사용자 가이드

AWS 확인된 액세스

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 확인된 액세스: 사용자 가이드

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS Verified Access란 무엇인가요? 1
Verified·Access의 이점 1
Verified·Access 액세스 1
요금 2
Verified·Access의 작동 방식 3
Verified·Access의 주요 구성 요소 3
시작하기 자습서 5
사전 조건 5
신뢰 공급자 생성 6
인스턴스 생성 6
그룹 생성 7
엔드포인트 생성 7
엔드포인트에 대한 DNS 구성 8
애플리케이션에 대한 연결 테스트 9
액세스 정책 추가 9
정리 9
Verified·Access 인스턴스 11
Verified Access 인스턴스 생성 및 관리 11
Verified·Access 인스턴스 생성 11
Verified Access 인스턴스에 신뢰 공급자 연결 12
Verified Access 인스턴스에서 신뢰 공급자 분리 12
사용자 지정 하위 도메인 추가 13
Verified·Access 인스턴스 삭제 13
와 통합 AWS WAF 14
필수 IAM 권한 15
AWS WAF 웹 ACL 연결 15
연결의 상태 확인 16
AWS WAF 웹 ACL 연결 해제 16
FIPS 규정 준수 17
기존 환경 17
새 환경 18
신뢰 공급자 19
사용자 자격 증명 19
IAM Identity Center 19

OIDC 신뢰 공급자 21
디바이스 기반 24
지원되는 디바이스 신뢰 공급자 24
디바이스 기반 신뢰 공급자 생성 25
디바이스 기반 신뢰 공급자 수정 25
디바이스 기반 신뢰 공급자 삭제 26
Verified·Access 그룹 27
Verified Access 그룹 생성 및 관리 27
Verified·Access 그룹 생성 28
Verified Access 그룹 수정 28
Verified·Access 그룹 정책 수정 29
다른 계정과 그룹 공유 29
고려 사항 30
리소스 공유 31
Verified·Access 그룹 삭제 31
Verified·Access 엔드포인트 33
Verified·Access 엔드포인트 유형 33
공유 VPC 및 서브넷에서 Verified Access의 작동 방식 34
로드 밸런서 엔드포인트 생성 34
네트워크 인터페이스 엔드포인트 생성 36
네트워크 CIDR 엔드포인트 생성 37
Amazon Relational Database Service 엔드포인트 생성 38
엔드포인트로부터의 트래픽 허용 40
Verified·Access 엔드포인트 수정 40
Verified·Access 엔드포인트 정책 수정 41
Verified·Access 엔드포인트 삭제 41
Verified Access 신뢰 데이터 43
기본 컨텍스트 43
HTTP 요청 44
TCP 흐름 45
AWS IAM Identity Center 컨텍스트 46
서드 파티 컨텍스트 48
브라우저 확장 48
Jamf 49
CrowdStrike 50
JumpCloud 52

사용자 클레임 통과 ... 54
OIDC 사용자 클레임용 JWT ... 54
IAM Identity Center 사용자 클레임용 JWT ... 55
퍼블릭 키 ... 56
JWT 검색 및 디코딩 ... 56
Verified·Access 정책 ... 58
정책 문 ... 58
정책 구성 요소 ... 59
설명 ... 59
여러 절 ... 59
예약 문자 ... 60
내장 연산자 ... 60
정책 평가 ... 62
정책 논리 단락 ... 62
정책 예제 ... 63
IAM Identity Center에서 그룹에 액세스 권한 부여 ... 64
서드 파티 공급자에서 그룹에 액세스 권한 부여 ... 64
CrowdStrike를 사용하여 액세스 권한 부여 ... 65
특정 IP 주소 허용 또는 거부 ... 65
정책 도우미 ... 65
1단계: 리소스 지정 ... 66
2단계: 정책 테스트 및 편집 ... 66
3단계: 변경 사항 검토 및 적용 ... 67
연결 클라이언트 ... 68
사전 조건 ... 68
연결 클라이언트 다운로드 ... 69
클라이언트 구성 파일 내보내기 ... 69
애플리케이션에 연결 ... 69
클라이언트 제거 ... 70
모범 사례 ... 70
문제 해결 ... 71
로그인할 때 브라우저가 열려 IdP에 의한 인증을 완료하지 않습니다. ... 71
인증 후 클라이언트 상태는 "연결되지 않음"입니다. ... 71
Chrome 또는 Edge 브라우저를 사용하여 연결할 수 없음 ... 72
버전 기록 ... 72
보안 ... 73

데이터 보호 73
전송 중 암호화 74
인터네트워크 트래픽 개인 정보 보호 74
유휴 시(저장된) 데이터 암호화 75
자격 증명 및 액세스 관리 89
대상 89
ID를 통한 인증 90
정책을 사용하여 액세스 관리 93
Verified·Access가 IAM과 함께 작동하는 방식 95
자격 증명 기반 정책 예제 101
문제 해결 105
서비스 연결 역할 사용 106
AWS 관리형 정책 108
규정 준수 확인 110
복원성 111
고가용성을 위한 다중 서브넷 111
모니터링 112
Verified·Access 로그 112
로깅 버전 113
로깅 권한 113
로그 활성화 또는 비활성화 114
신뢰 컨텍스트 활성화 또는 비활성화 116
OCSF 버전 0.1 로그 예제 117
OCSF 버전 1.0.0-rc.2 로그 예제 129
CloudTrail 로그 136
관리 이벤트 138
이벤트 예 138
할당량 140
문서 기록 142
..... cxliv

AWS Verified Access란 무엇인가요?

를 AWS Verified Access사용하면 가상 프라이빗 네트워크(VPN)를 사용하지 않고도 애플리케이션에 대한 보안 액세스를 제공할 수 있습니다. Verified Access는 각 애플리케이션 요청을 평가하여 사용자가 지정된 보안 요구 사항을 충족하는 경우에만 각 애플리케이션에 액세스할 수 있도록 합니다.

Verified·Access의 이점

- 보안 상태 개선 - 기존 보안 모델은 액세스를 한 번 평가하여 사용자에게 모든 애플리케이션에 대한 액세스 권한을 부여합니다. Verified·Access는 각 애플리케이션 액세스 요청을 실시간으로 평가합니다. 이로 인해 악의적인 공격자가 한 애플리케이션에서 다른 애플리케이션으로 이동하기가 어렵습니다.
- 보안 서비스와의 통합 - Verified Access는 및 타사 서비스를 포함한 ID AWS 및 디바이스 관리 서비스와 통합됩니다. Verified Access는 이러한 서비스의 데이터를 사용하여 보안 요구 사항 집합을 기준으로 사용자와 디바이스의 신뢰성을 확인하고 사용자가 애플리케이션에 액세스할 수 있는지 여부를 결정합니다.
- 사용자 환경 개선 - Verified Access를 사용하면 사용자가 VPN을 사용하여 애플리케이션에 액세스할 필요가 없습니다. 이를 통해 VPN 관련 문제로 인해 발생하는 지원 사례 수를 줄일 수 있습니다.
- 간소화된 문제 해결 및 감사 - Verified Access는 모든 액세스 시도를 기록하여 애플리케이션 액세스에 대한 중앙 집중식 가시성을 제공하여 보안 사고 및 감사 요청에 신속하게 대응할 수 있도록 지원합니다.

Verified·Access 액세스

다음 인터페이스 중 하나를 사용하여 Verified·Access로 작업할 수 있습니다.

