AWS 백서

Amazon Elastic File System을 사용하여 파일 데이터 암호화

Amazon Elastic File System을 사용하여 파일 데이터 암호화: AWS 백서

Table of Contents

Amazon Elastic File System을 사용하여 파일 데이터 암호화

게시 날짜: 2021년 2월 22일(문서 기록 및 기여자)

요약

AWS 보안은 최우선 사항이며, 고객에게 기업에 보안 작업을 최우선으로 수행할 수 있는 도구를 제공합니다. 정부 규정 및 업계 또는 회사의 규정 준수 정책은 암호화 정책, 암호화 알고리즘 및 적절한 키 관리를 사용하여 서로 다른 분류의 데이터를 보호해야 할 수 있습니다. 이 문서에서는 Amazon Elastic File System(Amazon EFS)을 암호화하는 모범 사례를 간략히 소개합니다.

소개

Amazon Elastic File System(Amazon EFS)는 클라우드에서 간단하고 확장 가능하며 가용성이 높고 내구성이 우수한 공유 파일 시스템을 제공합니다. Amazon EFS를 사용하여 생성하는 파일 시스템은 탄력적이므로 데이터를 추가 및 제거할 때 자동으로 확장 및 축소할 수 있습니다. 또한 페타바이트급까지 크기를 확장할 수 있으며, 여러 가용 영역(AZ)에 있는 제한 없는 수의 스토리지 서버에 데이터를 분산시킬 수 있습니다.

이러한 파일 시스템에 저장된 데이터는 Amazon EFS를 사용하여 저장 및 전송 중에 암호화할 수 있습니다. 저장된 데이터를 암호화하려면 AWS 관리 콘솔 또는 AWS Command Line Interface(AWS CLI)를 통해 암호화된 파일 시스템을 생성할 수 있습니다. 또는 Amazon EFS API 또는 AWS SDK 중 하나를 통해 프로그래밍 방식으로 암호화된 파일 시스템을 생성할 수 있습니다.

저장된 데이터의 암호화를 위해 Amazon EFS는 AWS Key Management Service(AWS KMS)와 통합하여 키를 관리합니다. 파일 시스템을 탑재하고 전송 계층 보안(TLS)을 통해 모든 NFS 트래픽을 전송하여 전송 중인 데이터를 암호화할 수도 있습니다.

이 백서에서는 Amazon EFS의 암호화 모범 사례를 간략하게 설명합니다. 클라이언트 연결 계층에서 전송 중인 데이터를 암호화하는 방법과 AWS 관리 콘솔 및 AWS CLI에서 암호화된 파일 시스템을 생성하는 방법이 설명되어 있습니다.

> **Note**
>
> API와 SDK를 사용하여 암호화된 파일 시스템을 만드는 작업은 이 백서의 범위를 벗어납니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 Amazon EFS 사용 설명서 또는 SDK 설명서의 Amazon EFS API를 참조하세요.

기본 개념 및 용어

이 단원에서는 이 백서에서 참조하는 개념과 용어를 정의합니다.

- Amazon Elastic File System(Amazon EFS) – AWS 클라우드에서 단순하고 확장 가능한 공유 파일 스토리지를 제공하는 가용성과 내구성이 우수한 서비스입니다. Amazon EFS는 표준 파일 시스템 인터페이스와 파일 시스템 의미 체계를 제공합니다. 여러 가용 영역에 있는 무제한의 스토리지 서버에 사실상 무제한의 데이터를 저장할 수 있습니다.
- AWS Identity and Access Management(IAM)– AWS 서비스 API에 대한 세분화된 액세스를 안전하게 제어할 수 있게 해 주는 서비스입니다. 정책은 개별 사용자, 그룹 및 역할에 대한 액세스를 제한하기 위해 만들어지고 사용됩니다. IAM 콘솔을 통해 AWS KMS 키를 관리할 수 있습니다.
- AWS KMS – 데이터 암호화에 사용하는 암호화 키인 고객 마스터 키(CMK)를 쉽게 생성하고 제어할 수 있게 해 주는 관리형 서비스입니다. AWS KMS CMK는 중국(베이징) 및 중국(닝샤) 리전을 제외하고 FIPS 140-2 Cryptographic Module Validation Program으로 검증된 하드웨어 보안 모듈(HSM)에 의해 보호됩니다. AWS KMS는 데이터를 암호화하는 다른 AWS 서비스와 통합됩니다. 또한 AWS CloudTrail과 완벽하게 통합되어 사용자를 대신해 AWS KMS에서 수행한 API 호출 로그를 제공하므로 조직에 적용되는 규정 준수 또는 규제 요구 사항을 충족하는 데 도움이 될 수 있습니다.
- 고객 마스터 키(CMK) – 키 계층 구조의 최상위를 나타냅니다. 데이터를 암호화하고 해독하는 데 필요한 키 자료가 포함되어 있습니다. AWS KMS에서 이 키 자료를 생성하거나 생성한 다음 AWS KMS로 가져올 수 있습니다. CMK는 AWS 계정 및 AWS 리전에 따라 다르며 고객 관리형 또는 AWS 관리형 계정일 수 있습니다.
- AWS 관리형 CMK – AWS에서 사용자를 대신하여 생성하는 CMK입니다. AWS 관리형 CMK는 통합 AWS 서비스의 리소스에 대해 암호화를 활성화할 때 생성됩니다. AWS 관리형 CMK 키 정책은 AWS에서 관리하며 변경할 수 없습니다. AWS 관리형 CMK를 생성하거나 저장할 때는 요금이 부과되지 않습니다.
- 고객 관리형 CMK – AWS 관리 콘솔 또는 API, AWS CLI 또는 SDK를 사용하여 생성하는 CMK입니다. CMK를 보다 세부적으로 제어해야 하는 경우 고객 관리형 CMK를 사용할 수 있습니다.
- KMS 키 정책 – 고객 관리형 CMK에 대한 액세스를 제어하는 리소스 정책입니다. 고객은 키 정책 또는 IAM 정책과 키 정책의 조합을 사용하여 이러한 권한을 정의합니다. 자세한 내용은 AWS KMS 개발자 안내서의 액세스 관리 개요를 참조하세요.
- 데이터 키 – AWS KMS 외부의 데이터를 암호화하기 위해 AWS KMS에서 생성한 암호화 키입니다. AWS KMS를 사용하여 권한 있는 엔터티(사용자 또는 서비스)가 CMK로 보호되는 데이터 키를 가져올 수 있습니다.
- 전송 계층 보안(TLS) – 보안 소켓 계층(SSL)의 후속 제품인 TLS는 네트워크를 통해 교환되는 정보를 암호화하는 데 필수인 암호화 프로토콜입니다.

- EFS 탑재 헬퍼 – EFS 파일 시스템의 탑재를 단순화하는 데 사용되는 Linux 클라이언트 에이전트(`amazon-efs-utils`)입니다. TLS 터널을 통해 모든 NFS 트래픽을 설정, 유지 관리 및 라우팅하는 데 사용할 수 있습니다.

기본 개념 및 용어에 대한 자세한 내용은 AWS KMS 개발자 안내서의 AWS Key Management Service 개념을 참조하세요.

유휴 데이터 암호화

AWS는 업계 표준 AES-256 암호화 알고리즘을 사용하여 저장된 모든 데이터와 메타데이터를 암호화하는 암호화된 파일 시스템을 생성하는 도구를 제공합니다. 암호화된 파일 시스템은 암호화 및 암호 해독을 자동으로 투명하게 처리하도록 설계되었으므로 애플리케이션을 수정하지 않아도 됩니다. 조직에 저장된 데이터 및 메타데이터의 암호화를 요구하는 기업 또는 규정 정책이 적용되는 경우, 암호화된 파일 시스템을 생성하는 것이 좋습니다.

주제

키 관리

Amazon EFS는 암호화된 파일 시스템의 암호화 키를 관리하는 AWS KMS와 통합되어 있습니다. 또한 AWS KMS는 Amazon Simple Storage Service(Amazon S3), Amazon Elastic Block Store(Amazon EBS), Amazon Relational Database Service(Amazon RDS), Amazon Aurora, Amazon Redshift, Amazon WorkMail, WorkSpaces 등 다른 AWS 서비스의 암호화를 지원합니다. 파일 시스템 콘텐츠를 암호화하기 위해 Amazon EFS는 XTS 모드 및 256비트 키(XTS-AES-256)와 함께 고급 암호화 표준 알고리즘을 사용합니다.

암호화 정책을 채택하여 저장된 데이터를 보호하는 방법을 고려할 때 대답해야 할 세 가지 중요한 질문이 있습니다. 이러한 질문은 Amazon EBS와 같은 관리형 및 비관리형 서비스에 저장된 데이터에 대해서도 동일하게 적용됩니다.

