AhnLab

White Paper

AhnLab Cloud 안전한 클라우드 환경 조성 방안

클라우드 전환 가속화: 보안의 중요성 ↑

운영과 유지보수에 초점을 둔 기존 MSP는 보안에서 한계 드러내

개요

바야흐로 클라우드 전성시대다. 시장조사기관 가트너(Gartner)에 따르면, 2021년 글로벌 퍼블릭 클라우드 시장 규모는 약 3,961억 달러(약 463조 원)이며, 2022년에는 약 4,821억 달러(약 562조 원) 수준까지 성장할 전망이다.

또한, 가트너는 2021년 3조 2400억 원 규모로 형성된 국내 퍼블릭 클라우드 시장 규모가 2022년에는 3조 7238억 원까지 늘어날 것으로 예측했다. 코로나19 팬데믹 이후 디지털 트랜스포메이션이 가속화되면서 클라우드 도입이 급증했고 클라우드 도입 확대는 포스트 코로나 시대에도 계속될 것으로 보인다.

다만, '경제적 이득'을 취할 수 있는 곳으로 몰리는 해커들의 특성에 따라 클라우드 환경을 노리는 공격 역시 증가 및 다변화되고 있다. 특히 기업 담당자들은 자원 공유, 서비스화 등 기존 레거시 환경과는 다른, 클라우드가 가진 특성 때문에 보안을 고민하지 않을 수 없게 되었다.

기존 MSP의 한계

그 동안 운영과 유지보수에 초점을 두고 클라우드 관리 서비스를 이용해 온 기업들은 보안 측면에서 어려움을 마주하는 경우가 많았다. 예를 들면, 클라우드 구축이 완료된 이후 보안 요소를 추가하고자 할 때 솔루션 적용이 불가능하거나 보안 요소를 추가할 때마다 보안 구성을 변경해야 하는 애로사항이 있었다. 또, 복수의 보안 업체와 협업하면서 요구사항이 충돌하는 경우도 발생했다.

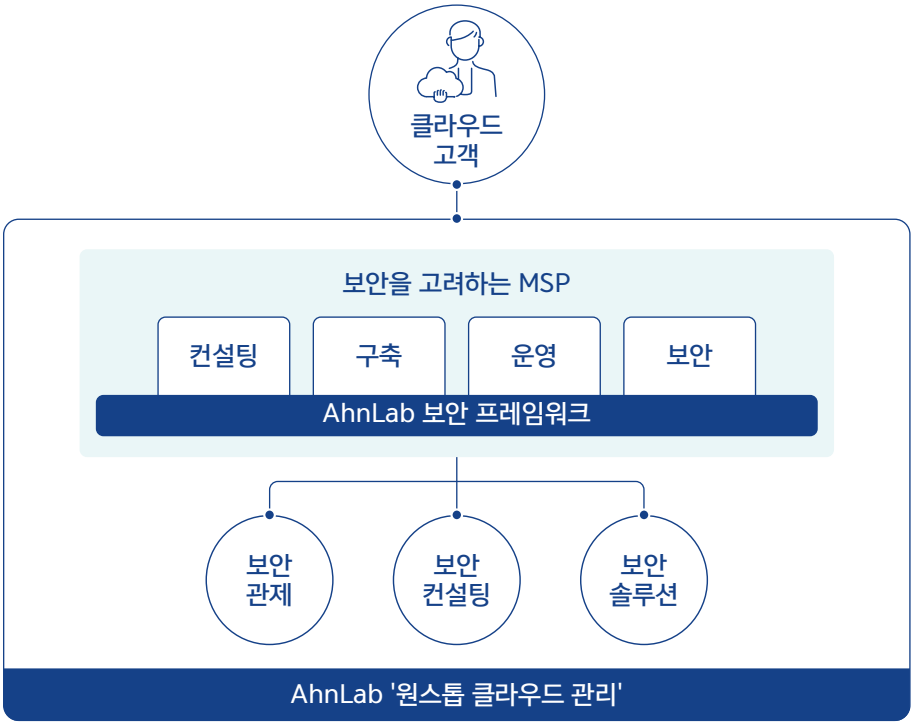
AhnLab Cloud의 차별성

기존 MSP에 탁월한 보안 역량을 더한 '시큐어 MSP'

클라우드 전환이 가속화되면서, 전통적인 MSP의 역할을 넘어 이전(Migration)과 운영뿐만 아니라 온프레미스 수준 이상의 보안이 중요한 이슈로 떠오르고 있다. 특히, 금융이나 공공 기관에서도 클라우드 사용이 가능해지면서 기존의 MSP 역할에 보안 역량을 갖춘 '시큐어 MSP(Secure MSP)'에 대한 요구가 증가하고 있다.