- AWS Management Console – Verified·Access 리소스를 생성하고 관리하는 데 사용할 수 있는 웹 인터페이스를 제공합니다. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/vpc/ Amazon VPC 콘솔을 엽니다.
- AWS Command Line Interface (AWS CLI) - 다음을 AWS 서비스포함한 광범위한에 대한 명령을 제공합니다 AWS Verified Access. AWS CLI 는 Windows, macOS 및 Linux에서 지원됩니다. 를 가져오려면 섹션을 AWS CLI참조하세요AWS Command Line Interface.
- AWS SDKs- 언어별 APIs 제공합니다. AWS SDK는 서명 계산, 요청 재시도 및 오류 처리와 같은 많은 연결 세부 정보를 관리합니다. 자세한 내용은 AWS SDK를 참조하십시오.

- 쿼리 API - HTTPS 요청을 사용하여 호출하는 하위 수준의 API 작업을 제공합니다. 쿼리 API 사용은 Verified·Access에 액세스할 수 있는 가장 직접적인 방법입니다. 하지만 이를 사용하려면 애플리케이션에서 요청에 서명할 해시 생성 및 오류 처리와 같은 하위 수준의 세부 정보를 처리해야 합니다. 자세한 내용은 Amazon EC2 API 참조의 Verified·Access 작업을 참조하십시오.

이 안내서에서는를 사용하여 Verified Access 리소스를 AWS Management Console 생성, 액세스 및 관리하는 방법을 설명합니다.

요금

Verified Access의 각 애플리케이션에 대해 시간당 요금이 청구되며, Verified Access에서 처리된 데이터 양에 대한 요금이 청구됩니다. 자세한 내용은 AWS Verified Access 요금을 참조하세요.

Verified·Access의 작동 방식

AWS Verified Access 는 사용자의 각 애플리케이션 요청을 평가하고 다음을 기반으로 액세스를 허용합니다.

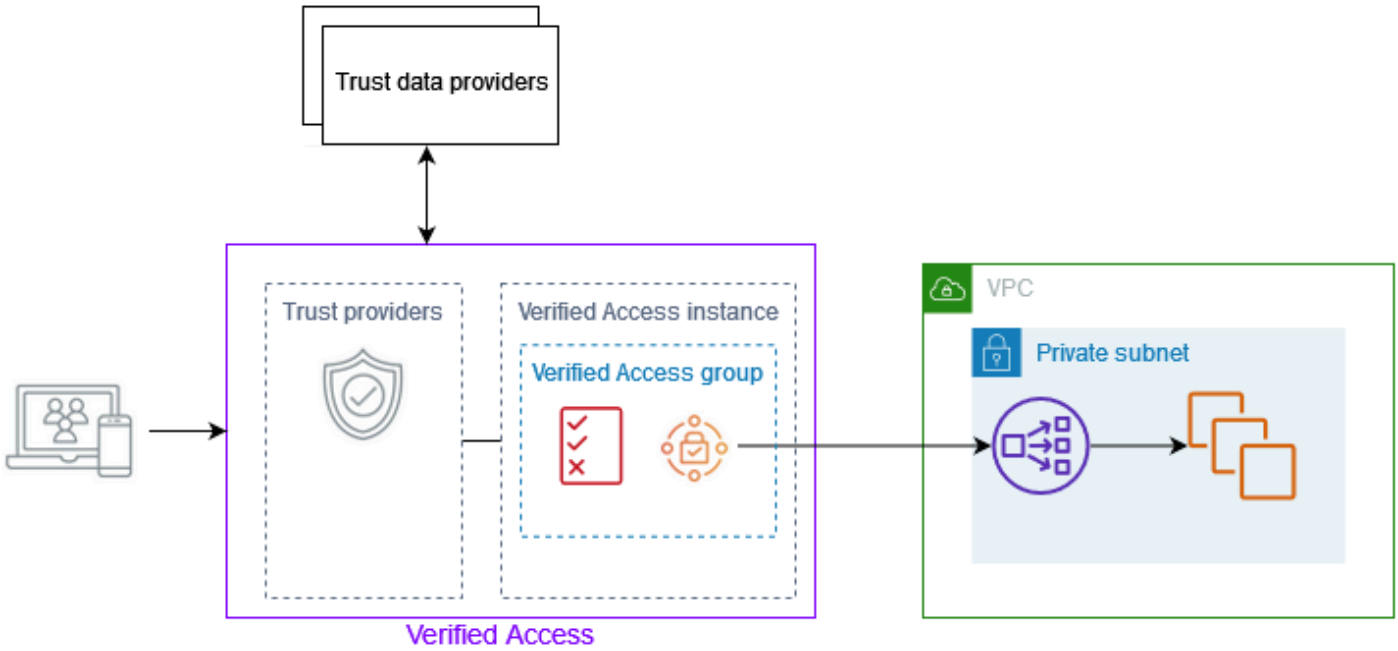
- 선택한 신뢰 공급자(AWS 또는 타사)가 전송한 신뢰 데이터입니다.
- Verified·Access에서 생성한 액세스 정책.

사용자가 애플리케이션에 액세스하려고 하면 Verified·Access는 신뢰 공급자로부터 데이터를 가져와 애플리케이션에 대해 설정한 정책과 비교하여 평가합니다. Verified·Access는 사용자가 지정된 보안 요구 사항을 충족하는 경우에만 요청된 애플리케이션에 대한 액세스 권한을 부여합니다. 정책이 정의될 때까지 모든 애플리케이션 요청은 기본적으로 거부됩니다.

또한 Verified·Access는 모든 액세스 시도를 기록하므로 보안 사고 및 감사 요청에 신속하게 대응할 수 있습니다.

Verified·Access의 주요 구성 요소

다음은 Verified·Access의 중요한 개요를 설명하는 다이어그램입니다. 사용자가 애플리케이션 액세스 요청을 보냅니다. Verified·Access는 그룹에 대한 액세스 정책 및 애플리케이션별 엔드포인트 정책을 기준으로 요청을 평가합니다. 액세스가 허용되면 해당 요청이 엔드포인트를 통해 애플리케이션에 전송됩니다.



- Verified·Access 인스턴스 – 인스턴스는 애플리케이션 요청을 평가하여 보안 요구 사항이 충족되는 경우에만 액세스를 부여합니다.
- Verified·Access 엔드포인트 - 각 엔드포인트는 애플리케이션을 나타냅니다. 위의 다이어그램에서 애플리케이션은 로드 밸런서의 대상인 EC2 인스턴스에서 호스팅됩니다.
- Verified·Access 그룹 – Verified·Access 엔드포인트의 모음입니다. 정책 관리를 단순화하려면 보안 요구 사항이 비슷한 애플리케이션의 엔드포인트를 그룹화하는 것이 좋습니다. 예를 들어, 모든 판매 애플리케이션의 엔드포인트를 함께 그룹화할 수 있습니다.
- 액세스 정책 - 애플리케이션에 대한 액세스를 허용할지 거부할지를 결정하는 사용자 정의 규칙 집합입니다. 사용자 ID 및 디바이스 보안 상태를 비롯한 여러 요소를 조합하여 지정할 수 있습니다. 각 Verified·Access 그룹에 대해 그룹 액세스 정책을 생성합니다. 이 정책은 그룹의 모든 엔드포인트에 상속됩니다. 선택적으로 애플리케이션별 정책을 생성하여 특정 엔드포인트에 연결할 수 있습니다.
- 신뢰 공급자 - 사용자 ID 또는 디바이스 보안 상태를 관리하는 서비스입니다. Verified Access는 AWS 및 타사 신뢰 공급자와 함께 작동합니다. 각 Verified·Access 인스턴스에 하나 이상의 신뢰 공급자를 연결해야 합니다. 각 Verified·Access 인스턴스에 단일 ID 신뢰 공급자와 여러 디바이스 신뢰 공급자를 연결할 수 있습니다.
- 신뢰 데이터 – 신뢰 공급자가 Verified·Access에 보내는 사용자 또는 디바이스의 보안 관련 데이터입니다. 사용자 클레임또는 신뢰 컨텍스트라고도 합니다. 사용자의 이메일 주소 또는 디바이스의 운영 체제 버전을 예로 들 수 있습니다. Verified·Access는 각 애플리케이션 액세스 요청을 받을 때 액세스 정책을 기준으로 이 데이터를 평가합니다.