키는 어디에 저장됩니까?

AWS KMS는 필요할 때 검색할 수 있도록 내구성이 우수한 스토리지에 암호화된 형식으로 마스터 키를 저장합니다.

키는 어디에 사용됩니까?

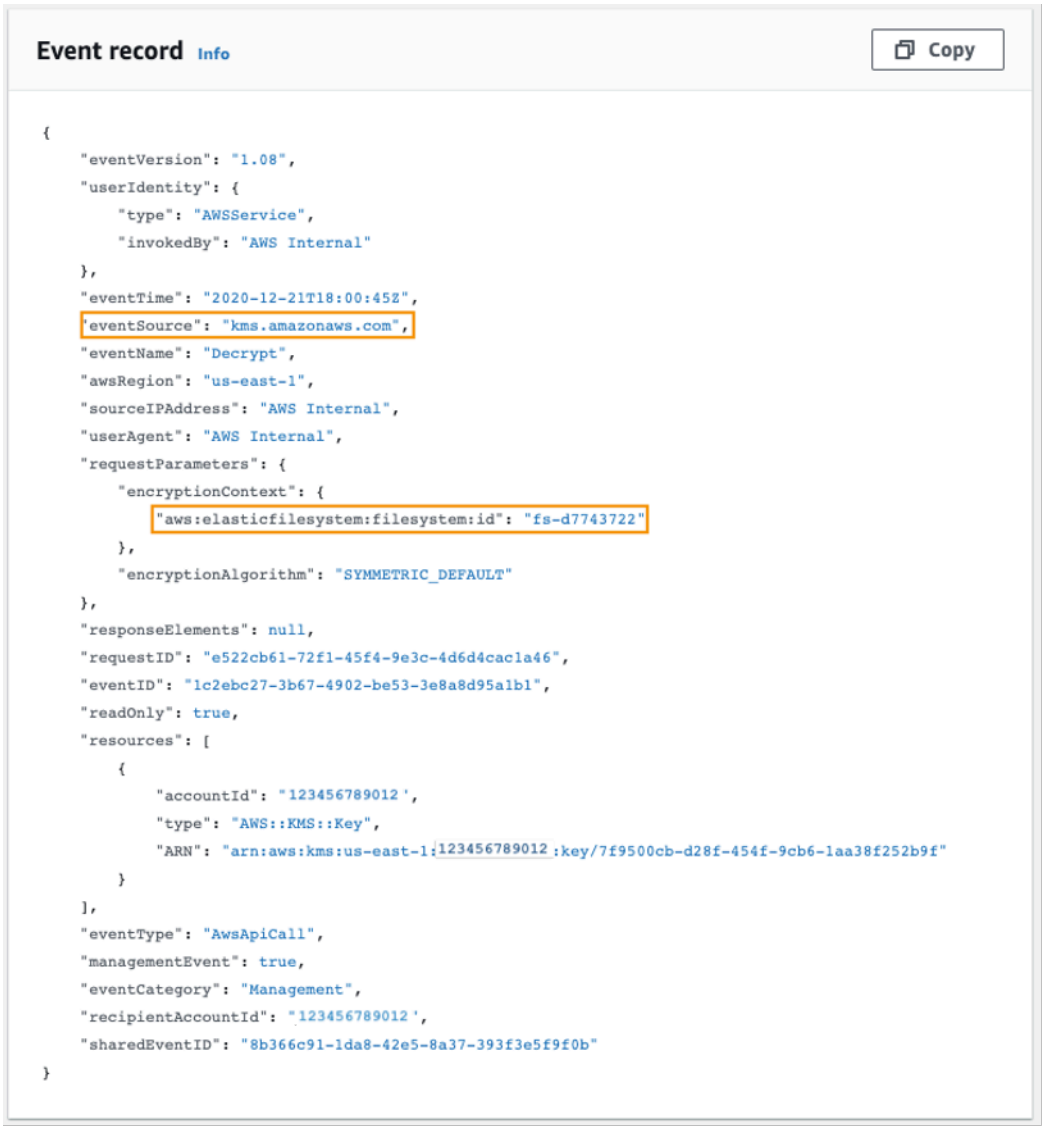
암호화된 Amazon EFS 파일 시스템을 사용하면 파일 시스템을 탑재하는 클라이언트에 아무런 영향을 미치지 않습니다. 데이터는 디스크에 기록되기 전에 암호화되고 클라이언트가 읽기 요청을 실행한 후 해독되므로 모든 암호화 작업은 EFS 서비스 내에서 실행됩니다.

키를 사용할 수 있는 사람은 누구입니까?

AWS KMS 키 정책은 암호화 키에 대한 액세스를 제어합니다.

IAM 정책과 결합하여 또 다른 제어 계층을 제공하는 것이 좋습니다. 각 키에는 키 정책이 있습니다. 키가 AWS 관리형 CMK인 경우 AWS에서 키 정책을 관리합니다. 키가 고객 관리형 CMK인 경우 사용자가 키 정책을 관리합니다. 이러한 키 정책은 CMK에 대한 액세스를 제어하는 기본적인 방법입니다. 키 사용 및 관리를 제어하는 권한을 정의합니다.

Amazon EFS를 사용하여 암호화된 파일 시스템을 생성하는 경우, Amazon EFS에게 사용자 대신 CMK를 사용할 수 있는 액세스 권한을 부여합니다. Amazon EFSS가 사용자를 대신하여 AWS KMS에 수행하는 호출은 AWS 계정에서 시작된 것처럼 CloudTrail 로그에 표시됩니다. 다음 스크린샷은 Amazon EFS에서 수행한 KMS Decrypt 호출에 대한 샘플 CloudTrail 이벤트를 보여 줍니다.



KMS Decrypt에 대한 CloudTrail 로그

AWS KMS 및 암호화 키에 대한 액세스를 관리하는 방법에 대한 자세한 내용은 AWS KMS 개발자 안내서의 AWS KMS CMK에 대한 액세스 관리를 참조하세요.

AWS KMS로 암호화를 관리하는 방법에 대한 자세한 내용은 AWS KMS 암호화 세부 정보 백서를 참조하세요.

관리자 IAM 사용자 및 그룹을 만드는 방법에 대한 자세한 내용은 IAM 사용 설명서의 첫 번째 IAM 관리 사용자 및 그룹 생성을 참조하세요.

암호화된 파일 시스템 생성

AWS 관리 콘솔, AWS CLI, Amazon EFS API 또는 AWS SDK를 사용하여 암호화된 파일 시스템을 생성할 수 있습니다. 파일 시스템을 생성할 때만 암호화를 활성화할 수 있습니다.

Amazon EFS는 키 관리를 위해 AWS KMS와 통합되며 CMK를 사용하여 파일 시스템을 암호화합니다. 파일 이름, 디렉터리 이름 및 디렉터리 콘텐츠와 같은 파일 시스템 메타데이터는 AWS 관리형 CMK를 사용하여 암호화되고 해독됩니다.

파일 콘텐츠 또는 파일 데이터는 선택한 CMK를 사용하여 암호화되고 해독됩니다. CMK는 다음 세 가지 형식 중 하나가 될 수 있습니다.

- Amazon EFS용 AWS 관리형 CMK
- AWS 계정의 고객 관리형 CMK
- 다른 AWS 계정의 고객 관리형 CMK

조직에는 CMK에 대한 액세스 제어 및 사용 정책 외에도 생성, 교체, 삭제에 대한 완전한 제어가 필요한 회사 또는 규제 정책이 적용될 수 있습니다. 그렇다면 고객 관리형 CMK를 사용하는 것이 좋습니다. 다른 시나리오에서는 AWS 관리형 CMK를 사용할 수 있습니다.

모든 사용자에게는 Amazon EFS용 AWS 관리형 CMK가 있으며, 별칭은 `aws/elasticfilesystem`입니다. AWS에서 이 CMK의 키 정책을 관리하므로 사용자가 변경할 수 없습니다. AWS 관리형 CMK를 생성하고 저장하는 데 드는 비용은 없습니다.

고객 관리형 CMK를 사용하여 파일 시스템을 암호화하려는 경우, 소유하고 있는 고객 관리형 CMK의 키 별칭을 선택합니다. 또는 다른 계정에서 소유한 고객 관리형 CMK의 Amazon 리소스 이름(ARN)을 입력해도 됩니다. 소유한 고객 관리형 CMK를 사용하여 키 정책 및 키 부여를 통해 키를 사용할 수 있는 서비스 및 사용자를 제어할 수 있습니다.