안랩의 시큐어 MSP 'AhnLab Cloud'

AhnLab Cloud는 클라우드 설계부터 구축과 운영까지 모든 단계에서 보안에 최적화된 클라우드 관리 역량을 '원스톱'으로 제공해 이러한 어려움들을 해결한다. 그 배경에는 안랩이 지난 25년간 축적해온 보안 기술력과 노하우가 집약된 보안 프레임워크가 자리하고 있다.



[그림 1] AhnLab Cloud 개념도

AhnLab Cloud는 컨설팅 단계서부터 자사의 보안 프레임워크에 기반해 클라우드 구축 방안을 도출하고, 보안 컴플라이언스를 빠짐없이 반영해 고객의 비즈니스 모델에 최적화된 서비스 아키텍처를 수립한다. 이를 바탕으로 클라우드 운영과 보안을 관리하는 '매니지드 서비스', 24시간 고객의 보안 현황을 모니터링하는 보안관제 및 기술지원까지 고객이 클라우드 환경에서 효율성과 안정성을 갖추기 위한 필수 요소들을 종합적으로 제공한다.

AhnLab Cloud의 가치

기존 MSP의 구축 & 운영 역량에 안랩의 보안 솔루션, 서비스 및 노하우를 더해 안정적인 클라우드 관리 가능

컨설팅		· 안랩의 보안 표준 아키텍처에 기반한 클라우드 구축 방안 도출 · 고객군 별 클라우드 보안 컴플라이언스 요소의 누락 없는 반영 · 고객사의 비즈니스 모델을 고려해 서비스에 최적화된아키텍처 수립 · 클라우드 보안 컴플라이언스, 인증 & 인허가, 기술 진단 컨설팅
구축	**분석**	· 환경 조사: 사용자 환경 분석 & IT 현황 분석 & 보안 현황 분석 · 요구사항 도출 · 주요 이슈 확인: 인프라 이슈 & 보안 이슈 · 전환 우선순위 선정
	설계	· 아키텍처 정의: 구성요소 범위 선정 및 아키텍처 프레임워크 정의 · 아키텍처 설계: 인프라 및 보안 아키텍처 설계 · 아키텍처 검증
	운영환경 구성	· 구축 환경 사전 검증: 정의 → 구성 → 검증 · 운영 환경 구성: 정의 → 구성 → 검증
	전환 및 테스트	· 전환 계획 수립 · 클라우드 전환: 인프라 전환 → 데이터 이관 → 애플리케이션 이관 → 보안 구성 전환
운영		· 24x7 운영지원 · 전문가 서비스
보안		· 보안 관제 · 보안 컨설팅 · 보안 솔루션

[표 1] AhnLab Cloud 세부 구성

AhnLab Cloud를 이용하면 안랩의 '클라우드 네이티브' 보안 솔루션과 다양한 클라우드 특화 서비스를 바탕으로 보안을 최적화하고 안전하게 클라우드를 설계할 수 있다. 뿐만 아니라 체계적이고 효율적인 클라우드 운영을 통해 전체적인 사용성을 향상시킬 수 있다.

또한, 최근 클라우드 보안 침해의 주요 원인으로 지목되고 있는 '설정 오류' 등 운영 실수로 인한 보안 사고도 예방 가능하다. 기존에 클라우드를 도입해 운영하고 있는 고객도 AhnLab Cloud를 이용해 클라우드 보안을 강화하거나 재구축 할 수 있다.

클라우드 보안의 핵심: 올바른 설정

기업의 클라우드 보안 사고를 살펴보면 권한 관리 실패로 인해 발생하는 경우가 대부분이다. 지난 2019년 7월, 1억 600만 명이 넘는 고객 정보가 유출된 은행 '캐피탈 원(Capital One)' 해킹 사건 역시 클라우드 사용자가 클라우드 설정을 잘못해 발생한 사고였다. 이는 클라우드 환경에서의 권한을 올바르게 관리하면 보안 침해의 상당 부분을 예방할 수 있다는 의미이기도 하다.

안전한 클라우드 환경 조성의 첫걸음

각 사용자의 업무 특성에 맞는 '권한' 부여 필요

클라우드 서비스 제공자(Cloud Service Provider: CSP)가 제시하는 책임 공유 모델(Shared Responsibility Model)에서도 확인할 수 있듯, 클라우드 내부 사용에 대한 관리는 사용자의 몫이자 책임이다. 따라서 외부자에 대한 경계 뿐만 아니라 내부자의 악의적인 행위나 실수에 의한 보안 사고를 스스로 방지해야 한다. 이를 위해서는 사용자별 역할과 책임에 맞는 최소 권한만 부여하는 것이 클라우드 보안의 첫 걸음이라 할 수 있다.