자습서: Verified Access 시작하기

이 자습서를 사용하여를 시작합니다 AWS Verified Access. Verified·Access 리소스를 생성하고 구성하는 방법을 알아봅니다.

이 자습서의 일부로 Verified Access에 애플리케이션을 추가합니다. 이 자습서를 마치면 특정 사용자가 VPN을 사용하지 않고도 인터넷을 통해 해당 애플리케이션에 액세스할 수 있습니다. 대신를 자격 증명 신뢰 공급자 AWS IAM Identity Center 로 사용합니다. 이 자습서에서는 디바이스 신뢰 공급자를 사용하지 않습니다.

업무

- Verified Access 자습서 사전 조건
- 1단계: Verified Access 신뢰 공급자 생성
- 2단계: Verified·Access 인스턴스 생성
- 3단계: Verified Access 그룹 생성
- 4단계: Verified Access 엔드포인트 생성
- 5단계: Verified Access 엔드포인트에 대한 DNS 구성
- 6단계: 애플리케이션에 대한 연결 테스트
- 7단계: Verified Access 그룹 수준 액세스 정책 추가
- Verified Access 리소스 정리

Verified Access 자습서 사전 조건

이 자습서를 완료하기 위한 사전 조건은 다음과 같습니다.

- AWS IAM Identity Center 작업 AWS 리전 중인에서 활성화됩니다. 그러면 Verified·Access를 통한 신뢰 공급자로 IAM Identity Center를 사용할 수 있습니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 활성화 AWS IAM Identity Center를 참조하세요.
- 애플리케이션에 대한 액세스를 제어하는 보안 그룹. VPC CIDR의 모든 인바운드 트래픽과 모든 아웃바운드 트래픽을 허용합니다.
- Elastic Load Balancing의 내부 로드 밸런서 뒤에서 실행되는 애플리케이션. 로드 밸런서와 해당 보안 그룹을 연결합니다.
- 자체 서명된 또는 퍼블릭 TLS 인증서입니다 AWS Certificate Manager. 키 길이가 1,024 또는 2,048인 RSA 인증서를 사용합니다.

- 퍼블릭 호스팅 도메인과 해당 도메인의 DNS 레코드를 업데이트하는 데 필요한 권한.
- AWS Verified Access 인스턴스를 생성하는 데 필요한 권한이 있는 IAM 정책입니다. 자세한 내용은 Verified·Access 인스턴스 생성 정책 단원을 참조하십시오.

1단계: Verified Access 신뢰 공급자 생성

다음 절차에 따라를 신뢰 공급자 AWS IAM Identity Center 로 설정합니다.

IAM Identity Center 신뢰 공급자를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택합니다.
3. Verified·Access 신뢰 공급자 생성을 선택합니다.
4. (선택 사항) 이름 태그 및 설명에 Verified·Access 신뢰 공급자의 이름과 설명을 입력합니다.
5. 나중에 정책 참조 이름에 대한 정책 규칙 작업 시 사용할 사용자 지정 식별자를 입력합니다. 예를 들면 **`idc`**를 입력할 수 있습니다.
6. 신뢰 공급자 유형에서 사용자 신뢰 공급자를 선택합니다.
7. 사용자 신뢰 공급자 유형에서 IAM Identity Center를 선택합니다.
8. Verified·Access 신뢰 공급자 생성을 선택합니다.

2단계: Verified·Access 인스턴스 생성

다음 절차에 따라 Verified·Access 인스턴스를 생성하십시오.

Verified·Access 인스턴스를 생성하려면

1. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
2. Verified·Access 인스턴스 생성을 선택합니다.
3. (선택 사항) 이름 및 설명에 Verified·Access 인스턴스의 이름과 설명을 입력합니다.
4. Verified·Access 신뢰 공급자에서 신뢰 공급자를 선택합니다.
5. Verified·Access 인스턴스 생성을 선택합니다.

3단계: Verified Access 그룹 생성

Verified·Access 그룹을 생성하려면 다음 절차를 따르십시오.

Verified·Access 그룹 생성

1. 탐색 창에서 Verified·Access 그룹을 선택합니다.
2. Verified·Access 그룹 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 그룹의 이름과 설명을 입력합니다.
4. Verified·Access 인스턴스에서 Verified·Access 인스턴스를 선택합니다.
5. 정책 정의를 비워 둡니다. 이후 단계에서 그룹 수준 정책을 추가합니다.
6. Verified·Access 그룹 생성을 선택합니다.

4단계: Verified Access 엔드포인트 생성

다음 절차에 따라 Verified Access 엔드포인트를 생성하세요. 이 단계에서는 Elastic Load Balancing의 내부 로드 밸런서 뒤에서 실행되는 애플리케이션과 AWS Certificate Manager의 퍼블릭 도메인이 있다고 가정합니다.

Verified·Access 엔드포인트 생성

1. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
2. Verified·Access 엔드포인트 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
4. Verified·Access 그룹에서 Verified·Access 그룹을 선택합니다.
5. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 프로토콜에서 로드 밸런서의 구성에 따라 HTTPS 또는 HTTP를 선택합니다.
   b. 연결 유형(Attachment type)에서 VPC를 선택합니다.
   c. 엔드포인트 유형에서 로드 밸런서를 선택합니다.
   d. 포트에 로드 밸런서 리스너에서 사용하는 포트 번호를 입력합니다. 예를 들어 HTTPS의 경우 443, HTTP의 경우 80입니다.
   e. 로드 밸런서 ARN에서 로드 밸런서를 선택합니다.
   f. 서브넷에서 로드 밸런서와 연결된 서브넷을 선택합니다.

g. 보안 그룹에서 해당 보안 그룹을 선택합니다. 로드 밸런서와 엔드포인트에 동일한 보안 그룹을 사용하는 경우 이들 간의 트래픽이 허용됩니다. 동일한 보안 그룹을 사용하지 않으려면 엔드포인트의 트래픽을 수락하도록 로드 밸런서에서 엔드포인트 보안 그룹을 참조해야 합니다.

h. 엔드포인트 도메인 접두사에 사용자 지정 식별자를 입력합니다. 예: **`my-ava-app`**. 이 접두사는 Verified Access가 생성하는 DNS 이름 앞에 추가됩니다.