또한 키에 대한 액세스를 비활성화, 다시 활성화, 삭제 또는 취소할 시기를 선택하여 이러한 키의 수명 및 교체를 제어할 수 있습니다. 다른 AWS 계정의 키에 대한 액세스 관리에 대한 자세한 내용은 AWS KMS 개발자 안내서의 키 정책 변경을 참조하세요.

고객 관리형 CMK를 관리하는 방법에 대한 자세한 내용은 AWS KMS 개발자 안내서의 고객 마스터 키(CMK)를 참조하세요.

다음 단원에서는 AWS 관리 콘솔과 AWS CLI를 사용하여 암호화된 파일 시스템을 생성하는 방법에 대해 설명합니다.

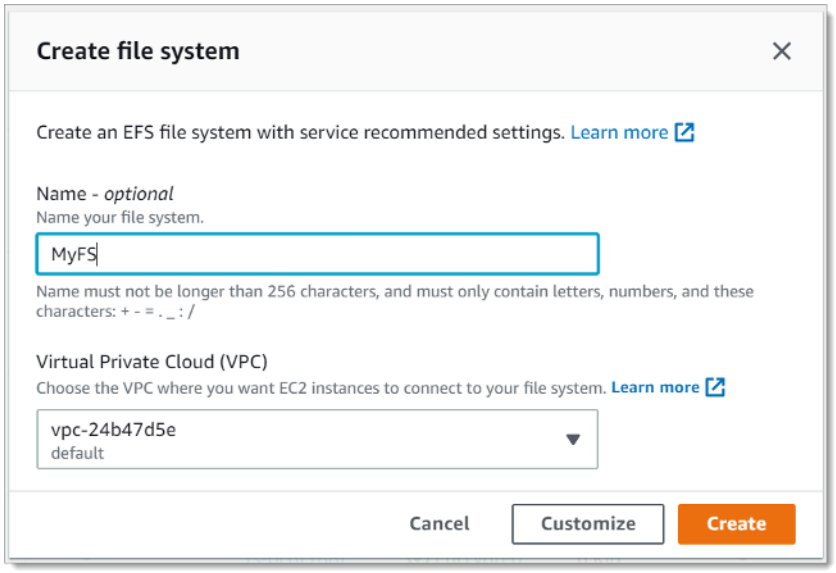
AWS 관리 콘솔을 사용하여 암호화된 파일 시스템 생성

다음 절차에 따라 AWS 관리 콘솔을 사용하여 암호화된 Amazon EFS 파일 시스템을 생성합니다.

1단계. 파일 시스템 설정 구성

이 단계에서는 수명 주기 관리, 성능 및 처리량 모드, 저장된 데이터의 암호화 등 일반 파일 시스템의 설정을 구성합니다.

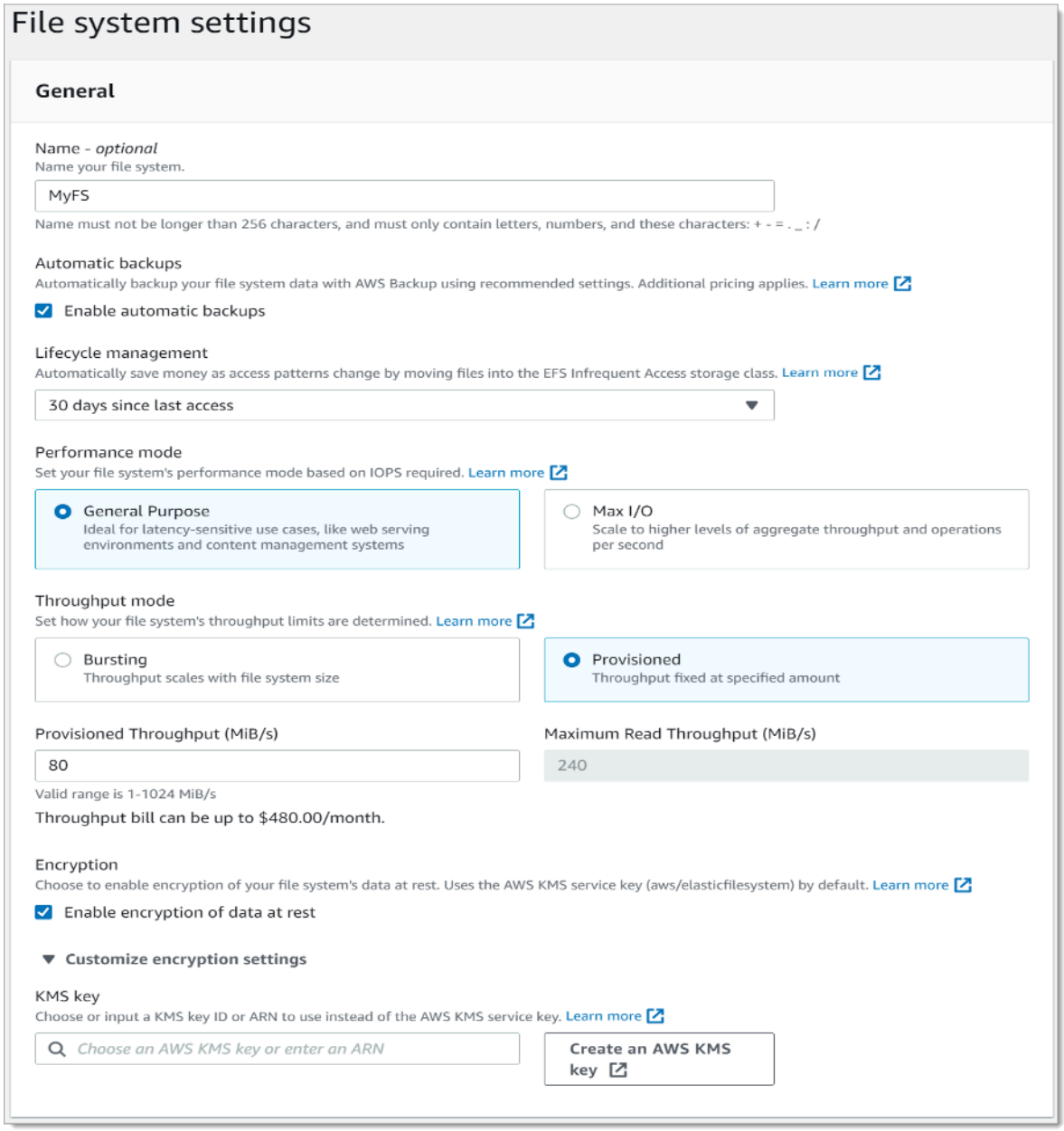
1. AWS 관리 콘솔에 로그인한 후 Amazon EFS 콘솔을 엽니다.
2. 파일 시스템 생성(Create file system)을 선택하여 파일 시스템 생성(Create file system) 대화 상자를 엽니다. 기본 암호화 활성화를 포함하는 권장 설정을 사용하여 파일 시스템을 생성하는 방법에 대한 자세한 내용은 Amazon EFS 파일 시스템 생성을 참조하세요.



EFS 파일 시스템 생성

3. (선택 사항) 서비스 권장 설정을 사용하여 파일 시스템을 만드는 대신 사용자 지정된 파일 시스템을 만들려면 사용자 지정(Customize)을 선택합니다.

   파일 시스템 설정 페이지가 표시됩니다.



EFS 파일 시스템 생성: 일반 설정

4. 일반(General) 설정에 다음 세부 정보를 입력합니다.
   - (선택 사항) 파일 시스템의 이름(Name)을 입력합니다.
   - 자동 백업(Automatic backups)은 기본적으로 설정되어 있습니다. 확인란을 선택 취소하여 자동 백업을 해제할 수 있습니다. 자세한 내용은 Amazon EFS와 함께 AWS Backup 사용을 참조하세요.
   - 수명 주기 관리(Lifecycle management) 정책을 선택합니다. Amazon EFS 수명 주기 관리는 파일 시스템에 대한 비용 효율적인 파일 스토리지를 자동으로 관리합니다. 수명 주기 관리가 활성화되

면 설정된 기간 동안 액세스하지 않은 파일을 Infrequent Access(IA) 스토리지 클래스로 마이그레이션합니다. 수명 주기 정책을 사용하여 해당 기간을 정의합니다. 수명 주기 관리를 사용하지 않으려면 없음(None)을 선택합니다. 자세한 내용은 Amazon EFS 사용 설명서의 EFS 수명 주기 관리를 참조하세요.