간단히 말하면, 각 사용자 업무 특성에 맞는 권한을 부여해야 한다는 것이다. 예를 들어 클라우드 환경에 개발 시스템과 운영 시스템이 구성되어 있다고 하면, 개발자는 개발 시스템, 운영자는 운영 시스템에만 접근이 가능하도록 권한을 부여해 각자의 업무와 역할에 맞는 접근 제어를 구현해야 한다.

AhnLab Cloud 역시 수년에 걸친 보안 솔루션 및 서비스 경험을 통해 클라우드 보안 요소 중 '식별'과 '접근 제어'를 필수적인 보안 요소로 고려하고 있다. 또, 사용자와 클라우드 리소스를 정확하게 식별하고 각 서비스에 대한 접근 권한 관리에 초점을 두어 고객사 환경에 적합한 클라우드 환경을 제안 및 구성한다.

전 세계적으로 가장 많이 사용되는 클라우드 서비스 중 하나인 아마존웹서비스(Amazon Web Services: AWS)에 대한 AhnLab Cloud의 접근 통제 및 설정 관리 방안을 통해 보다 자세한 내용을 살펴보도록 하자.

AhnLab Cloud로 안정적인 AWS 아키텍처 구축하기

AWS에 처음 회원 가입하면 생성되는 것은 계정, 즉 'Account'다. 이 때 발급되는 'User'는 'Root User'로 해당 Account에 대한 모든 권한을 갖는다. 강력한 권한을 가지는 만큼 안전하게 관리해야 하므로 꼭 필요한 경우를 제외하고는 Root User를 사용하지 않을 것을 권고한다. 일반 사용자의 경우 'AWS IAM(Identity and Access Management)' 서비스를 이용해 역할과 권한을 부여할 수 있다.

AWS IAM이란 AWS 리소스에 대한 접근을 안전하게 제어할 수 있는 서비스다. 이 서비스를 통해 클라우드 리소스에 접근하는 주체를 인증하고 권한을 부여해 접근 제어를 할 수 있다. IAM이라는 이름에서도 알 수 있듯, 'Identity'는 AWS로 요청을 할 수 있는 보안 주체를 의미하며 'Access Management'는 해당 보안 주체들이 리소스에 대해 어떤 일을 할 수 있는지에 대한 권한을 의미한다.

AWS IAM 서비스의 세부 항목을 살펴보면 크게 ▲User ▲Group ▲Policy ▲Role로 나눌 수 있다. User는 AWS를 사용하는 개별 이용자를 말하고 Group은 User를 모아 놓은 것으로 일반적으로 역할 별로 Group을 만들어 사용한다. 또한 각 User는 여러 개의 Group에 속할 수도 있다.

AWS 아키텍처 구축 Tip 1

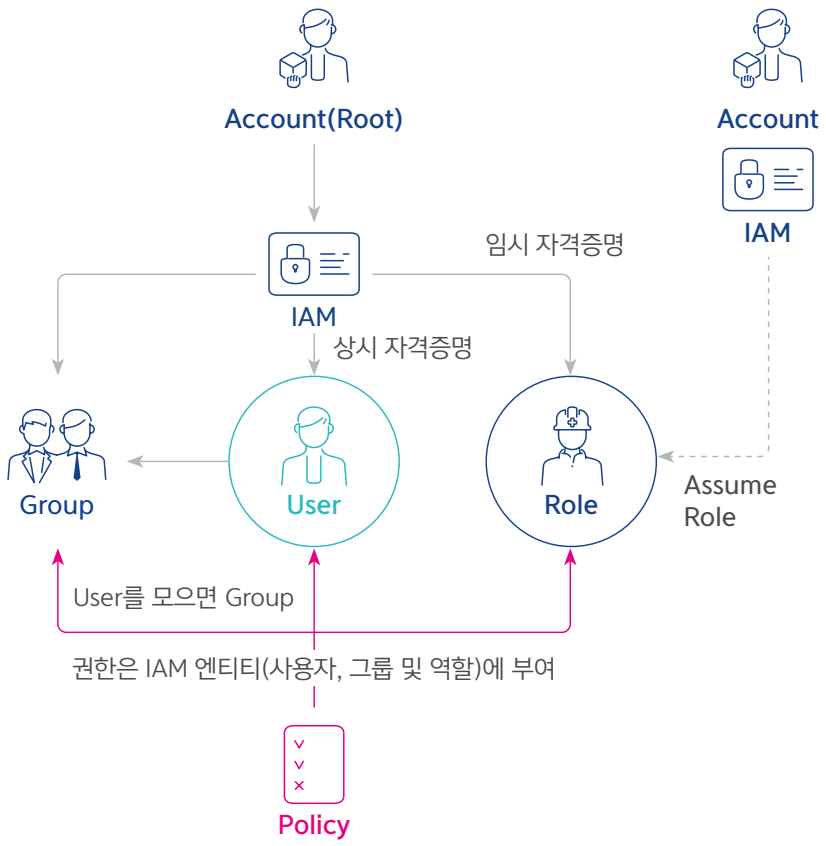
AWS IAM에서 실제 업무나 역할 별로 Policy를 연결해 사용할 경우, Group 단위로 권한을 관리할 것