6. 애플리케이션 세부 정보를 보려면 다음을 수행하십시오.

   a. 애플리케이션 도메인에 애플리케이션의 DNS 이름을 입력합니다. 이 도메인은 도메인 인증서의 도메인과 일치해야 합니다.

   b. 도메인 인증서 ARN에서 AWS Certificate Manager의 도메인 인증서의 Amazon 리소스 이름(ARN)을 선택합니다.

7. 정책 세부 정보를 비워 둡니다. 이후 단계에서 그룹 수준 액세스 정책을 추가합니다.

8. Verified·Access 엔드포인트 생성을 선택합니다.

5단계: Verified Access 엔드포인트에 대한 DNS 구성

이 단계에서는 애플리케이션의 도메인 이름(예: www.myapp.example.com)을 Verified·Access 엔드포인트의 도메인 이름에 매핑합니다. DNS 매핑을 완료하려면 DNS 공급자를 통해 Canonical Name Record(CNAME)를 생성합니다. CNAME 레코드를 생성하면 사용자가 애플리케이션에 보내는 모든 요청이 Verified·Access로 전송됩니다.

엔드포인트의 도메인 이름을 얻으려면

1. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
2. 엔드포인트를 선택합니다.
3. 세부 정보 탭을 선택하십시오.
4. 엔드포인트 도메인에서 도메인을 복사합니다. 다음은 엔드포인트 도메인 이름의 예입니다. `my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com`.

DNS 공급자가 제공한 지침에 따라 CNAME 레코드를 생성합니다. 애플리케이션의 도메인 이름을 레코드 이름으로 사용하고 Verified Access 엔드포인트의 도메인 이름을 레코드 값으로 사용합니다.

6단계: 애플리케이션에 대한 연결 테스트

이제 애플리케이션 연결을 테스트할 수 있습니다. 웹 브라우저에 애플리케이션의 도메인 이름을 입력합니다. Verified Access의 기본 동작은 모든 요청을 거부하는 것입니다. 그룹 또는 엔드포인트에 Verified Access 정책을 추가하지 않았으므로 모든 요청이 거부됩니다.

7단계: Verified Access 그룹 수준 액세스 정책 추가

다음 절차에 따라 Verified·Access 그룹을 수정하고 애플리케이션에 대한 연결을 허용하는 액세스 정책을 구성합니다. 정책의 세부 사항은 IAM Identity Center에 구성된 사용자 및 그룹에 따라 달라집니다. 자세한 내용은 Verified·Access 정책을 참조하세요.

Verified·Access 그룹을 수정하려면

1. 탐색 창에서 Verified·Access 그룹을 선택합니다.
2. 그룹을 선택합니다.
3. 작업, Verified·Access 그룹 정책 수정을 선택합니다.
4. 정책 활성화를 켭니다.
5. IAM Identity Center의 사용자가 애플리케이션에 액세스할 수 있도록 허용하는 정책을 입력합니다. 예시는 the section called “정책 예제” 섹션을 참조하세요.
6. Verified·Access 그룹 정책 수정을 선택합니다.
7. 이제 그룹 정책이 적용되었으므로 이전 단계의 테스트를 반복하여 요청이 허용되는지 확인합니다. 요청이 허용되는 경우 IAM Identity Center 로그인 페이지를 통해 로그인하라는 메시지가 표시됩니다. 사용자 이름과 암호를 입력한 후 애플리케이션에 액세스할 수 있습니다.

Verified Access 리소스 정리

이 자습서를 완료한 후 다음 절차에 따라 Verified Access 리소스를 삭제합니다.

Verified Access 리소스를 삭제하려면

1. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다. 엔드포인트를 선택하고 작업, Verified Access 엔드포인트 삭제 를 선택합니다.
2. 탐색 창에서 Verified·Access 그룹을 선택합니다. 그룹을 선택하고 작업, Verified Access 그룹 삭제를 선택합니다. 엔드포인트 삭제 프로세스가 완료될 때까지 기다려야 할 수 있습니다.

3. 탐색 창에서 Verified·Access 인스턴스를 선택합니다. 인스턴스를 선택하고 작업, Verified Access 신뢰 공급자 분리를 선택합니다. 신뢰 공급자를 선택하고 Verified Access 신뢰 공급자 분리를 선택합니다.
4. 탐색 창에서 Verified·Access 신뢰 공급자를 선택합니다. 신뢰 공급자를 선택하고 작업, Verified Access 신뢰 공급자 삭제를 선택합니다.
5. 탐색 창에서 Verified·Access 인스턴스를 선택합니다. 인스턴스를 선택하고 작업, Verified Access 인스턴스 삭제를 선택합니다.

Verified·Access 인스턴스

AWS Verified Access 인스턴스는 신뢰 공급자와 Verified Access 그룹을 구성하는 데 도움이 되는 AWS 리소스입니다. 인스턴스는 애플리케이션 요청을 평가하여 보안 요구 사항이 충족되는 경우에만 액세스 권한을 부여합니다.

업무

- Verified Access 인스턴스 생성 및 관리
- Verified·Access 인스턴스 삭제
- Verified Access를와 통합 AWS WAF
- Verified·Access의 FIPS 규정 준수

Verified Access 인스턴스 생성 및 관리

Verified Access 인스턴스는 신뢰 공급자와 Verified Access 그룹을 구성하는 데 사용됩니다. 다음 절차에 따라 Verified Access 인스턴스를 생성한 다음 Verified Access에 신뢰 공급자를 연결하거나 Verified Access에서 신뢰 공급자를 분리하세요.

업무

- Verified·Access 인스턴스 생성
- Verified Access 인스턴스에 신뢰 공급자 연결
- Verified Access 인스턴스에서 신뢰 공급자 분리
- 사용자 지정 하위 도메인 추가

Verified·Access 인스턴스 생성

다음 절차에 따라 Verified·Access 인스턴스를 생성하십시오.

콘솔을 사용하여 Verified Access 인스턴스를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택한 다음 Verified·Access 인스턴스 생성을 선택합니다.
3. (선택 사항) 이름 및 설명에 Verified·Access 인스턴스의 이름과 설명을 입력합니다.

4. (네트워크 CIDR 엔드포인트) 네트워크 CIDR 엔드포인트의 사용자 지정 하위 도메인에 사용자 지정 하위 도메인을 입력합니다.
5. (선택 사항) Verified Access가 FIPS를 준수하도록 하려면 FIPS(연방 정보 프로세스 표준) 활성화를 선택합니다.
6. (선택 사항) Verified Access 신뢰 공급자에서 Verified Access 인스턴스에 연결할 신뢰 공급자를 선택합니다.
7. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
8. Verified·Access 인스턴스 생성을 선택합니다.

를 사용하여 Verified Access 인스턴스를 생성하려면 AWS CLI

create-verified-access-instance 명령을 사용합니다.

Verified Access 인스턴스에 신뢰 공급자 연결

다음 절차에 따라 Verified Access 인스턴스에 신뢰 공급자를 연결하십시오.

콘솔을 사용하여 Verified Access 인스턴스에 신뢰 공급자를 연결하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, Verified·Access 신뢰 공급자 연결을 선택합니다.
5. Verified·Access 신뢰 공급자에서 신뢰 공급자를 선택합니다.
6. Verified·Access 신뢰 공급자 연결을 선택합니다.

를 사용하여 Verified Access 인스턴스에 신뢰 공급자를 연결하려면 AWS CLI

attach-verified-access-trust-provider 명령을 사용합니다.