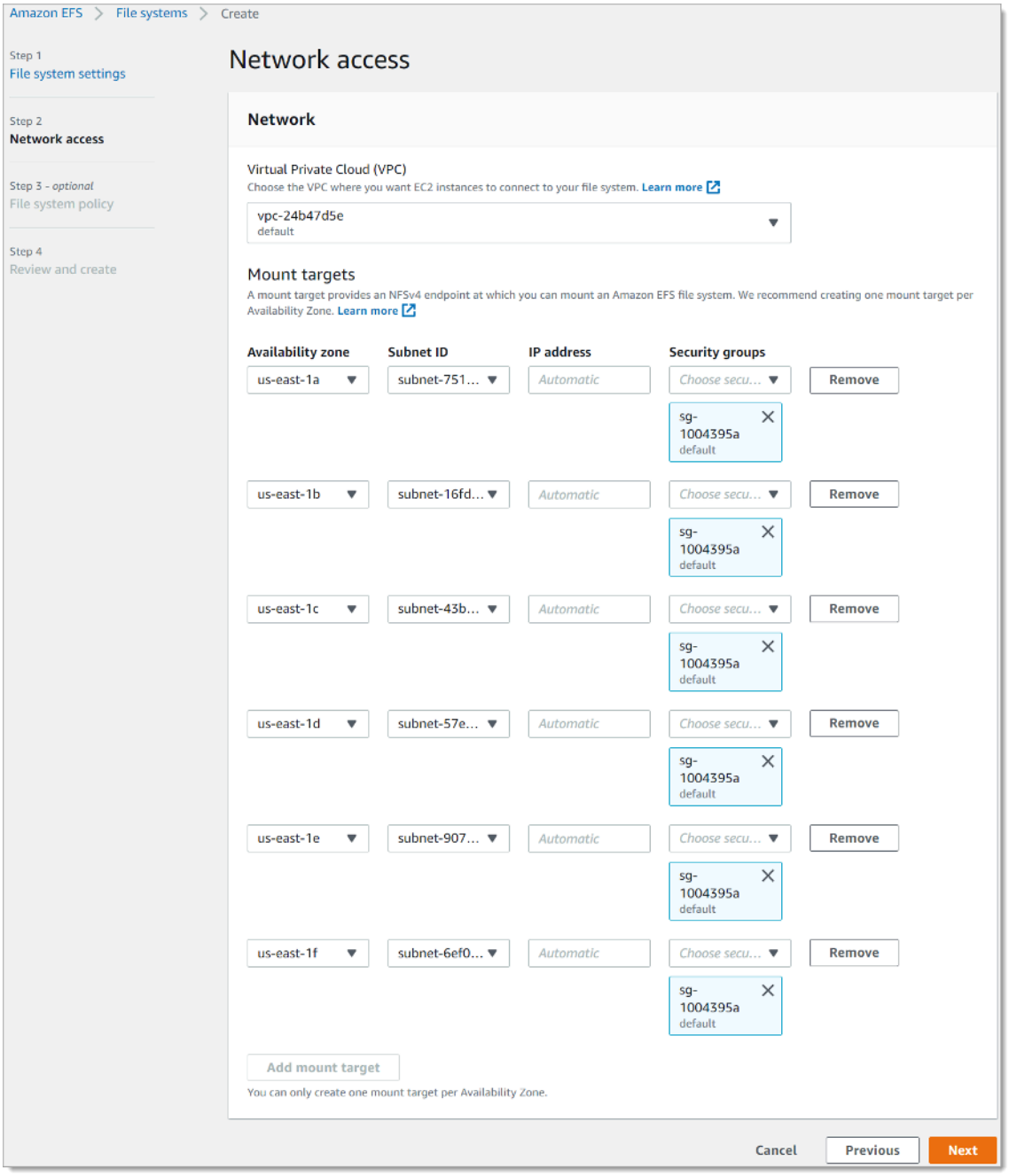
- 성능 모드(Performance mode)를 선택합니다(기본 범용 모드(General Purpose mode) 또는 최대 I/O(Max I/O)). 자세한 내용은 Amazon EFS 사용 설명서의 성능 모드를 참조하세요.
- 처리량 모드(Throughput mode)를 선택합니다(기본 버스팅 모드(Bursting mode) 또는 프로비저닝 모드(Provisioned mode)).
- 프로비저닝(Provisioned)을 선택한 경우 (프로비저닝된 처리량(Mib/s)(Provisioned Throughput (MiB/s)) 필드가 표시됩니다. 파일 시스템에 프로비저닝할 처리량을 입력합니다. 처리량을 입력하면 콘솔이 필드 옆에 월별 예상 비용을 표시합니다. 자세한 내용은 Amazon EFS 사용 설명서의 처리량 모드를 참조하세요.
- 암호화(Encryption)의 경우, 저장된 데이터 암호화가 기본적으로 활성화되어 있습니다. 기본적으로 AWS Key Management Service(AWS KMS) EFS 서비스 키(`aws/elasticfilesystem`)를 사용합니다. 암호화에 사용할 다른 KMS 키를 선택하려면 암호화 설정 사용자 지정을 확장하고 목록에서 키를 선택합니다. 또는 사용하려는 KMS 키에 대한 KMS 키 ID 또는 Amazon 리소스 이름(ARN)을 입력합니다.

  새 키를 만들어야 하는 경우 AWS KMS 키 생성(Create an AWS KMS key)을 선택하여 AWS KMS 콘솔을 시작하고 새 키를 만듭니다.

5. (선택 사항) 태그 추가(Add tag)를 선택하여 파일 시스템에 키-값 쌍을 추가합니다.
6. 다음(Next)을 선택하여 구성 프로세스의 네트워크 액세스(Network Access) 단계를 계속합니다.

2단계. 네트워크 액세스 구성

이 단계에서는 Virtual Private Cloud(VPC) 및 탑재 대상을 포함하여 파일 시스템의 네트워크 설정을 구성합니다. 각 탑재 대상에 대해 가용 영역, 서브넷, IP 주소 및 보안 그룹을 설정합니다.



EFS 파일 시스템 생성: 네트워크 액세스

1. EC2 인스턴스를 파일 시스템에 연결할 Virtual Private Cloud(VPC)를 선택합니다. 자세한 내용은 Amazon EFS 사용 설명서의 파일 시스템 네트워크 액세스 가능성 관리를 참조하세요.
   - 가용 영역(Availability zone) – 기본적으로 탑재 대상은 AWS 리전의 가용 영역별로 하나씩 구성됩니다. 특정 가용 영역에 탑재 대상을 사용하지 않으려면 제거(Remove)를 선택하여 해당 영역에서 탑재 대상을 삭제합니다. 파일 시스템에 액세스하려는 모든 가용 영역에 탑재 대상을 만듭니다. 이 작업에 비용이 들지 않습니다.