Policy는 클라우드 리소스에 대한 권한을 'Json' 형식으로 기술한 규칙으로, 특정 Group이나 User에 Policy를 적용해 업무 용도에 따라 정책을 정의할 수 있다. Policy는 AWS 서비스와 리소스에 대한 '인가' 기능을 제공하는데, ▲Effect ▲Principal ▲Action ▲Resource ▲Condition의 요소로 이루어져 있다.

Policy의 각 구성 요소를 간단히 살펴보면, Effect는 명시된 정책에 대한 '허용' 또는 '거부'를 정의하는 항목이다. Principal은 '누가', Action은 '무엇을', Resource는 '적용할 대상', Condition은 '적용되는 조건'에 해당하는 항목이다. 이처럼 Policy의 다양한 구성 요소를 적절히 조합하면 정책을 상세하게 수립할 수 있다.

실제 업무나 역할 별로 Policy를 연결해 사용할 경우, User에 직접 Policy를 부여하는 것보다 Group 단위로 권한을 관리할 것을 권장한다. Group 단위로 권한을 관리하면, 담당 인력이 변경되어도 별도 Policy 수정 없이 Group 내에서 변경된 인력에 대한 User만 수정(생성 혹은 삭제)하면 되므로 관리가 용이하고 실수를 줄일 수 있다.

Role은 특정 서비스나 User에게 임시 권한을 부여할 때 사용하는데, Role을 이용해 임시 토큰을 발행 받아 정해진 리소스를 이용할 수 있는 권한을 부여할 수 있다.



[그림 2] IAM 개념도

AWS 아키텍처 구축 Tip 2

AWS CloudTrail, Access Advisor 등을 활용, 계정에 대한 여러가지 감사 활동을 통해 권한과 역할을 지속적으로 관리할 것

AWS 리소스에 대한 작업을 요청할 수 있는 보안 주체는 IAM User와 IAM Role로 구분할 수 있다. User는 상시 자격, Role은 임시 자격을 가진다. 만약 User가 가진 접근 키(Access Key)나 비밀 키(Secret Key)가 유출된다면 이는 User의 모든 권한을 탈취당하는 것과 다름이 없다. 따라서 각별히 주의하여 관리해야 한다.

이에 반해 Role은 Assume Role이라는 신뢰 관계를 통해 타계정에 User를 만들지 않고도 접근할 수 있다. Role은 별도 키 없이 임시 토큰으로 접근하는 개념이기 때문에 운영 상황에 따라 Role을 잘 활용한다면 보안 상 이점을 가질 수 있다.

IAM 보안 주체가 각 리소스에 접근하는 과정을 살펴보면, 최초에 누구인지 확인하는 신원 '인증' 절차를 거친 후 Policy를 통해 권한을 확인한다. 적절한 권한을 가지고 있는 것으로 확인되면 '인가'를 받아 해당 리소스에 접근이 가능하다. 더불어, 'AWS CloudTrail' 서비스를 이용해 앞서 인증과 인가를 받은 보안 주체가 언제, 무엇을, 어떻게 했는지 모든 API 요청의 처리 내역을 로깅해 계정 활동을 추적하고 관리 감독할 수 있다.

또한, 'Access Advisor'를 활용해 미사용 권한을 탐지하고 이를 계정 감사 활동의 일부로 활용할 수 있다. Access Advisor는 IAM에서 제공하는 기능으로 User나 Role이 최대 1년 동안 권한을 어떻게 사용했는지 확인할 수 있다. 예를 들어 어떤 User가 최근 접속 기록과 서비스 접근 기록이 없다면 보안 관점에서 해당 사용자는 이 권한이 필요 없다는 뜻으로 해석해 권한 수정이 필요하다고 판단할 수 있다.

이처럼 계정에 대한 여러가지 감사 활동을 통해 권한과 역할을 지속적으로 관리하고 개선점을 찾아 나가며 최소 권한의 원칙을 준수할 것을 권고한다.

지금까지 하나의 Account 내에 클라우드 리소스를 생성하고 그 리소스에 여러 사용자가 접근할 경우의 계정과 권한 관리 방안에 대해 살펴봤다. 다만, 클라우드 사용 규모가 크고 비용 관리가 중요한 경우에는 여러 개의 Account를 사용해야 한다.