Verified Access 인스턴스에서 신뢰 공급자 분리

다음 절차에 따라 Verified·Access 인스턴스에서 신뢰 공급자를 분리하십시오.

콘솔을 사용하여 Verified Access 인스턴스에서 신뢰 공급자를 분리하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, Verified·Access 신뢰 공급자 분리를 선택합니다.
5. Verified·Access 신뢰 공급자에서 신뢰 공급자를 선택합니다.
6. Verified·Access 신뢰 공급자 분리를 선택합니다.

를 사용하여 Verified Access 인스턴스에서 신뢰 공급자를 분리하려면 AWS CLI

detach-verified-access-trust-provider 명령을 사용합니다.

사용자 지정 하위 도메인 추가

다음 절차에 따라 사용자 지정 하위 도메인을 추가하거나 업데이트합니다. 이 하위 도메인은 네트워크 CIDR 엔드포인트를 생성할 때만 사용됩니다.

콘솔을 사용하여 사용자 지정 하위 도메인을 추가하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, Verified Access 인스턴스 수정을 선택합니다.
5. 네트워크 CIDR 엔드포인트에 대한 사용자 지정 하위 도메인에 사용자 지정 하위 도메인을 입력합니다.
6. Verified Access 인스턴스 수정을 선택합니다.
7. Verified Access에서 제공하는 네임서버를 입력하여 하위 도메인의 네임서버를 업데이트합니다. 이 목록은 인스턴스의 세부 정보 탭에 있는 Nameservers에서 사용할 수 있습니다.

를 사용하여 사용자 지정 하위 도메인을 추가하려면 AWS CLI

modify-verified-access-instance 명령을 사용합니다.

Verified·Access 인스턴스 삭제

Verified·Access 인스턴스를 마치면 이를 삭제할 수 있습니다. 인스턴스를 삭제하기 전에 먼저 연결된 신뢰 공급자 또는 Verified Access 그룹을 모두 제거해야 합니다.

콘솔을 사용하여 Verified Access 인스턴스를 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 작업, Verified·Access 인스턴스 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **delete**를 입력한 다음 삭제(Delete)를 선택합니다.

를 사용하여 Verified Access 인스턴스를 삭제하려면 AWS CLI

delete-verified-access-instance 명령을 사용합니다.

Verified Access를와 통합 AWS WAF

Verified Access에서 적용하는 인증 및 권한 부여 규칙 외에도 경계 보호를 적용할 수 있습니다. 이렇게 하면 추가 위협으로부터 애플리케이션을 보호하는 데 도움이 될 수 있습니다. Verified Access 배포 AWS WAF 에를 통합하여이 작업을 수행할 수 있습니다. AWS WAF 는 보호된 웹 애플리케이션 리소스로 전달되는 HTTP 요청을 모니터링할 수 있는 웹 애플리케이션 방화벽입니다. 자세한 내용은 개발자 안내서AWS WAF를 참조하세요.

AWS WAF 웹 액세스 제어 목록(ACL)을 Verified Access 인스턴스 AWS WAF 와 연결하여 Verified Access와 통합할 수 있습니다. 웹 ACL은 보호된 AWS WAF 리소스가 응답하는 모든 HTTP 웹 요청을 세밀하게 제어할 수 있는 리소스입니다. AWS WAF 연결 또는 연결 해제 요청이 처리되는 동안 인스턴스에 연결된 Verified Access 엔드포인트의 상태가 로 표시됩니다`updating`. 요청이 완료되면 상태가 `active`(으)로 돌아갑니다. 를 사용하여 엔드포인트를 설명하여 AWS Management Console 또는에서 상태를 볼 수 있습니다 AWS CLI.

사용자 자격 증명 신뢰 공급자가가 트래픽을 AWS WAF 검사하는 시기를 결정합니다. IAM Identity Center를 사용하는 경우는 사용자 인증 전에 트래픽을 AWS WAF 검사합니다. OpenID Connect(OIDC) AWS WAF 를 사용하는 경우는 사용자 인증 후 트래픽을 검사합니다.

내용

- 필수 IAM 권한
- AWS WAF 웹 ACL 연결
- 연결의 상태 확인
- AWS WAF 웹 ACL 연결 해제

필수 IAM 권한

Verified Access AWS WAF 와 통합에는 API 작업에 직접 해당하지 않는 권한 전용 작업이 포함됩니다. 이러한 작업은 AWS Identity and Access Management 서비스 승인 참조에 `[permission only]`(으)로 나와 있습니다. 자세한 정보는 서비스 승인 참조의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하십시오.

웹 ACL로 작업하려면 AWS Identity and Access Management 보안 주체에 다음 권한이 있어야 합니다.

- `ec2:AssociateVerifiedAccessInstanceWebAcl`
- `ec2:DisassociateVerifiedAccessInstanceWebAcl`
- `ec2:DescribeVerifiedAccessInstanceWebAclAssociations`
- `ec2:GetVerifiedAccessInstanceWebAcl`

AWS WAF 웹 ACL 연결

다음 단계에서는 Verified Access 콘솔을 사용하여 AWS WAF 웹 액세스 제어 목록(ACL)을 Verified Access 인스턴스와 연결하는 방법을 보여줍니다.

전제 조건

시작하기 전에 AWS WAF 웹 ACL을 생성합니다. 자세한 정보는 AWS WAF 개발자 안내서의 웹 ACL 생성을 참조하세요.

AWS WAF 웹 ACL을 Verified Access 인스턴스에 연결하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 통합 탭을 선택합니다.
5. 작업을 선택한 다음 웹 ACL 연결을 선택합니다.
6. 웹 ACL에서 기존 웹 ACL을 선택한 다음 웹 ACL 연결을 선택합니다.

또는 AWS WAF 콘솔을 사용할 수 있습니다. AWS WAF 콘솔 또는 API를 사용하는 경우 Verified Access 인스턴스의 Amazon 리소스 이름(ARN)이 필요합니다. AVA ARN은 `arn:`

`${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}` 형식을 갖습니다. 자세한 내용은 AWS WAF 개발자 안내서의 웹 ACL을 AWS 리소스와 연결을 참조하세요.

연결의 상태 확인

Verified Access 콘솔을 사용하여 AWS WAF 웹 액세스 제어 목록(ACL)이 Verified Access 인스턴스와 연결되어 있는지 여부를 확인할 수 있습니다.

Verified Access 인스턴스와의 AWS WAF 통합 상태를 보려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 통합 탭을 선택합니다.
5. WAF 통합 상태 아래에 나열된 세부 정보를 확인합니다. 연결 상태인 경우 상태는 웹 ACL 식별자와 함께 연결됨 또는 연결되지 않음으로 표시됩니다.

AWS WAF 웹 ACL 연결 해제

다음 단계에서는 Verified Access 콘솔을 사용하여 Verified Access 인스턴스에서 AWS WAF 웹 액세스 제어 목록(ACL)의 연결을 해제하는 방법을 보여줍니다.

Verified Access 인스턴스에서 AWS WAF 웹 ACL을 연결 해제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 통합 탭을 선택합니다.
5. 작업을 선택한 다음 웹 ACL 연결 해제를 선택합니다.
6. 웹 ACL 연결 해제를 선택하여 확인합니다.

또는 AWS WAF 콘솔을 사용할 수 있습니다. 자세한 내용은 AWS WAF 개발자 안내서의 AWS 리소스에서 웹 ACL 연결 해제를 참조하세요.