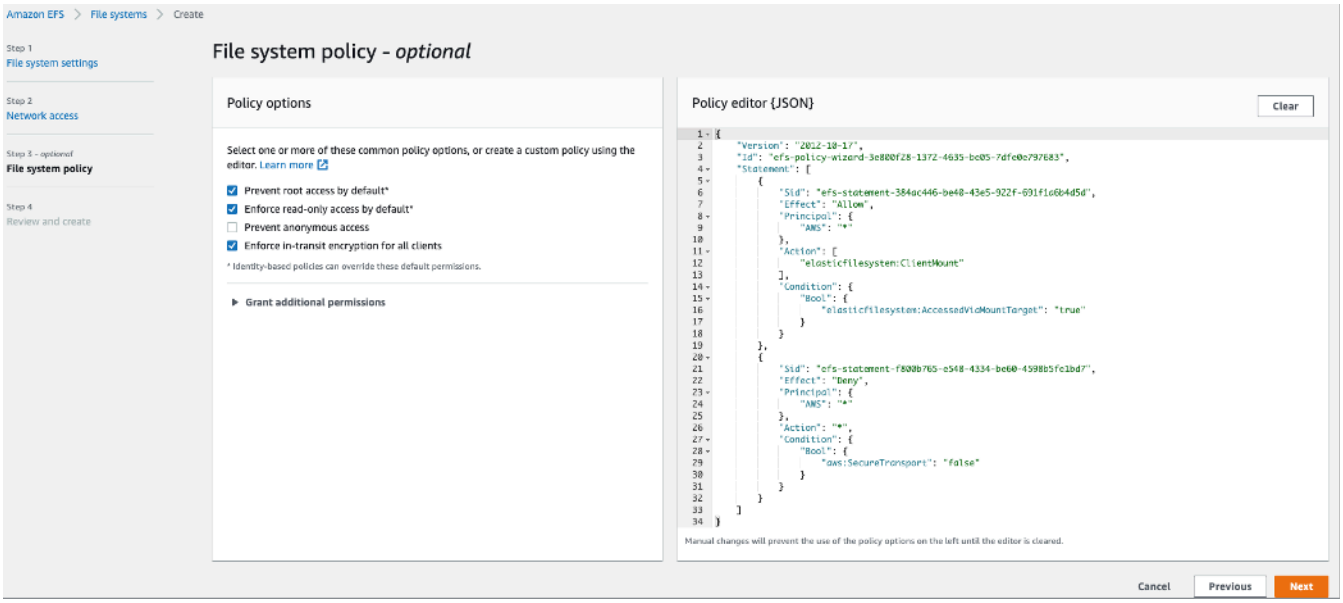
- 서브넷 ID(Subnet ID) – 가용 영역의 사용 가능한 서브넷 중에서 선택합니다. 기본 서브넷이 미리 선택되어 있습니다. 모범 사례에 따라 선택한 서브넷이 퍼블릭 또는 프라이빗인지 확인하는 것이 가장 좋습니다.
- IP 주소(IP Address) – 기본적으로 Amazon EFS는 서브넷의 사용 가능한 주소에서 IP 주소를 자동으로 선택합니다. 또는 서브넷에 있는 특정 IP 주소를 입력할 수 있습니다. 탑재 대상은 단일 IP 주소를 사용하지만 중복된 고가용성 네트워크 리소스입니다.
- 보안 그룹(Security groups) – 탑재 대상에 하나 이상의 보안 그룹을 지정할 수 있습니다. 모범 사례에 따라 보안 그룹이 EFS 탑재용(NFS 포트 2049)으로만 사용되고 인바운드 규칙이 다른 VPC CIDR 블록 범위의 포트 2049만 허용하거나 보안 그룹을 EFS에 액세스해야 하는 리소스의 소스로 사용하는 것이 가장 좋습니다. 자세한 내용은 Amazon EFS 사용 설명서의 Amazon EC2 인스턴스 및 탑재 대상에 보안 그룹 사용을 참조하세요.

  다른 보안 그룹을 추가하거나 보안 그룹을 변경하려면 보안 그룹 선택(Choose security groups)을 선택하고 목록에서 다른 보안 그룹을 추가합니다. 기본 보안 그룹을 사용하지 않으려면 삭제하면 됩니다. 자세한 내용은 Amazon EFS 사용 설명서의 보안 그룹 생성을 참조하세요.

2. 탑재 대상이 없는 가용 영역에 탑재 대상을 만들려면 탑재 대상 추가(Add mount target)를 선택합니다. 탑재 대상이 각 가용 영역에 대해 구성된 경우 이 선택 항목을 사용할 수 없습니다.
3. 다음(Next)을 선택하여 계속 진행합니다. 파일 시스템 정책(File system policy) 페이지가 표시됩니다.

3단계. 파일 시스템 정책 생성

이 단계에서는 파일 시스템에 대한 NFS 클라이언트 액세스를 제어하는 파일 시스템 정책을 만듭니다. EFS 파일 시스템 정책은 파일 시스템에 대한 NFS 클라이언트 액세스를 제어하는 데 사용하는 IAM 리소스 정책입니다. 자세한 내용은 Amazon EFS 사용 설명서의 IAM을 사용하여 Amazon EFS에 대한 NFS 액세스 제어를 참조하세요.

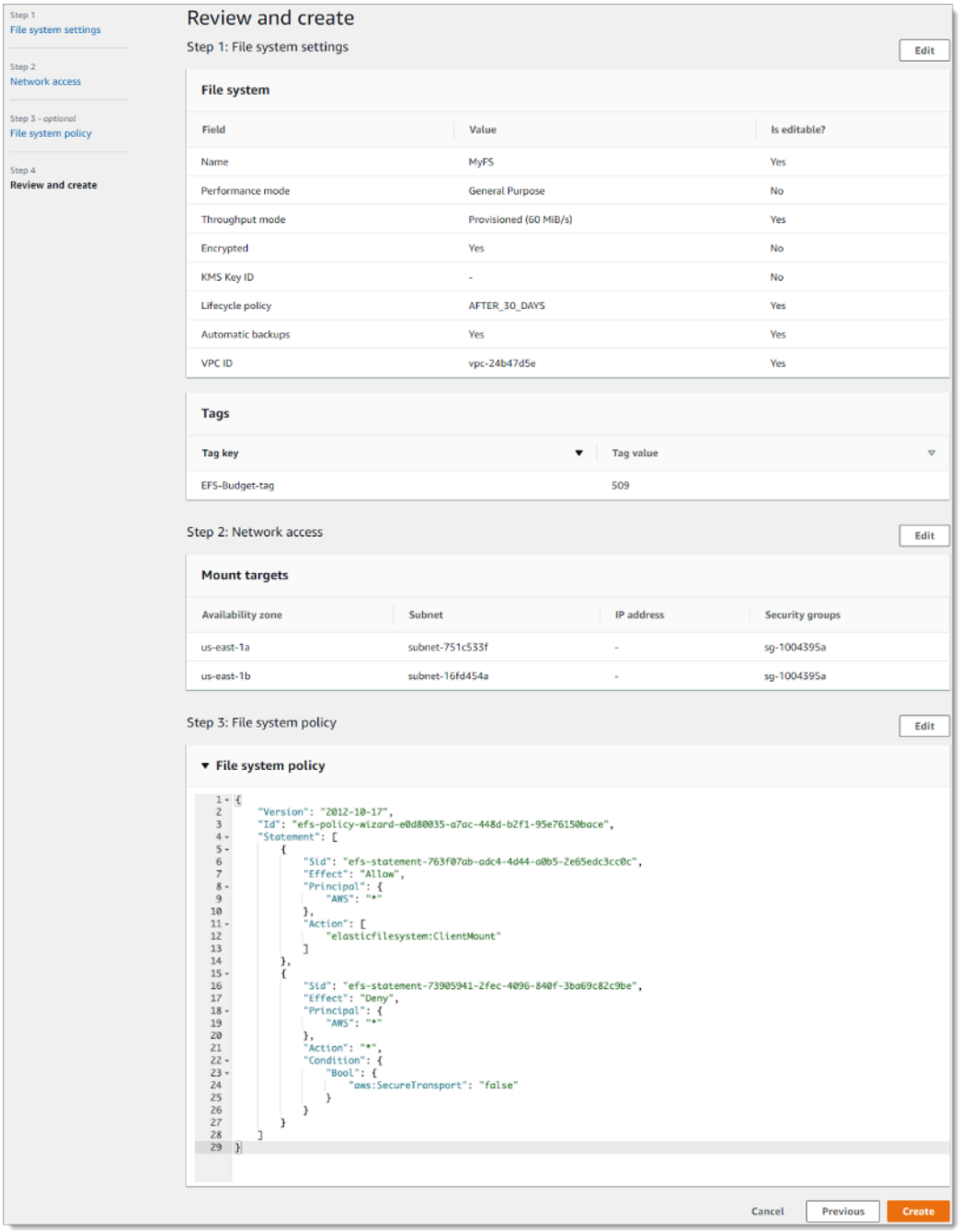


EFS 파일 시스템 생성: 파일 시스템 정책

1. 정책 옵션에서 다음과 같은 사용 가능한 사전 구성된 정책 옵션을 선택하는 것이 좋습니다.
   - 기본적으로 루트 액세스 차단
   - 기본적으로 읽기 전용 액세스 적용
   - 모든 클라이언트에 전송 중 데이터 암호화 적용
2. 추가 권한 부여(Grant additional permissions)를 사용하여 다른 AWS 계정을 비롯한 추가 IAM 보안 주체에게 파일 시스템 권한을 부여합니다. 추가(Add)를 선택한 다음, 권한을 부여할 엔터티의 보안 주체 ARN을 입력하고 부여할 권한(Permissions)을 선택합니다.
3. 정책 편집기(Policy editor)를 사용하여 미리 구성된 정책을 사용자 지정하거나 요구 사항에 따라 고유한 정책을 만듭니다. 미리 구성된 정책 중 하나를 선택하면 정책 편집기에 JSON 정책 정의가 표시됩니다.
4. 다음(Next)을 선택하여 계속 진행합니다. 검토 및 생성(Review and create) 페이지가 표시됩니다.

