instance Type 의 세대가 높을 수록 성능은 높고 비용은 낮은 경향이 있음.

15. 온디맨드 인스턴스 : 컴퓨팅 파워에 대한 초당(Amazon Linux 및 Ubuntu) 또는 시간당(다른 모든 OS) 비용을 지불

16. 예약 인스턴스(RI) : 리전내에 특정 Instance Type에 대해 1년 3년간에 장기 사용을 약속하고 할인

17. Savings Plans : 리전내에 특정 인스턴스 패밀리에 장기 사용 약속하고 할인

18. 스팟 인스턴스 : 미사용 Amazon EC2 용량 구매 , 인스턴스 종료 가능성을 염두해 두고 사용하는 요금제, 종료 2분전 사전 경고 활용 (https://docs.aws.amazon.com/ko\_kr/AWSEC2/latest/UserGuide/spot-interruptions.html#spot-instance-termination-notices)

19. 전용 인스턴스 : 비 특정 하드웨어에서 실행되는 단일 테넌스 요구 사항에 적합

20. 전용 호스트 : 특정 하드웨어에서 실행되는 단일 테넌스 요구및 host id기반 라이센스 프로그램 사용에 적합

21. AWS Compute Optimizer : 인스턴스 사용량에 대한 지능적인 분석

22. 클러스터 배치 그룹 : 컴퓨팅 계층이 가능한 한 가장 짧은 지연 시간 및 가장 높은 초당 패킷 네트워크 성능요구시

23. 분산형 배치 그룹 : 서로 분리되어야 하는 소수의 크리티컬 인스턴스에 도움

24. 파티션 배치 그룹 : 대규모의 분산 및 복제 워크로드에 적용 (HDFS, HBase, Cassandra 와 같은환경에서 )

Module 4 : 데이터 베이스 계층 추가

----------------------------------------------------

1. Database 사용시 고려 사항 : 확장성, 총스토리지 요구사항 , 객체크기및 유형, 내구성 고려

2. AWS에서의 Database 사용은 관리형과 비관리형 서비스를 이용할 수 있다.

3. RDS는 관리형 관계형 Database들 (Oracle, MS Sql, MySql, PostrgeSQL, MariaDB) 를 생성하고 운영 관리 한다. 64T 까지

4. DynamoDB 는 관리형 비관계형 database 이고, Key Value 방식과 Document 방식을 지원한다.

5. DynamoDB의 글로벌 테이블 : 여러 리전에 공통된 테이블을 만들고 데이터를 복제한다.

6. DynamoDB는 RCU (초당 4k 읽는 수)/ WCU (초당 1k 쓸 수 있는 수)

7. DynamoDB는 읽기 일관성 옵션을 제공 : 최종 일관성 (RCU 0.5) 과 강력한 일관성(RCU 1)

8. Amazon Neptune : 관리형 그래프 데이터베이스

Module 5 : AWS 기반 네트워킹 1부

-------------------------------------------

1. VPC : AWS Cloud의 프라이빗 네트워크 공간으로 리전별로 구성

2. VPC는 리전별 최대 5개까지 Soft 제한이 있음.

3. VPC 는 CIDR 형식으로 IP를 부여함. 최대 /16까지 설정

4. 서브넷은 리소스 그룹을 격리할 수 있는 VPC IP 주소 범위의 세그먼트 또는 파티션으로

5. 라우팅 테이블 : VPC 내에 리소스간에 트레픽의 경로설정

6. IGW : Internet Gateway 로 VPC가 인터넷에 연결하기 위한 Gateway로써 사용함.

7. Private Subnet : 인터넷에 직접적으로 연결하지 않는 네트워크로 인터넷 연결이 필요할 경우 NAT 이용

8. Public Subnet : 인터넷에 직접연결하는 네트워크로 반드시 IGW와 Route 설정이 필요함.

9. NAT Gateway : Private subnet 내에 인스턴스가 인터넷 외부로 나가기 위한 목적으로 생성

10. 탄력적 네트워크 인터페이스(ENI) : 인스턴스에 논리적인 Network card 개념으로 IP address설정, 인스턴스 간에 이동이 가능함.