AWS는 리소스의 가용성을 보장하고 필요 이상으로 프로비저닝(Provisioning) 하는 것을 방지하기 위해, 각 계정에서 최대로 이용할 수 있는 서비스 한도를 정해 놓았다. 그리고 AWS 사용료 청구의 최소 단위는 'Account'이기 때문에 여러 부서 다수의 사용자가 AWS에 접근하여 업무를 수행해야 한다면, 예산 및 비용 관리 관점에서도 Account를 나누어 사용하는 것이 좋다.

계정을 여러 개 사용할 경우 'AWS Organizations' 서비스를 이용해 통합 관리할 수 있다. AWS Organizations는 여러 계정을 그룹화해 관리하는 서비스로 OU(Organization Unit)이라는 개념을 통해 여러 계정을 개념적으로 묶어 관리할 수 있는데, 이는 윈도우 폴더와 유사하다고 이해하면 쉽다. 하나의 폴더 안에 여러 폴더나 파일이 있는 것처럼 OU 안에 여러 계정을 배치하거나 하위에 다른 OU를 생성할 수도 있다.

AWS 아키텍처 구축 Tip 3

- 다수의 사용자가 AWS에서 업무를 수행할 경우, Account를 나누어 사용
- 여러 계정을 사용하는 환경에서는 AWS Organizations 서비스를 적절히 활용

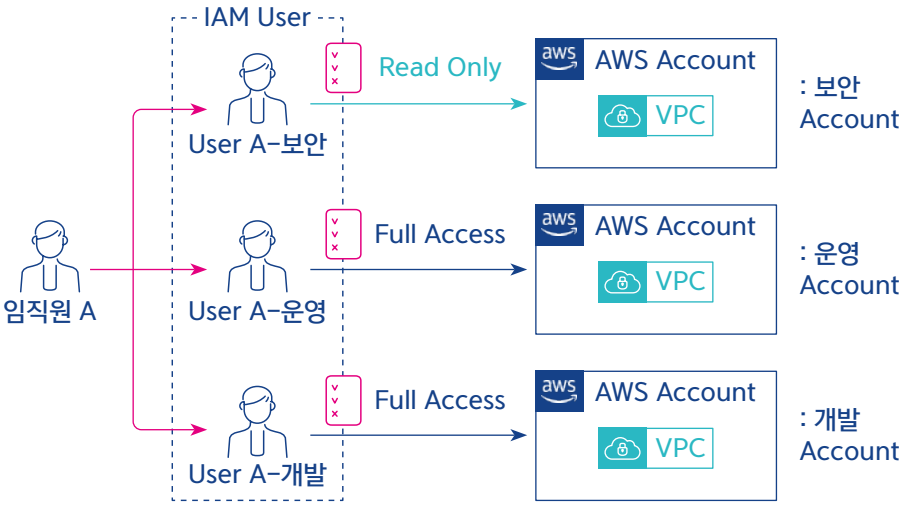
이렇게 구성하면 운영 측면에서 하위 계정들에 서비스 제어 정책(SCP)을 적용해 중앙 관리가 가능하며, 비용 측면에서 각 계정에서 발생한 사용료를 통합 결제할 수 있고 각 계정 별 세부 사용료 확인도 가능하다. 중앙 관리가 용이하고, 통합 결제를 통한 할인 혜택의 장점도 있기 때문에 여러 계정을 사용하는 환경에서는 AWS Organizations 서비스를 적절히 사용하는 것이 좋다.

이제 실제로 기업 환경에서 AWS를 이용할 때, 보안을 고려해 클라우드 아키텍처를 구성하고 각 사용자의 계정과 권한을 관리하는 방안에 대하여 알아보자. 크게 3단계로 나누어 계정 분리, 네트워크 설계, 서비스 적용의 단계로 살펴본다.

AWS에서 보안, 운영, 개발 등 환경을 분리하는 방법은 굉장히 다양하다. 환경 별로 Account를 분할할 수 있고, VPC(Virtual Private Cloud)를 분할할 수도 있으며, 나아가서는 서브넷(Subnet) 단위로 분할할 수도 있다. 어느 방식을 선택할지는 시스템 규모나 특성에 따라서 다르고 각 방식 모두 장단점이 존재한다. 하지만 추후 서비스의 확장성을 고려하고 운영과 비용 관리 측면에서의 장점을 극대화하기 위해서는 환경별로 Account를 분할하는 것이 가장 효과적인 구성이라 할 수 있다.

이처럼 환경 별로 Account를 모두 분리해서 사용하는 것은 초기 설정이 조금 복잡할 수 있다. 하지만, 운영 측면에서 권한 관리를 간결하게 할 수 있고 비용 관리 측면에서도 통합 결제 기능과 볼륨 할인 등 여러가지 장점이 존재하기 때문에 장기적인 관점에서는 가장 좋은 구성 방안이 될 수 있다.