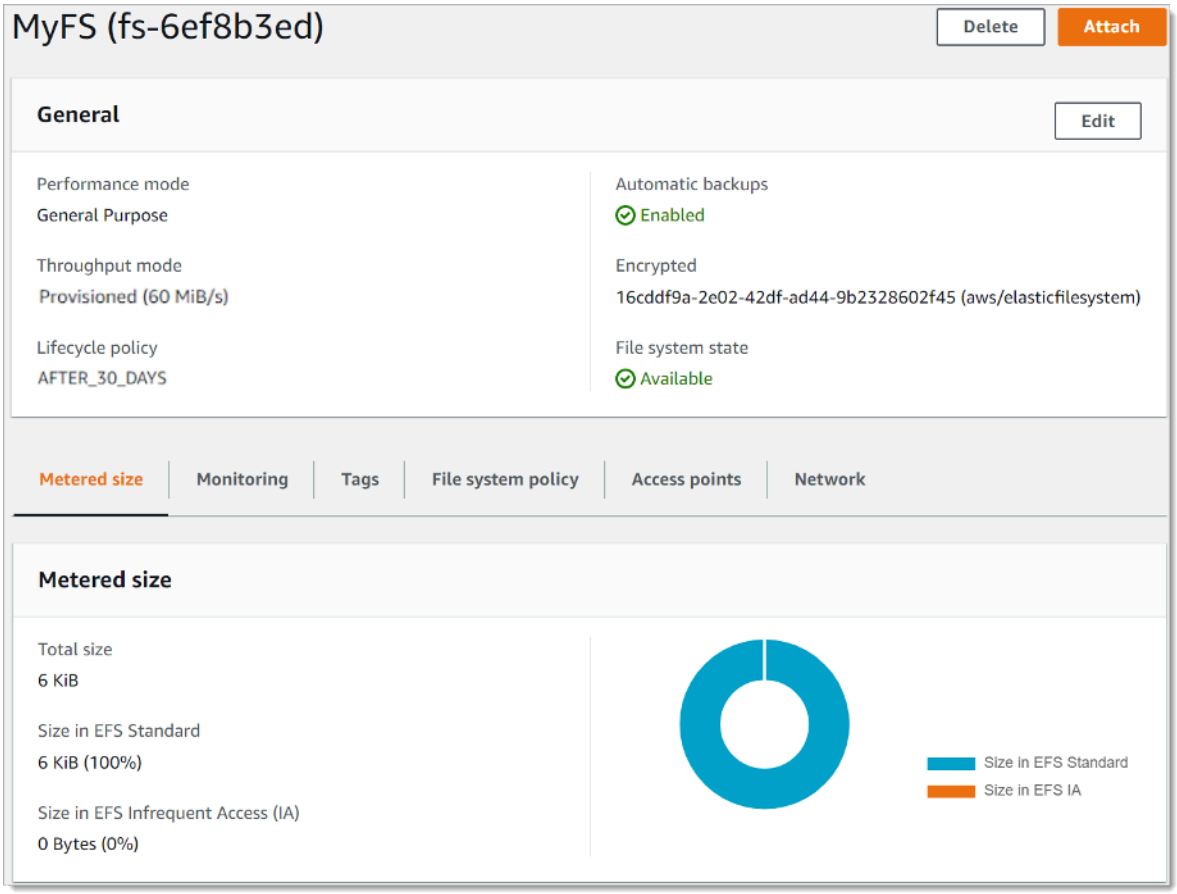
4단계. 검토 및 생성

이 단계에서 파일 시스템 설정을 검토하고 수정한 다음 파일 시스템을 만듭니다.



EFS 파일 시스템 생성: 검토 및 생성

1. 각 파일 시스템의 구성 그룹을 검토합니다. 편집(Edit)을 선택하여 각 그룹을 변경할 수 있습니다.
2. 생성(Create)을 선택하여 파일 시스템을 만들고 파일 시스템 페이지로 돌아갑니다.
3. 파일 시스템(File systems) 페이지에는 다음 이미지와 같이 파일 시스템 및 해당 구성 세부 정보가 표시됩니다.



파일 시스템

AWS CLI를 사용하여 암호화된 파일 시스템 생성

AWS CLI를 사용하여 암호화된 파일 시스템을 생성할 때 추가 파라미터를 사용하여 암호화 상태와 고객 관리형 CMK를 설정할 수 있습니다. 최신 버전의 AWS CLI를 사용하고 있는지 확인합니다. AWS CLI를 업그레이드하는 방법에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 AWS CLI 설치, 업데이트 및 제거를 참조하세요.

`CreateFileSystem` 작업에서 `--encrypted` 파라미터는 부울이며 암호화된 파일 시스템을 만드는 데 필요합니다. `--kms-key-id`는 고객 관리형 CMK를 사용하고 키의 별칭 또는 ARN을 포함하는 경우에만 필요합니다. AWS 관리형 CMK를 사용하는 경우 이 파라미터를 포함하지 마세요.

```
$ aws efs create-file-system \
--creation-token $(uuidgen) \
--performance-mode generalPurpose \
--encrypted \
```

```
        --kms-key-id user/customer-managedCMKalias
```

AWS 관리 콘솔, AWS CLI, AWS SDK 또는 Amazon EFS API를 사용하여 Amazon EFS 파일 시스템을 생성하는 방법에 대한 자세한 내용은 Amazon EFS 사용 설명서의 Amazon Elastic File System이란 무엇입니까?를 참조하세요.

저장된 데이터의 암호화 비활성화

암호화는 I/O 대기 시간 및 처리량에 최소한의 영향을 미칩니다. 암호화 및 암호 해독은 사용자, 애플리케이션 및 서비스에 아무런 영향을 미치지 않습니다. 모든 데이터와 메타데이터는 디스크에 기록되기 전에 사용자를 대신하여 Amazon EFS에서 암호화되고 클라이언트가 읽기 전에 암호 해독됩니다. 암호화된 파일 시스템에 액세스하기 위해 클라이언트 도구, 애플리케이션 또는 서비스를 변경할 필요가 없습니다.

조직에서 특정 분류를 충족하거나 특정 애플리케이션이나 워크로드, 환경과 연결된 모든 데이터를 암호화해야 할 수 있습니다. AWS Identity and Access Management(IAM) 자격 증명 기반 정책을 사용하여 Amazon EFS 파일 시스템 리소스에 대해 저장된 데이터의 암호화를 적용할 수 있습니다. IAM 조건 키로 사용자가 암호화되지 않은 EFS 파일 시스템을 생성하지 못하도록 할 수 있습니다.

예를 들어 사용자가 암호화된 EFS 파일 시스템만 만들 수 있도록 명시적으로 허용하는 IAM 정책은 다음과 같은 효과, 작업 및 조건을 조합하여 사용합니다.

- `Effect`는 `Allow`입니다.
- `Action`은 `elasticfilesystem:CreateFileSystem`입니다.
- `Condition` `elasticfilesystem:Encrypted`는 `true`입니다.

다음 예제에서는 보안 주체에서 암호화된 파일 시스템만 생성하도록 권한을 부여하는 IAM 자격 증명 기반 정책을 보여 줍니다.

```
{
  “Version”: “2012-10-17”,
  “Statement”: [
    {
      “Sid”: “VisualEditior0”,
      “Effect”: “Allow”,
      “Action”: “elasticfilesystem:CreateFileSystem”,
      “Condition”: {
```

```
        "Bool": {
          "elasticfilesystem:Encrypted": "true"
        }
      },
      "Resource": "*"
    }
}
```

*로 설정된 Resource 속성은 IAM 정책이 생성된 모든 EFS 리소스에 적용됨을 의미합니다. 태그를 기반으로 조건부 속성을 추가하여 데이터 분류가 필요한 EFS 리소스의 하위 집합에만 적용할 수 있습니다.

또한 조직의 모든 AWS 계정 또는 OU에 대한 서비스 제어 정책을 사용하여 AWS Organizations 수준에서 암호화된 Amazon EFS 파일 시스템을 생성할 수 있습니다. AWS Organizations의 서비스 제어 정책에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.

모든 EFS 파일 시스템을 암호화해야 하는 IAM 정책 생성

콘솔, AWS CLI 또는 API를 사용하여 암호화된 Amazon EFS 파일 시스템만 생성하도록 사용자에게 권한을 부여하는 IAM 자격 증명 기반 정책을 생성할 수 있습니다. 다음 절차에서는 IAM 콘솔을 사용하여 이러한 정책을 생성한 다음, 계정 사용자에게 정책을 적용하는 방법을 설명합니다.

암호화된 EFS 파일 시스템을 적용하기 위한 IAM 정책을 생성하는 방법은 다음과 같습니다.