11. 탄­­­­­­력적 IP 주소(EIP)는 동적 클라우드 컴퓨팅을 위해 설계된 고정 퍼블릭 IPv4 주소

12. 보안 그룹 : 인스턴스의 가상 방화벽 , 인바운드, 아웃바운드에 대한 IP ,Port 설정,

13. 네트워크 ACL : Subnet 경계에 대한 방화벽 설정. 기본은 모두 open

Module 6 : AWS 기반 네트워킹 2부

----------------------------------------------

1. VGW : 가상 프라이빗 게이트웨이, Amazon VPC와 다른 네트워크 사이에 프라이빗 연결(VPN)

2. 온프라이머스 와 AWS VPC 연결은 , VPN , DX (Direct connect ) 이용

3. Direct connect : 인터넷을 통하지 않고 온프라이머스와 AWS간에 프라이빗 네트워크연결

4. DX 복원력 향상을 위해 VPN과 함께 사용

5. VPC Peering : VPC 와 VPC간에 1:1 연결 , 리전간 연결, AWS Account 간 연결

6. VPC 엔드포인트 : VPC 내에서 AWS Service들(S3,DynamoDB등)을 사용시 인터넷을 사용하지 않고 내부 Private 하게 연결

7. Elastic Load Balancing (ELB) : 수신되는 트레픽을 여러 EC2 , Container 및 IP 주소로 분산

8. ELB 는 ALB (Application Load Balancer) , NLB (Network Load Balancer), CLB (Classic Load Balancer) , GLB

9. 고 가용성 구성의 예로 DB 이중화, ELB를 이용한 가용 영역간 로드벨런스

10. 다중 리전간 고가용성을 위해 Route53 를 이용한 리전간 Failover 구성 , Route53은 DNS 기능.

Module 7 : IAM

------------------

1. AWS 계정 루트사용자는 가능한 사용을 하지 말고 IAM 사용자를 만들어 최소 권한의 원칙을 적용하여 운영

2. AWS Identity and Access Management(IAM) 로 사용자, 그룹, 역할, 연동 사용자로 관리

3. IAM 사용자 : 자체적인 인증 수단 ( User/Password 또는 Key ) 를 보유하고 정책을 통해 권한부여됨.

4. 권한 부여 방식 : ‘리소스 기반’ – 연결된 AWS 리소스 과 ‘자격 증명 기반’ – 연결된 IAM 보안 주체

5. IAM 사용자 그룹 : 사용자들을 그룹으로 묶어서 권한 부여

6. IAM 역할(Role) : 사용자 또는 서비스가 필요한 리소스에 액세스하기 위한 권한 집합. 타 AWS계정 자원도 액세스 가능(교차 계정)

7. 외부에서 STS 자격 증명 브로커 또는 SAML 를 통한 인증

8. Amazon Cognito : 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공 , User pool 과 Identity pool (facebook, google, Twiter등과 연계)