Verified·Access의 FIPS 규정 준수

연방 정보 처리 표준(FIPS)은 민감한 정보를 보호하는 암호화 모듈의 보안 요구 사항을 지정하는 미국 및 캐나다 정부 표준입니다.는 FIPS 간행물 140-2를 준수하도록 환경을 구성하는 옵션을 AWS Verified Access 제공합니다. Verified Access에 대한 FIPS 규정 준수는 다음 AWS 리전에서 사용할 수 있습니다.

- 미국 동부(오하이오)
- 미국 동부(버지니아 북부)
- 미국 서부(캘리포니아 북부)
- 미국 서부(오레곤)
- 캐나다(중부)
- AWS GovCloud (US) 서부
- AWS GovCloud (US) 동부

이 페이지에서는 FIPS를 준수하도록 신규 또는 기존의 Verified Access 환경을 구성하는 방법을 보여줍니다.

내용

- FIPS 규정 준수를 위해 기존 Verified Access 환경 구성
- FIPS 규정 준수를 위한 새로운 Verified·Access 환경 구성

FIPS 규정 준수를 위해 기존 Verified Access 환경 구성

기존 Verified Access 환경이 있고 FIPS를 준수하도록 구성하려는 경우, FIPS 규정 준수를 활성화하려면 일부 리소스를 삭제하고 다시 생성해야 합니다.

기존 AWS Verified Access 환경을 FIPS 호환으로 다시 구성하려면 아래 단계를 따르세요.

1. 원래 Verified·Access 엔드포인트, 그룹 및 인스턴스를 삭제합니다. 구성된 신뢰 공급자는 재사용할 수 있습니다.
2. Verified·Access 인스턴스를 생성하고 생성 시 연방 정보 처리 표준(FIPS)이 활성화되도록 합니다. 또한 생성 중에 사용하려는 Verified·Access 신뢰 공급자를 드롭다운 목록에서 선택하여 연결합니다.

3. Verified·Access 그룹을 생성합니다. 그룹을 생성하는 동안 그룹을 방금 만든 Verified·Access 인스턴스와 연결합니다.
4. 하나 이상의 Verified·Access 엔드포인트를 생성합니다. 엔드포인트를 생성하는 동안 이전 단계에서 생성한 그룹과 엔드포인트를 연결합니다.

FIPS 규정 준수를 위한 새로운 Verified·Access 환경 구성

FIPS를 준수하는 새 AWS Verified Access 환경을 구성하려면 아래 단계를 따릅니다.

1. 신뢰 공급자를 구성합니다. 필요에 따라 사용자 자격 증명 신뢰 공급자와 디바이스 기반 신뢰 공급자(선택 사항)를 생성해야 합니다.
2. 프로세스 중에 연방 정보 처리 표준(FIPS)을 활성화하여 Verified·Access 인스턴스를 생성합니다. 또한 생성 중에 이전 단계에서 만든 Verified·Access 신뢰 공급자를 드롭다운 목록에서 선택하여 연결합니다.
3. Verified·Access 그룹을 생성합니다. 그룹을 생성하는 동안 그룹을 방금 만든 Verified·Access 인스턴스와 연결합니다.
4. 하나 이상의 Verified·Access 엔드포인트를 생성합니다. 엔드포인트를 생성하는 동안 이전 단계에서 생성한 그룹과 엔드포인트를 연결합니다.

Verified·Access에 대한 신뢰 공급자

신뢰 공급자는 사용자 및 디바이스에 대한 정보를 보내는 서비스입니다 AWS Verified Access. 이러한 정보를 신뢰 컨텍스트라고 합니다. 여기에는 이메일 주소, "영업" 조직의 멤버십 등 사용자 ID에 기반한 속성이나 설치된 보안 패치, 바이러스 백신 소프트웨어 버전 등 디바이스 정보가 포함될 수 있습니다.

Verified Access는 다음 범주의 신뢰 공급자를 지원합니다.

- 사용자 자격 증명 - 사용자의 디지털 ID를 저장하고 관리하는 ID 제공업체(IdP) 서비스입니다.
- 디바이스 관리 - 노트북, 태블릿, 스마트폰과 같은 디바이스의 디바이스 관리 시스템입니다.

내용

- Verified Access의 사용자 자격 증명 신뢰 공급자
- Verified Access의 디바이스 기반 신뢰 공급자

Verified Access의 사용자 자격 증명 신뢰 공급자

AWS IAM Identity Center 또는 OpenID Connect 호환 사용자 자격 증명 신뢰 공급자를 사용하도록 선택할 수 있습니다.

내용

- 신뢰 공급자로 IAM Identity Center 사용
- OpenID Connect 신뢰 공급자 사용

신뢰 공급자로 IAM Identity Center 사용

AWS Verified·Access에서를 사용자 자격 증명 신뢰 공급자 AWS IAM Identity Center 로 사용할 수 있습니다.

필수 조건 및 고려 사항

- IAM Identity Center 인스턴스는 AWS Organizations 인스턴스여야 합니다. 독립 실행형 AWS 계정 IAM Identity Center 인스턴스는 작동하지 않습니다.
- Verified Access 신뢰 공급자를 생성하려는 리전과 동일한 AWS 리전에서 IAM Identity Center 인스턴스를 활성화해야 합니다.

- Verified Access는 최대 1,000개의 그룹에 할당된 IAM Identity Center의 사용자에게 액세스 권한을 제공할 수 있습니다.

다양한 인스턴스 유형에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center의 조직 및 계정 인스턴스 관리를 참조하세요.

IAM Identity Center 신뢰 공급자 생성

AWS 계정에서 IAM Identity Center를 활성화한 후 다음 절차를 사용하여 IAM Identity Center를 Verified Access의 신뢰 공급자로 설정할 수 있습니다.

IAM Identity Center 신뢰 공급자를 생성하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 신뢰 공급자의 이름과 설명을 입력합니다.
4. 정책 참조 이름에 나중에 정책 규칙 작업 시 사용할 식별자를 입력합니다.
5. 신뢰 공급자 유형에서 사용자 신뢰 공급자를 선택합니다.
6. 사용자 신뢰 공급자 유형에서 IAM Identity Center를 선택합니다.
7. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
8. Verified·Access 신뢰 공급자 생성을 선택합니다.

IAM Identity Center 신뢰 공급자를 생성하려면(AWS CLI)

- create-verified-access-trust-provider (AWS CLI)

IAM Identity Center 신뢰 공급자 삭제

신뢰 공급자를 삭제하기 전에 먼저 신뢰 공급자가 연결된 인스턴스에서 모든 엔드포인트 및 그룹 구성을 제거해야 합니다.

IAM Identity Center 신뢰 공급자를 삭제하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자에서 삭제하려는 신뢰 공급자를 선택합니다.

3. 작업을 선택한 다음 Verified·Access 신뢰 공급자 삭제를 선택합니다.
4. 텍스트 상자에 `delete`를 입력하여 삭제를 확인합니다.
5. 삭제를 선택합니다.

IAM Identity Center 신뢰 공급자를 삭제하려면(AWS CLI)

- delete-verified-access-trust-provider (AWS CLI)

OpenID Connect 신뢰 공급자 사용

AWS Verified Access 는 표준 OpenID Connect(OIDC) 메서드를 사용하는 자격 증명 공급자를 지원합니다. OIDC 호환 공급자를 Verified·Access를 통한 사용자 자격 증명 신뢰 공급자로 사용할 수 있습니다. 그러나 다양한 잠재적 OIDC 공급자로 인해 AWS 는 Verified Access와의 각 OIDC 통합을 테스트할 수 없습니다.