환경 별로 Account를 분할하는 것이 운영과 비용 관리 측면에서 왜 효율적인지 알아보자. 예를 들어 회사에 보안, 운영, 개발 환경이 존재하고, 각 환경을 계정으로 분리하여 사용한다고 가정해본다.



[그림 3] 환경 별로 Account를 분할하는 경우 권한 부여 예시

AWS 아키텍처 구축 Tip 4

특정 임직원이 각 환경에 다른 권한이 필요하다면, 환경 별 Account에 개별 User를 생성하고 권한에 맞게 간결한 정책을 부여해 관리

특정 임직원이 운영 환경과 개발 환경에 대해서는 모든 작업을 허가하고 보안 환경은 참조 권한만 필요하다면, [그림 3]과 같이 각 환경 별 Account에 개별 User를 생성하고 권한에 맞게 간결한 정책을 부여해 관리할 수 있다. 이렇게 하면 Account와 시스템이 새롭게 추가되더라도 그에 맞는 User를 생성하고, 최소한의 권한을 부여해 관리할 수 있다.

만약 환경 별로 Account를 분리하지 않고 VPC를 분할해 구성했다면, 해당 User에게 부여할 정책을 리소스 별로 세세하게 만들어줘야 한다. 이 경우, 정책의 복잡성이 증가해 실수가 발생할 위험이 있고, 관리 측면에서도 어려움이 따른다. 따라서 최대한 Account를 세분화하여 설계하는 것이 잠재적인 실수를 최소화하는 가장 이상적인 방법이라 할 수 있다.

이제 환경 별로 Account를 분리하는 것이 비용 관점에서 어떤 장점을 갖는지 알아보자. 환경 별로 Account를 분리하면 보안, 운영, 개발 각 팀에서 Account을 별도로 사용하고, 앞서 소개한 AWS Organizations 서비스로 각 계정에서 발생한 사용료를 하나의 Master Account로 통합해 결제하게 된다. 기본적으로 AWS 사용료는 Account 단위로 청구되기 때문에 하나의 Account 안에 여러 서비스가 존재하는 경우, 각 서비스마다 발생하는 비용을 산정하기 쉽지 않다.

따라서 서비스 종류마다 Account를 나눠 놓으면, Account 당 청구서 1개가 발행되므로 단순하고 정확하게 관리할 수 있다. 또한 볼륨 할인도 Account 단위가 아닌 통합 결제 Account를 대상으로 적용할 수 있고, Account 간 크레딧이나 예약 인스턴스(Reserved Instances: RI) 공유 등 여러 경제적 이점이 있다.

앞선 내용들을 종합해보면, 가장 먼저 결제를 위한 Master Account를 생성하고, 같은 속성을 가진 서비스 별로 Account를 생성한다. 그리고 앞서 생성한 Master Account에 링크시켜 비용을 통합 관리할 수 있도록 한다.

그리고 하나의 서비스 안에 다수 사용자의 업무 및 역할이 연관되어 있을 시, 다시 역할 별로 Account를 분리한다. 이를테면, 하나의 서비스를 운영할 때에도 역할 별로 운영, QA, 개발 등 업무 구분이 있을 것이고 이 경우 업무 별로 Account를 모두 분리해 사용해야 한다는 것이다.