Module 8 : 다수의 IAM 사용자 및 계정 관리

------------------------------------------------------

1. AWS Organizations : AWS 계정들에 대한 중앙 집중식 관리 , OU와 계정별 SCP 정책 설정

2. 복수 계정의 결제 : AWS Organizations 을 이용하여 비용을 중앙화

3. Amazon CloudWatch 의 결제 알림 기능을 통해 AWS요금을 모니터링

4. Tag : AWS 리소스의 추적에 도움이는 AWS 자원들에 대한 메터 데이터

5. AWS resource group : 동일한 TAG 기반 또는 AWS CloudFormation stack 기반으로 그룹화한 리소스 집합

6. AWS config 서비스를 이용하여 규정 준수 TAG 설정

Module 9 : 탄력성, 고가용성 및 모니터링

----------------------------------------------------

1. 고가용성 요소 : 내결함성 , 복구성 , 확장성 및 탄력성

2. 탄력성 : 용량 요구사항이 변화함에 따라 지능적으로 확장 및 축소

3. 세 가지 유형의 탄력성 : 시간 기반, 볼륨 기반, 예측 기반

4. Amazon CloudWatch : AWS 자원들에 대한 지표, 로그, 이벤트, 경보,규칙 등을 생성 및 모니터링

5. AWS CloudTrail : 계정에서 이루어지는 모든 API 호출을 기록하고, 지정된 Amazon S3 버킷에 로그를 저장

6. VPC Flow Logs : VPC의 트래픽 흐름 세부 정보를 캡처하여 CloudWatch Logs 나 S3에 저장

7. Amazon EC2 Auto Scaling : 지정된 조건에 따라 인스턴스를 시작 또는 종료, ELB에 자동 등록

8. RDS 에 대한 조정 : Read Only 복제본으로 수평조정, Instance type 및 Storage 사이즈 변경으로 수직 조정

9. DynamoDB 조정 : RCU/WCU , AutoScaling , 온디맨드

Module 10 : 자동화

-------------

1. AWS CloudFormation : AWS 리소스들을 안전하고 반복 가능하도록 구성하는 방식을 Json 또는 Yaml 형식의 템플릿 파일 사용

2. AWS CloudFormation의 드리프트 감지 기능으로 탬플릿과 다르게 구성이 변경되는 것을 감지함

3. AWS Quick Start : 보안 및 고가용성 관련 AWS 모범 사례를 기반으로 구축할 때 활용할 수 있도록 제공, <https://aws.amazon.com/quickstart/>

4. AWS OpsWorks : Chef를 사용하여 모든 형태와 규모의 애플리케이션을 구성하고 운영하도록 지원하는 구성 관리 서비스

5. AWS Elastic Beanstalk : 개발자가 기본 인프라에 대해 걱정할 필요 없이 클라우드에 확장 가능한 웹 애플리케이션 및 서비스를 배포하고 유지 관리할 수 있도록 도움

Module 11 : 배포 방법

-------------

1. 배포전략 : Blue-Green , Canary

2. 리전 및 계정간 배포 : AWS CloudFormation StackSets 이용

3. 배포간 트레픽 라우팅. : Route 53 라우팅 가중치 기반

4. AWS Systems Manager : 소프트웨어 인벤토리 수집, OS 패치 적용, 시스템 이미지 생성, Windows 및 Linux 운영 체제 구성을 자동

5. CI/CD : 지속적이 통합 / 지속적인 전달과 지속적인 배포

Module 12 : 캐싱

---------------------

1. Amazon CloudFront : 웹 사이트, API, 동영상 콘텐츠 또는 기타 웹 자산의 전송을 가속화하는 글로벌 CDN 서비스

2. 콘텐츠 만료 방법 : Time To Live(TTL) , 객체 이름 변경, 객체 무효화(Invalidation)

3. WEB 세션 관리 는 ELB를 이용한 고정 세션 과 분산 캐쉬를 이용한 방법

4. Amazon DynamoDB Accelerator(DAX) : 3개노드부터 10개 노드까지 클러스터 단위로 구성되며 DynamoDB의 응답 시간을 줄이기 위해 사용

5. Amazon ElastiCache 는 In-memory를 이용한 메모리 캐쉬로서 Memcached , Redis 의 Opensource를 관리형으로 지원

6. Cache 전략에는 연속쓰기와 레이지 로딩 전략이 있다.

7. Cache 에 TTL을 설정하여 일정 기간 존재해야 하는 데이터를 제거할 때 효과적으로 사용

Module 13: 데이터 보안

------------------------------

1. AWS KMS는 Key 관리형 서비스

① 봉투 암호화 (Envelope encryption) 을 사용하면 암호화된 Data와 암호화된 Data Key를 사용하게 되며 Data Key를 관리하지 않아도 됨.

② KMS Master key는 AWS 관리형 키를 자동으로 사용할 수 있으며, 고객 관리형 기본키 및 고객 마스터키(CMK)를 등록하여 사용, KMS의 마스터 키는 절대 외부로 유출 안됨.