1. AWS 관리 콘솔에 로그인하고 IAM 콘솔을 엽니다.
2. 탐색 창의 액세스 관리(Access Management)에서 정책(Policies)을 선택합니다.
3. 정책 생성(Create policy)을 선택하면 정책 생성(Create policy) 페이지가 표시됩니다.
4. 시각적 편집기(Visual Editor) 탭에서 다음 정보를 입력합니다.
   - 서비스(Service)에서 EFS를 선택합니다.
   - 작업(Actions)의 경우, 검색 필드에 `create`를 입력한 다음 CreateFileSystem을 선택합니다.
   - 요청 조건(Request conditions)의 경우, 조건 추가(Add condition) 링크를 클릭하고 조건 키(Condition Key)에 대해 `elasticfilesystem:Encrypted`, 연산자(Operator)에 대해 `Bool`, 값(Value)에 대해 `true`를 검색합니다.
5. 정책의 이름(Name) 및 설명(Description)을 입력합니다. 암호화(Encrypted) 요구 조건을 포함하여 정책 요약을 확인합니다.
6. 정책 생성(Create policy)을 선택하여 정책을 생성합니다.

계정 사용자에게 정책을 적용하는 방법:

1. IAM 콘솔의 액세스 관리(Access management)에서 사용자(Users)를 선택합니다.
2. 정책을 적용할 사용자를 선택합니다.
3. 권한 추가(Add permissions)를 선택하여 권한 추가(Add permissions) 페이지를 표시합니다.
4. 기존 정책 직접 연결(Attach existing policies directly)을 선택합니다.
5. 이전 절차에서 생성한 EFS 정책의 이름을 입력합니다.
6. 정책을 선택하고 확장합니다. 그런 다음 {}JSON을 선택하여 정책 내용을 확인합니다. 다음 JSON 정책과 같아야 합니다.

```
{
  “Version”: “2012-10-17”,
  “Statement”: [
    {
      “Sid”: “VisualEditior0”,
      “Effect”: “Allow”,
      “Action”: “elasticfilesystem:CreateFileSystem”,
      “Condition”: {
        “Bool”: {
          “elasticfilesystem:Encrypted”: “true”
        }
      },
      “Resource”: “*”
    }
}
```

암호화되지 않은 파일 시스템 감지

조직에서 암호화되지 않은 Amazon EFS 리소스를 식별해야 할 수도 있습니다. AWS Config 관리형 규칙을 사용하여 암호화되지 않은 파일 시스템을 감지할 수 있습니다. AWS Config는 AWS 관리형 규칙을 제공하며, 이는 AWS Config에서 AWS 리소스가 일반적인 모범 사례를 준수하는지 평가하고 규칙에 실패한 리소스를 NON_COMPLIANT로 표시하는 데 사용하는 사용자 지정 가능한 사전 정의된 규칙입니다.

AWS 관리형 Config 규칙 `efs-encrypted-check`를 사용하여 Amazon Elastic File System(Amazon EFS)이 AWS Key Management Service(AWS KMS)를 사용하여 파일 데이터를 암호화하도록 구성되어 있는지 확인할 수 있습니다. AWS 관리형 규칙 설정 및 활성화에 대한 자세한 내용은 AWS Config 관리형 규칙 작업을 참조하세요.

전송 중인 데이터 암호화

모든 NFS 트래픽이 산업 표준 AES-256 암호와 함께 전송 계층 보안(TLS) 1.2를 사용하여 전송 중에 암호화되도록 파일 시스템을 탑재할 수 있습니다. TLS는 네트워크를 통해 교환되는 정보를 암호화하는 데 사용되는 업계 표준 암호화 프로토콜 집합입니다. AES-256은 TLS에서 데이터를 전송하는 데 사용되는 256비트 암호화 암호입니다. 파일 시스템에 액세스하는 모든 클라이언트에서 전송 중에 암호화를 설정하는 것이 좋습니다.

IAM 정책을 사용하여 Amazon EFS에 대한 NFS 클라이언트 액세스에 전송 중 암호화를 적용할 수 있습니다. 클라이언트가 파일 시스템에 연결하면, Amazon EFS에서 파일 시스템의 IAM 리소스 정책(즉, 파일 시스템 정책)과 자격 증명 기반의 IAM 정책을 평가하여 부여할 적절한 파일 시스템 액세스 권한을 결정합니다. 파일 시스템 리소스 정책의 `aws:SecureTransport` 조건 키를 사용하여 EFS 파일 시스템에 연결할 때 NFS 클라이언트가 TLS를 사용하도록 적용할 수 있습니다.

> **Note**
>
> IAM 권한 부여를 사용하여 NFS 클라이언트의 액세스를 제어하려면 EFS 탑재 헬퍼를 사용하여 Amazon EFS 파일 시스템을 탑재해야 합니다. 자세한 내용은 Amazon EFS 사용 설명서의 IAM 권한 부여를 사용하여 탑재를 참조하세요.

다음 예제인 EFS 파일 시스템 정책은 전송 중에 암호화를 적용하며 다음과 같은 특징이 있습니다.

- `effect`는 `allow`입니다.
- 보안 주체는 모든 IAM 엔터티에 대해 *로 설정됩니다.
- 작업은 `ClientMount`, `ClientWrite` 및 `ClientRootAccess`로 설정됩니다.
- 권한 부여 조건은 `SecureTransport`로 설정됩니다. TLS를 사용하여 파일 시스템에 연결하는 NFS 클라이언트에만 액세스 권한이 부여됩니다.

```
{
  “Version”: “2012-10-17”,
  “Id”: “ExamplePolicy01”,
  “Statement”: [
    {
      “Sid”: “VisualEditior0”,
      “Effect”: “Allow”,
      “Principal”: {
```

```
        "AWS": "*",
      },
      "Action": [
"elasticfilesystem:ClientRootAccess",
"elasticfilesystem:ClientMount",
"elasticfilesystem:ClientWrite"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

Amazon EFS 콘솔 또는 AWS CLI를 사용하여 파일 시스템 정책을 생성할 수 있습니다.

EFS 콘솔을 사용하여 파일 시스템 정책을 생성하는 방법은 다음과 같습니다.

1. Amazon EFS 콘솔을 엽니다.
2. 파일 시스템(File Systems)을 선택합니다.
3. 파일 시스템(File systems) 페이지에서 파일 시스템 정책을 편집 또는 생성할 파일 시스템을 선택합니다. 해당 파일 시스템의 세부 정보 페이지가 표시됩니다.
4. 파일 시스템 정책(File system policy)을 선택한 다음, 편집(Edit)을 선택합니다. 파일 시스템 정책(File system policy) 페이지가 표시됩니다.

파일 시스템 정책 생성

5. 정책 옵션(Policy options)에서 다음과 같이 사용 가능한 사전 구성된 정책 옵션을 선택하는 것이 좋습니다.

   - 기본적으로 루트 액세스 차단
   - 기본적으로 읽기 전용 액세스 적용
   - 모든 클라이언트에 전송 중 데이터 암호화 적용

   미리 구성된 정책을 선택하면 정책 JSON 개체가 정책 편집기(Policy editor) 패널에 표시됩니다.

6. 추가 권한 부여(Grant additional permissions)를 사용하여 다른 AWS 계정을 비롯한 추가 IAM 보안 주체에게 파일 시스템 권한을 부여합니다. 추가(Add)를 선택한 다음, 권한을 부여할 엔터티의 보안 주체 ARN을 입력하고 부여할 권한(Permissions)을 선택합니다.
7. 정책 편집기(Policy editor)를 사용하여 미리 구성된 정책을 사용자 지정하거나 요구 사항에 따라 고유한 정책을 만듭니다. 이 편집기를 사용하면 미리 구성된 정책 옵션을 사용할 수 없게 됩니다. 정책 변경 사항을 실행 취소하려면 지우기(Clear)를 선택합니다.

편집기를 지우면 미리 구성된 정책을 다시 사용할 수 있게 됩니다.

8. 정책 편집 또는 작성을 완료한 후 저장(Save)을 선택합니다.

   파일 시스템의 세부 정보 페이지가 표시되고 파일 시스템 정책(File system policy)에 정책이 표시됩니다.

   AWS CloudFormation, AWS SDK 또는 Amazon EFS API를 직접 사용하여 프로그래밍 방식으로 파일 시스템 정책을 생성할 수도 있습니다. 파일 시스템 정책 생성에 대한 자세한 내용은 Amazon EFS 사용 설명서의 파일 시스템 정책 생성을 참조하세요.

전송 중인 데이터의 암호화 설정

전송 중인 데이터의 암호화를 설정하려면 각 클라이언트에서 EFS 탑재 헬퍼를 다운로드하는 것이 좋습니다. EFS 탑재 헬퍼는 전송 중인 데이터의 암호화 설정을 비롯하여 EFS 사용을 간소화하기 위해 AWS에서 제공하는 오픈 소스 유틸리티입니다. 탑재 헬퍼는 기본적으로 EFS 권장 탑재 옵션을 사용합니다.

EFS 탑재 헬퍼는 다음과 같은 Linux 배포판에서 지원됩니다.

- Amazon Linux 2017.09+
- Amazon Linux 2+
- Debian 9+
- Fedora 28+
- Red Hat Enterprise Linux / CentOS 7+
- Ubuntu 16.04+

전송 중인 데이터의 암호화를 설정하는 방법은 다음과 같습니다.

1. EFS 탑재 헬퍼를 설치합니다.
   - Amazon Linux의 경우 다음 명령을 사용합니다.

     ```
     sudo yum install -y amazon-efs-utils
     ```

   - 다른 Linux 배포판의 경우 GitHub에서 다운로드하여 설치합니다.

Amazon-efs-utils 패키지는 NFS 클라이언트(nfs-utils), 네트워크 릴레이(stunnel), OpenSSL 및 Python과 같은 종속성을 자동으로 설치합니다.

2. 파일 시스템을 탑재합니다.