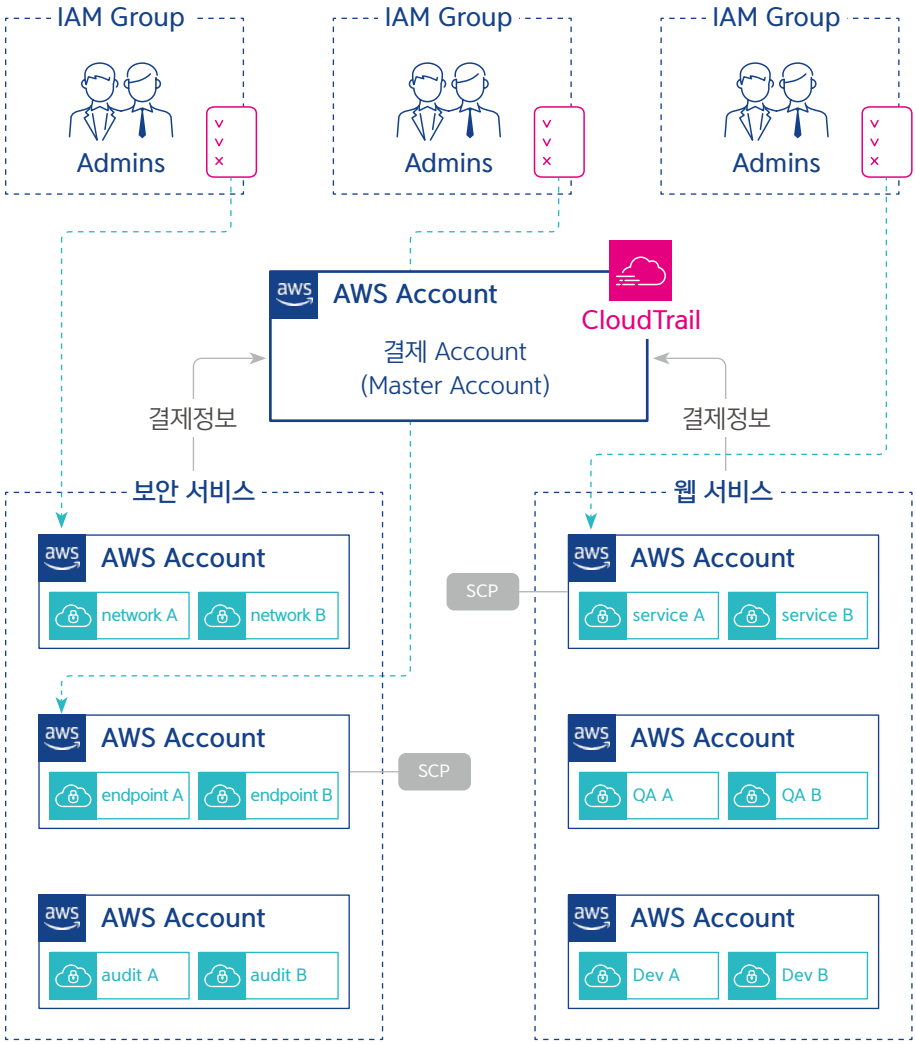
Account를 모두 분리했으면 실제 임직원의 업무 별로 User와 Group을 생성하고, 서비스 성격에 맞게 정책을 부여해 담당 업무 별로 최소 권한의 원칙에 따라 상세하게 권한을 관리한다.

이와 같은 절차로 계정 체계를 수립하고 각 사용자 별 권한을 알맞게 잘 부여했다면, 계정 활동 감사를 통한 보안 강화를 위해 AWS CloudTrail을 활성화한다. 또, 서비스

AWS 아키텍처 구축 Tip 5

서비스 종류마다 Account를 나눠 놓으면, Account 당 청구서 1개가 발행되므로 단순하고 정확하게 관리 가능

제어 정책(SCP)을 OU 또는 Account 별로 적용해 권한을 제한할 수 있다.



[그림 4] AWS 계정 및 권한 관리 방안

다음으로, 각 계정 내에서 네트워크를 설계하는 절차와 보안 영역 별 서비스 구성 절차를 차례로 살펴보자.

우선 네트워크를 설계하는 방법은 어떤 기능 단위로 구성할지에 따라, VPC나 서브넷 구성 방법을 다양하게 선택할 수 있다. 네트워크 설계 시, 가장 먼저 리전(Region)을 선택하고 서비스 성격에 맞는 네트워크 구성을 설계해 VPC, 서브넷 순서로 네트워크 영역을 구성한다.

다음으로, 설계한 네트워크 간 연계를 구성한다. 서로 다른 서브넷을 라우팅 테이블을 이용해 연결을 구성할 수 있고, 'VPC 피어링(VPC Peering)'으로 서로 다른 VPC의 연

AWS 아키텍처 구축 Tip 6

각 네트워크 연계 방안의 기능과 제약 사항을 파악해 용도에 맞는 연계를 구성

결을 구성할 수도 있다. 이 밖에 'VPC 엔드포인트(VPC Endpoint)'를 이용해 엔드포인트를 지원하는 서비스와 프라이빗한 연결 구성도 가능하다. 이와 같이 각 연계 방안의 기능과 제약 사항을 파악해 용도에 맞는 연계 구성을 할 수 있다.

Account 내부 네트워크 간 연계를 모두 마친 후에는 외부 네트워크와의 통신 구간 설계가 필요하다. 이 과정에서는 인터넷 연결을 위한 '인터넷 게이트웨이', 프라이빗 서브넷(Private Subnet)에서 인터넷과 통신하기 위한 'NAT(Network Address Translation) 게이트웨이', 프라이빗한 외부 연결을 위한 VPN 및 Direct Connect(전용선) 서비스 등을 이용할 수 있다.

내/외부 네트워크 통신 구성까지 모두 마친 후에는 서비스 연속성과 가용성을 고려하여 로드 밸런싱(Load Balancing)을 설계하고, 네트워크 트래픽을 다루는 여러 ISV(Independent Software Vendor) 솔루션들을 손쉽게 효율적으로 관리해주는 '게이트웨이 로드밸런서(Gateway Load Balancer: GWLB) 서비스 등을 활용할 수도 있다.