③ CMK는 매년 자동 Key 교체 설정 가능, KEY 스케줄 삭제 기능 제공

④ KMS의 Key 저장소로 CloudHSM 에 가능

2. CloudHSM은 암호화 작업을 처리하고 암호화 키에 대한 보안 스토리지를 제공하는 컴퓨팅 디바이스

3. 서버측 암호화 : 서버의 컴퓨팅 파워를 활용하여 암호화.

4. 클라이언트 측 암호화: 클라이언트에서 데이터을 암호화 하여 전송, 전송 데이터 보호

5. ELB에서 TLS Termination (TLS 종료) 지원

6. AWS Certification Manager (ACM) : SSL /TLS 인증서 관리

7. Amazon Macie: S3 를 검색하여 개인정보와 같이 잘못 공개된 내용 검색

8. Amazon GuardDuty : 계정 내에/계정간 악의적인 활동및 승인되지 않은 동작을 지속적으로 모니터링

Module 14 : 결합 해제된 아키텍처 구축

----------------------------------------------

1. Amazon Simple Queue Service (Amazon SQS) : 완전 관리형 메시지 대기열 서비스, 메세지가 처리 및 삭제 될 때까지 저장됨.

2. 대기열은 표준 대기열과 FIFO 대기열을 제공함.

3. 배달 못한 편지 대기열 (DLQ) 를 지원하여 메세지가 처리 되지 않는 상황 파악

4. 가시성 제한 시간 : SQS에 저장된 메세지를 수신하는 경우 다른 수신자가 메제지를 보지 못하도록 설정하는 시간

5. 긴 폴링 기능을 통해서 메시지가 큐에 들어올 때 까지 긴 시간 동안 기다릴 수 있다.

6. Amazon Simple Notification Service (Amazon SNS) : 손쉽게 알림 기능을 설정, 작동 및 전송할 수 있는 웹 서비스, “게시-구독”(pub-sub) 메시징 패러다임을 따르며 “푸시” 메커니즘을 사용하여 클라이언트에 알림을 전달

7. SNS 구독 유형 : Email, HTTP/HTTPS , SMS , SQS , Lambda 함수

8. Amazon MQ : Apache MQ용 관리형 메시지 브로커 서비스로 JMS, NMS, AMQP, STOMP, MQTT, WebSocket 프로토콜과 호환됨.

Module 15 : 최적화 검토

----------------------------------------------

1. 더 적합한 인스턴스 유형을 사용하고 있나요?

2. 관리형 서비스를 사용을 먼저 고려해 보세요.

3. 아키텍처에 복원력이 있나요?

4. 아키텍처 구성요소간에 긴밀한 종속성을 제거 합니다. 마이크로 아키텍처, 결합 해제

5. 다양한 요금 모델을 사용합니다.

6. 모범사례

① 확장성을 활성화

② 환경을 자동화

③ 삭제 가능한 리소스 사용

④ 구성 요소의 소결합

⑤ 서버가 아닌 서비스 설계

⑥ 알맞은 데이터베이스 솔루션 선택

⑦ 단일 장애점 (SPOF) 제거

⑧ 비용 최적화

⑨ 캐싱 사용

⑩ 전체 인프라 보안

Module 16: 마이크로 서비스

----------------------------------

1. 마이크로 서비스 아키텍쳐 : 잘 정의된 API를 통해 통신하는 독립 서비스로 구성된 애플리케이션 / 자제 적인 전용 데이터 스토어를 갖는 독립적 형태

① 퍼블릭 API에만 의존

② 해당 작업에 적합한 도구를 사용

③ 서비스를 보호한다.

④ 에코시스템 내에서 훌륭한 시민이 된다.

⑤ 기술 혁신 이상이어야 한다.

⑥ 모든것을 자동화 한다.