Verified·Access는 OIDC 공급자의 `UserInfo Endpoint`(가) 평가하는 신뢰 데이터를 얻습니다. `Scope` 파라미터는 검색할 신뢰 데이터 집합을 결정하는 데 사용됩니다. 신뢰 데이터를 수신한 후에는 이를 기준으로 Verified·Access 정책이 평가됩니다.

OIDC 신뢰 공급자의 ID 토큰 클레임은 2025년 2월 24일 이후에 생성된 신뢰 공급자의 `addition_user_context` 키에 포함됩니다.

2025년 2월 24일 또는 그 이전에 생성된 신뢰 공급자의 경우 Verified·Access는 OIDC 공급자가 `ID token` 전송한의 신뢰 데이터를 사용하지 않습니다. `UserInfo Endpoint`의 신뢰 데이터만 정책에 따라 평가됩니다.

2025년 2월 24일 또는 그 이전에 생성된 신뢰 공급자의 경우 기본 세션 기간은 1일입니다. 2025년 2월 24일 이전에 생성된 신뢰 공급자의 경우 기본 세션 기간은 7일입니다.

새로 고침 토큰이 지정된 경우 Verified·Access는 새로 고침 토큰의 만료를 세션 기간으로 사용합니다. 새로 고침 토큰이 없는 경우 기본 세션 기간이 사용됩니다.

내용

- OIDC 신뢰 공급자를 생성하기 위한 사전 조건
- OIDC 신뢰 공급자 생성
- OIDC 신뢰 공급자 수정
- OIDC 신뢰 공급자 삭제

OIDC 신뢰 공급자를 생성하기 위한 사전 조건

신뢰 공급자 서비스에서 직접 다음 정보를 수집해야 합니다.

- Issuer
- 권한 부여 엔드포인트
- Token 엔드포인트
- UserInfo 엔드포인트
- 클라이언트 ID
- 클라이언트 보안 암호(client secret)
- 범위

OIDC 신뢰 공급자 생성

다음 절차에 따라 OIDC를 신뢰 공급자로 생성하십시오.

OIDC 신뢰 공급자를 생성하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 신뢰 공급자의 이름과 설명을 입력합니다.
4. 정책 참조 이름에 나중에 정책 규칙 작업 시 사용할 식별자를 입력합니다.
5. 신뢰 공급자 유형에서 사용자 신뢰 공급자를 선택합니다.
6. 사용자 신뢰 공급자 유형에서 OIDC(OpenID Connect)를 선택합니다.
7. OIDC(OpenID Connect)에서 신뢰 공급자를 선택합니다.
8. 발급자에 OIDC 발급자의 식별자를 입력합니다.
9. 권한 부여 엔드포인트에 권한 부여 엔드포인트의 전체 URL을 입력합니다.
10. 토큰 엔드포인트에 토큰 엔드포인트의 전체 URL을 입력합니다.
11. 사용자 엔드포인트에 사용자 엔드포인트의 전체 URL을 입력합니다.
12. (기본 애플리케이션 OIDC) 퍼블릭 서명 키 URL에 퍼블릭 서명 키 엔드포인트의 전체 URL을 입력합니다.
13. 클라이언트 ID에 OAuth 2.0 클라이언트 식별자를 입력합니다.
14. 클라이언트 암호에 OAuth 2.0 클라이언트 암호를 입력합니다.

15. 자격 증명 공급자가 정의한 공백으로 구분된 범위 목록을 입력합니다. 최소한 openid 범위는 범위에 필요합니다.
16. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
17. Verified·Access 신뢰 공급자 생성을 선택합니다.
18. OIDC 공급자의 허용 목록에 리디렉션 URI를 추가해야 합니다.
    - HTTP 애플리케이션 - URI를 사용합니다**`https://application_domain/oauth2/idpresponse`**. 콘솔의 Verified·Access 엔드포인트에 대한 세부 정보 탭에서 애플리케이션 도메인을 찾을 수 있습니다. AWS CLI 또는 AWS SDK를 사용하면 Verified·Access 엔드포인트를 설명할 때 애플리케이션 도메인이 출력에 포함됩니다.
    - TCP 애플리케이션 - URI를 사용합니다**`http://localhost:8000`**.

OIDC 신뢰 공급자를 생성하려면(AWS CLI)

- create-verified-access-trust-provider (AWS CLI)

OIDC 신뢰 공급자 수정

신뢰 공급자 생성 후 해당 구성을 업데이트할 수 있습니다.

OIDC 신뢰 공급자를 수정하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자에서 수정하려는 신뢰 공급자를 선택합니다.
3. 작업을 선택한 다음 Verified·Access 신뢰 공급자 수정을 선택합니다.
4. 변경할 옵션을 수정합니다.
5. Verified·Access 신뢰 공급자 수정을 선택합니다.

OIDC 신뢰 공급자를 수정하려면(AWS CLI)

- modify-verified-access-trust-provider (AWS CLI)

OIDC 신뢰 공급자 삭제

사용자 신뢰 공급자를 삭제하기 전에 먼저 신뢰 공급자가 연결된 인스턴스에서 모든 엔드포인트 및 그룹 구성을 제거해야 합니다.

OIDC 신뢰 공급자를 삭제하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자에서 삭제하려는 신뢰 공급자를 선택합니다.
3. 작업을 선택한 다음 Verified·Access 신뢰 공급자 삭제를 선택합니다.
4. 텍스트 상자에 `delete`를 입력하여 삭제를 확인합니다.
5. 삭제를 선택합니다.

OIDC 신뢰 공급자를 삭제하려면(AWS CLI)

- delete-verified-access-trust-provider (AWS CLI)

Verified Access의 디바이스 기반 신뢰 공급자

AWS Verified·Access에서 디바이스 신뢰 공급자를 사용할 수 있습니다. Verified Access 인스턴스를 통해 여러 디바이스 신뢰 공급자를 사용할 수 있습니다.

내용

- 지원되는 디바이스 신뢰 공급자
- 디바이스 기반 신뢰 공급자 생성
- 디바이스 기반 신뢰 공급자 수정
- 디바이스 기반 신뢰 공급자 삭제

지원되는 디바이스 신뢰 공급자

다음 디바이스 신뢰 공급자는 Verified Access와 통합될 수 있습니다.

- CrowdStrike - CrowdStrike 및 AWS Verified Access를 사용하여 프라이빗 애플리케이션 보호
- Jamf - Verified·Access를 Jamf 디바이스 자격 증명과 통합
- JumpCloud - JumpCloud와 AWS Verified Access 통합

디바이스 기반 신뢰 공급자 생성

다음 단계에 따라 Verified·Access와 함께 사용할 디바이스 신뢰 공급자를 생성하고 구성하십시오.

Verified Access 디바이스 신뢰 공급자를 생성하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 신뢰 공급자의 이름과 설명을 입력합니다.
4. 나중에 정책 참조 이름에 대한 정책 규칙을 작업할 때 사용할 식별자를 입력합니다.
5. 신뢰 공급자 유형에서 디바이스 자격 증명을 선택합니다.
6. 디바이스 자격 증명 유형에서 Jamf, CrowdStrike, 또는 JumpCloud를 선택합니다.
7. 테넌트 ID에는 테넌트 애플리케이션의 식별자를 입력합니다.
8. (선택 사항) 퍼블릭 서명 키 URL에서 디바이스 신뢰 공급자가 공유하는 고유한 키 URL을 입력합니다. (이 파라미터는 Jamf, CrowdStrike 또는 Jumpcloud에는 필요하지 않습니다.)
9. Verified Access 신뢰 공급자 생성을 선택합니다.