```
sudo mount -t efs -o tls file-system-id
efs-mount-point
```

- `mount -t efs`는 EFS 탑재 헬퍼를 호출합니다.
- EFS 탑재 헬퍼를 사용하여 탑재하는 경우 파일 시스템의 DNS 이름 또는 탑재 대상의 IP 주소를 사용할 수 없습니다. 대신 파일 시스템 ID를 사용하세요.
- EFS 탑재 헬퍼는 기본적으로 AWS 권장 탑재 옵션을 사용합니다. 이러한 기본 탑재 옵션을 재정의하는 것은 권장되지 않지만 상황에 따라 유연하게 변경할 수 있습니다. 탑재 옵션 재정의를 철저히 테스트하여 이러한 변경 사항이 파일 시스템 액세스 및 성능에 어떤 영향을 미치는지 파악하는 것이 좋습니다.
- 다음 표에는 EFS 탑재 헬퍼에서 사용하는 기본 탑재 옵션이 나와 있습니다.

옵션	설명
nfsvers=4.1	NFS 프로토콜 버전
rsize=1048576	NFS 클라이언트가 각 네트워크 읽기 요청에 대해 수신할 수 있는 최대 데이터 바이트 수
wsize=1048576	NFS 클라이언트가 각 네트워크 쓰기 요청에 대해 전송할 수 있는 최대 데이터 바이트 수

옵션	설명
hard	NFS 요청 시간이 초과된 후에는 NFS 클라이언트의 복구 동작을 설정하여 서버가 응답할 때까지 NFS 요청을 무기한 재시도
timeo=600	NFS 클라이언트가 NFS 요청을 재시도하기 전에 응답을 기다리는 데 사용하는 제한 시간 값(데시초)
retrans=2	NFS 클라이언트가 추가 복구 작업을 시도하기 전에 요청을 재시도하는 횟수
noresvport	네트워크 연결이 다시 설정되면 NFS 클라이언트에 새 TCP 소스 포트를 사용하도록 지시

- 시스템을 다시 시작한 후 파일 시스템을 자동으로 다시 탑재하려면 `/etc/fstab`에 다음 줄을 추가합니다.

```
file-system-id efs-mount-point efs _netdev, tls, iam 0 0
```

전송 중인 데이터 암호화 사용

조직에서 전송 중인 데이터 암호화를 요구하는 기업 정책 또는 규제 정책이 적용되는 경우, 파일 시스템에 액세스하는 모든 클라이언트에서 전송 중인 데이터의 암호화를 사용하는 것이 좋습니다. 암호화 및 암호 해독은 연결 수준에서 구성되며 또 다른 보안 계층을 추가합니다.

EFS 탑재 헬퍼를 사용하여 파일 시스템을 탑재하면 클라이언트와 Amazon EFS 간에 TLS 1.2 터널이 설정 및 유지되며, 모든 NFS 트래픽이 이 암호화된 터널을 통해 라우팅됩니다. 암호화된 TLS 연결을 설정하는 데 사용되는 인증서는 Amazon 인증 기관(CA)에서 서명하고 대부분의 최신 Linux 배포판에서 신뢰합니다. 또한 EFS 탑재 헬퍼는 감시 프로세스를 생성하여 각 파일 시스템에 대한 모든 보안 터널을 모니터링하고 실행 중인지 확인합니다.

EFS 탑재 헬퍼를 사용하여 Amazon EFS에 대한 암호화된 연결을 설정한 후에는 다른 사용자 입력이나 구성이 필요하지 않습니다. 암호화는 파일 시스템에 액세스하는 사용자 연결 및 애플리케이션에 영향을 미치지 않습니다.

EFS 탑재 헬퍼를 사용하여 EFS 파일 시스템에 암호화된 연결을 탑재하고 설정하면 탑재 명령의 출력에 파일 시스템이 탑재되고 localhost(127.0.0.1)를 네트워크 릴레이로 사용하여 암호화된 터널이 설정되었음을 알 수 있습니다. 다음 샘플 출력을 참조하세요.

```
127.0.0.1:/ on efs-mount-point type nfs4

(rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,port=20059,timeo=6
```

`efs-mount-point`를 EFS 파일 시스템에 매핑하려면 `/var/log/amazon/efs`에서 mount.log 파일을 쿼리하고 마지막으로 성공한 탑재 작업을 찾습니다. 이 작업은 다음과 같은 간단한 grep 명령을 사용하여 수행할 수 있습니다.

```
grep -E "Successfully
mounted.*efs-mount-point"
/var/log/amazon/efs/mount.log | tail -1
```

이 grep 명령의 출력은 탑재된 EFS 파일 시스템의 DNS 이름을 반환합니다. 아래 샘플 출력을 참조하세요.

```
2018-03-15 07:03:42,363 - INFO - Successfully mounted
file-system-id.efs.region.amazonaws.com
at efs-mount-point
```

결론

Amazon EFS 파일 시스템 데이터는 저장 및 전송 중에 암호화할 수 있습니다. AWS KMS를 사용하여 제어 및 관리할 수 있는 CMK로 저장된 데이터를 암호화할 수 있습니다. 암호화된 파일 시스템을 생성하는 일은 AWS 관리 콘솔의 Amazon EFS 파일 시스템 생성 마법사에서 확인란을 선택하거나 AWS CLI, AWS SDK 또는 Amazon EFS API의 `CreateFileSystem` 작업에 단일 파라미터를 추가하는 것만 큼 간단합니다.

AWS IAM 자격 증명 기반의 정책 및 파일 시스템의 정책으로 저장 및 전송 시 암호화를 적용하여 보안 요구 사항을 더욱 강화하고 규정 준수 요구 사항을 충족할 수 있습니다. 암호화된 파일 시스템을 사용하면 서비스, 애플리케이션 및 사용자에게도 영향을 미치지 않으면서 파일 시스템 성능에 미치는 영향은 최소화됩니다. EFS 탑재 헬퍼를 사용하여 각 클라이언트에 암호화된 TLS 터널을 설정하고 클라이언트와 탑재된 EFS 파일 시스템 간의 모든 NFS 트래픽을 암호화하여 전송 중인 데이터를 암호화할 수 있습니다. IAM 자격 증명 정책을 사용하거나 EFS 파일 시스템 정책을 사용하여 전송 중인 Amazon EFS 저장된 데이터의 암호화를 추가 비용 없이 적용할 수 있습니다.

리소스

- AWS KMS 암호화 세부 정보 백서
- Amazon EFS 사용 설명서

문서 기록 및 기여자

문서 기록

이 백서의 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하세요.

업데이트 기록-변경	update-history-description	update-history-date
마이너 업데이트	조정된 페이지 레이아웃	2021년 4월 30일
백서 업데이트됨	IAM을 사용하여 저장 및 전송 중인 암호화 적용 추가	2021년 2월 22일
백서 업데이트됨	전송 중인 데이터 암호화 추가	2018년 4월 1일
첫 게시	게시된 Amazon EFS 암호화 파일 시스템으로 저장된 데이터 암호화	2017년 9월 1일

> **Note**
>
> RSS 업데이트에 가입하려면 사용 중인 브라우저에 대해 RSS 플러그인이 활성화되어 있어야 합니다.

기여자

이 문서를 작성하는 데 도움을 주신 분들입니다.

- Darryl S. Osborne, AWS 스토리지 전문 솔루션스 아키텍트
- Joseph Travaglini, Amazon EFS 선임 제품 관리자
- Peter Buonora, AWS 수석 솔루션스 아키텍트
- Siva Rajamani, AWS 선임 솔루션스 아키텍트