2. Amazon Elastic Container Service (Amazon ECS) : 도커 컨테이너를 지원하는 확장성과 성능이 뛰어난 컨테이너 관리 서비스

3. AWS Fargate : 완전 관리형 컨테이너 서비스

Module 17: 서버리스 아키텍처

------------------------------------

1. AWS Lambda : 완전 관리형 컴퓨팅 서비스, Node.js, Java, Python, C#, Go, Ruby 등의 언어를 지원함. 이벤트에 대한 응답으로 코드 실행 방식

2. AWS Lambda 는 최대 15분까지 실행함.

3. Amazon API Gateway : 애플리케이션의 "현관" 역할을 하는 API를 생성 , 최대 수십만 건의 동시 API 호출을 처리, REST API, HTTP API , WEB Socket 지원

4. AWS Step Functions : 시각적 워크플로를 사용한 마이크로 서비스 조정,

Module 18: 복원력을 위한 구축

-------------------------------------

1. Amazon Aurora Global Database : 리전간 복제, 1초 이내의 복제 지연, 리전 장애시 보조 리전은 1분 이내에 읽기/쓰기 가능

2. AWS CloudMap : 모든 엔드포인트 추적

3. WAF 는 OSI 모델의 애플리케이션 계층(계층 7)을 보호,

4. WAF 는 CloudFront , ALB , API Gateway , AWS AppSync 를 보호함.

Module 19: AWS 기반 네트워킹 3부

-----------------------------------------

1. 향상된 네트워킹 ( enhanced networking ) : SR-IOV를 사용하여 고성능 네트워크 기능, ENA (탄력적 네트워크 어댑터) 를 사용

2. 점보 프레임 지원 : MTU 사이즈를 최대 9001 Byte , VPC내에 Instance간 , DX를 통해 온프라이머간에 지원,

3. AWS Global Accelerator : Global 사용자에게 제공하는 애플리케이션의 가용성과 성능을 개선하는 네트워크

4. AWS Site-to-Site VPN – 관리형 VPN으로 온프라이머스와 VPC 간에 연결 구성, 정적 구성과 동적 구성 지원, 2개의 Tunnel 지원

5. AWS Transit Gateway (TGW) : VPC , VPN ,DX 등을 HUB처럼 연결 또는 Routing을 이용한 전송 제어

Module 20: 비용 절감

-------------------------

1. AWS 월 사용량 계산기 :<https://calculator.aws>

2. Trusted Advisor : AWS 모범 사례를 따르는 데 도움이 되는 권장 사항을 제공

3. 최적의 인스턴스 크기 조정

4. 수요를 고려한 설계 : Auto Scaling을 사용한 사용자 수요 기반 비용 설계, Amazon SQS 를 사용한 대기열 기반 비용 설계

5. 비용의 목표를 맞춰 적절한 리소스, 적절한 요금를 선택

6. 사용량을 통제하기 위한 액세스 제어및 절차 사용

7. 사용량과 지출에 대한 모니터링

8. 필요없는 리소스를 폐기하거나 일시적으로 필요없는 리소스를 중단

9. 데이터 전송 비용을 고려

Module 21 : 마이그레이션

----------------------------

1. 마이그레이션 전략 : 리호스트, 리플랫폼, 재구매, 리팩터링, 사용중지, 유지

2. Database Migration Service (AWS DMS) 은 동일 기존 또는 다른 기종간에 데이터베이스를 마이그레이션 할 때 사용된다.

3. DMS 로 마이그레이션시 물리적인 매체를 이용하고자 하는 경우 AWS Snowball Edge를 이용함.

4. 이 기종간에 데이터베이스 마이그레이션 일 경우 Schema Conversion Tool(SCT) 을 사용함

5. AWS Server Migration Service (AWS SMS)는온프레미스VMware 가상시스템을 Amazon EC2 클라우드로 마이그레이션 하는 작업을자동화

Module 22 : RTO/RPO 및 백업 복구 설정

-----------------------------------------------

1. 복구 시점 목표(RPO) , 복구 시간 목표(RTO)

2. AWS Backup : 중앙 집중식 백업 관리

3. AWS Storage Gateway : 온프라이머스와 AWS의 스토리지 서비스(S3, Glacier , EBS) 간에 연결,

4. 백업 복구 전략 : 백업 및 복원 , 파일럿 라이트 , 완전 동작 저용량 스탠바이, 다중 사이트 액티브-액티브