네트워크의 구역과 연계 설계 등 모든 프레임을 다 구성한 이후에는 실제 네트워크 트래픽을 제어할 수 있도록 서브넷팅과 라우팅, 시큐리티 그룹(Security Group)과 NACL(Network Access Control List) 등을 충분히 활용해 네트워크 트래픽 필터링을 통한 접근 제어를 설계하면 된다.

서비스 특성을 고려한 계정 분리와 권한 설정 그리고 네트워크 설계까지 모두 마쳤으면 이제 클라우드 아키텍처 구성을 위한 큰 프레임은 모두 완성되었다. 이제 이 완성된 프레임을 기반으로 각 요구사항에 맞는 서비스를 배치하면 풀 아키텍처를 완성할 수 있다.

비즈니스 요구사항이나 컴플라이언스 항목에 맞춰 인증 및 권한 관리, 접근 통제, 암호화 및 키 관리, 인프라 및 데이터 보호, 위험탐지 및 규정준수, 시스템 및 서비스 운영 관리 등 각 보안 영역 별 항목에 대응하는 서비스와 솔루션을 아키텍처 프레임 안에 배치 구성한다.

각 영역 별로 알맞은 서비스를 배치하고 보안 설정까지 모두 완료하면, 보안을 고려한 클라우드 아키텍처 구성을 모두 마치게 된다.

도입 사례로 알아보는 AhnLab Cloud의 장점

AhnLab Cloud는 최초 클라우드 도입하는 시점부터 현황 분석을 통해 Account 분리, 네트워크 설계, 그리고 서비스 배치의 모든 과정 속 모든 요소에 보안을 고려하여 진행한다는 점에서 기존 MSP와 차별화된다. 고객의 업무를 파악하고 시스템 사용 및 관리 현황을 상세하게 분석함으로써 고객사 환경에 최적화된 클라우드 환경을 제안하고 가

AWS 아키텍처 구축 Tip 7

네트워크 프레임 구성 후, 트래픽 필터링을 통한 접근 제어를 설계하고 요구사항에 맞는 서비스를 배치해 풀 아키텍처 완성

AhnLab Cloud 고객 사례

1. 아키텍처 현황 상세 분석
2. 보안 고려한 서비스 구성
3. 개선점 & 변경사항 도출
4. Account를 나누고 최소 권한 부여
5. 적합한 네트워크 설계
6. AWS 보안 서비스와 ISV 솔루션 적재적소에 구성

☞ 안정성 ↑ 자원 투입 ↓

장 안전한 클라우드 환경을 제공하고 있다.

소프트웨어 개발사인 A기업은 예전부터 클라우드를 활용해 다양한 서비스를 개발 및 테스트해오고 있었다. 이를 체계적인 계획에 따라 관리하기 보다는 상시적으로 필요에 따라 Account를 생성해 사용하고 있었다. 하지만, 운영 대상과 준수해야 하는 컴플라이언스가 늘어나면서 여러 도전과제를 마주한 상황이었다. 그리고 이를 해결하기 위해 AhnLab Cloud 아키텍처 방법론과 보안 프레임워크를 활용해 전체적인 서비스 아키텍처를 재구성했다.

안랩은 먼저 클라우드 아키텍처 현황을 면밀히 분석하고, 거버넌스(Governance), 보안 기술, 운영 관리 관점에서 여러 보안 요소들을 고려해 서비스를 구성했다. 먼저, 현 상황을 분석해 개선점을 도출하고 개선 및 변경해야 할 사항들을 선정했다. 이후, 조직 구성원이 수행하는 서비스와 업무에 따라 Account를 나누고 최소한의 권한을 부여했다. 서비스에 가장 적합한 네트워크를 설계하고 리소스를 배치했으며, 내/외부 위협 대응을 위해 AWS 보안 서비스와 ISV 솔루션을 적재적소에 구성했다.

결론적으로, 클라우드 환경은 최초 설계부터 Account와 권한을 적절하게 분리하고 보안을 고려해 네트워크를 설계하면 추후 서비스 추가나 변경이 있더라도 안정성을 유지하면서 구성 관리에 투입되는 자원을 최소화할 수 있다. AhnLab Cloud를 활용할 경우, 고객은 고객은 안랩이라는 단 하나의 컨택 포인트로 클라우드 보안 컨설팅부터 아키텍처 설계, 구축, 운영 그리고 클라우드 보안까지 모두 해결해 클라우드를 안전하고 편리하게 사용할 수 있다.

AhnLab

경기도 성남시 분당구 판교역로 220 (우)13493
홈페이지: www.ahnlab.com
대표전화: 031-722-8000 팩스: 031-722-8901