> **Note**
>
> OIDC 공급자의 허용 목록에 리디렉션 URI를 추가해야 합니다. 이 용도로는 Verified·Access 엔드포인트의 `DeviceValidationDomain`를 사용하는 것이 좋습니다. 이는 Verified·Access 엔드포인트의 세부 정보 탭 AWS Management Console아래 또는를 사용하여 엔드포인트를 설명하는 AWS CLI 방법으로 찾을 수 있습니다. OIDC 공급자의 허용 목록에 다음을 추가하십시오. https://`DeviceValidationDomain`/oauth2/idpresponse

Verified Access 디바이스 신뢰 공급자(AWS CLI)를 생성하려면

- create-verified-access-trust-provider (AWS CLI)

디바이스 기반 신뢰 공급자 수정

신뢰 공급자 생성 후 해당 구성을 업데이트할 수 있습니다.

Verified Access 디바이스 신뢰 공급자를 수정하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택합니다.
3. 신뢰 공급자를 선택합니다.
4. 작업을 선택한 다음 Verified·Access 신뢰 공급자 수정을 선택합니다.
5. 필요에 따라 설명을 수정합니다.
6. (선택 사항) 퍼블릭 서명 키 URL에서 디바이스 신뢰 공급자가 공유하는 고유한 키 URL을 수정합니다. (디바이스 신뢰 제공자가 Jamf, CrowdStrike 또는 Jumpcloud인 경우에는 이 파라미터가 필요하지 않습니다.)
7. Verified Access 신뢰 공급자 수정을 선택합니다.

Verified Access 디바이스 신뢰 공급자(AWS CLI)를 수정하려면

- modify-verified-access-trust-provider (AWS CLI)

디바이스 기반 신뢰 공급자 삭제

신뢰 공급자 사용을 마치면 이를 삭제할 수 있습니다.

Verified Access 디바이스 신뢰 공급자를 삭제하려면(AWS 콘솔)

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택합니다.
3. Verified·Access 신뢰 공급자에서 삭제하려는 신뢰 공급자를 선택합니다.
4. 작업을 선택한 다음 Verified·Access 신뢰 공급자 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **`delete`**를 입력한 다음 삭제(Delete)를 선택합니다.

Verified Access 디바이스 신뢰 공급자(AWS CLI)를 삭제하려면

- delete-verified-access-trust-provider (AWS CLI)

Verified·Access 그룹

Verified Access 그룹은 Verified Access 엔드포인트와 그룹의 모든 엔드포인트에 적용되는 Verified Access 정책으로 구성됩니다. 일반적인 보안 요구 사항이 있는 엔드포인트를 그룹화하여 여러 엔드포인트의 최소 보안 요구 사항을 충족하는 단일 그룹 정책을 정의할 수 있습니다. 따라서 각 엔드포인트에 대한 정책을 생성하고 유지 관리할 필요가 없습니다.

예를 들어 모든 판매 애플리케이션을 그룹화하고 그룹 전체 액세스 정책을 설정할 수 있습니다. 그런 다음 이 정책을 사용하여 모든 판매 애플리케이션에 대해 하나의 공통된 최소 보안 요구 사항 집합을 정의할 수 있습니다. 이 접근 방식은 정책 관리를 단순화하는 데 도움이 됩니다.

그룹을 생성할 때 Verified·Access 인스턴스와 그룹을 연결해야 합니다. 엔드포인트를 생성하는 과정에서 엔드포인트를 그룹과 연결합니다.

Verified·Access 그룹의 또 다른 기능은를 사용하여 다른 AWS 계정과 공유할 수 있다는 것입니다 AWS RAM. 이렇게 하면 한 계정에서 중앙 집중식으로 그룹을 생성 및 관리하며 여러 계정과 공유할 수 있습니다.

업무

- Verified Access 그룹 생성 및 관리
- Verified·Access 그룹 정책 수정
- Verified Access 그룹을 다른와 공유 AWS 계정
- Verified·Access 그룹 삭제

Verified Access 그룹 생성 및 관리

Verified Access 그룹을 사용하여 보안 요구 사항에 따라 엔드포인트를 구성합니다. Verified·Access 엔드포인트를 생성할 때 엔드포인트를 그룹과 연결합니다.

업무

- Verified·Access 그룹 생성
- Verified Access 그룹 수정

Verified·Access 그룹 생성

다음 절차에 따라 Verified Access 그룹을 생성합니다. Verified·Access 그룹을 생성하기 전에 Verified·Access 인스턴스를 생성해야 합니다. 자세한 내용은 the section called " Verified·Access 인스턴스 생성" 단원을 참조하십시오.

콘솔을 사용하여 Verified Access 그룹을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 그룹을 선택한 다음 Verified·Access 그룹 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 그룹의 이름과 설명을 입력합니다.
4. Verified·Access 인스턴스의 경우 그룹과 연결할 Verified·Access 인스턴스를 선택합니다.
5. (선택 사항) 정책 정의에는 그룹에 적용할 Verified·Access 정책을 입력합니다.
6. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
7. Verified·Access 그룹 생성을 선택합니다.

를 사용하여 Verified Access 그룹을 생성하려면 AWS CLI

create-verified-access-group 명령을 사용합니다.

Verified Access 그룹 수정

Verified Access 그룹을 수정하려면 다음 절차를 따르세요.

콘솔을 사용하여 Verified Access 그룹을 수정하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 그룹을 선택한 다음 Verified·Access 그룹 생성을 선택합니다.
3. 그룹을 선택한 다음 작업, Verified Access 그룹 수정을 선택합니다.
4. (선택 사항) 설명을 업데이트합니다.
5. Verified·Access 그룹 생성을 선택합니다.
6. 그룹과 연결할 Verified·Access 인스턴스를 선택합니다.

를 사용하여 Verified Access 그룹을 수정하려면 AWS CLI

modify-verified-access-group 명령을 사용합니다.

Verified·Access 그룹 정책 수정

AWS Verified Access 는 사용자가 생성한 액세스 정책에 따라 애플리케이션에 대한 액세스를 허용합니다. 그룹에 연결하는 Verified·Access 정책은 그룹의 모든 엔드포인트에서 상속됩니다. 선택적으로 애플리케이션별 정책을 특정 엔드포인트에 연결할 수 있습니다.

다음 절차에 따라 Verified Access 그룹 정책을 수정하세요. 변경한 후에는 변경 사항이 적용되기까지 몇 분 정도 걸립니다.

콘솔을 사용하여 Verified Access 그룹 정책을 수정하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 그룹을 선택합니다.
3. 그룹을 선택합니다.
4. 작업, Verified·Access 그룹 정책 수정을 선택합니다.
5. (선택 사항) 필요에 따라 정책 활성화를 켜거나 끕니다.
6. (선택 사항) 정책에 그룹에 적용할 Verified Access 정책을 입력합니다.
7. Verified·Access 그룹 정책 수정을 선택합니다.

를 사용하여 Verified Access 그룹 정책을 수정하려면 AWS CLI

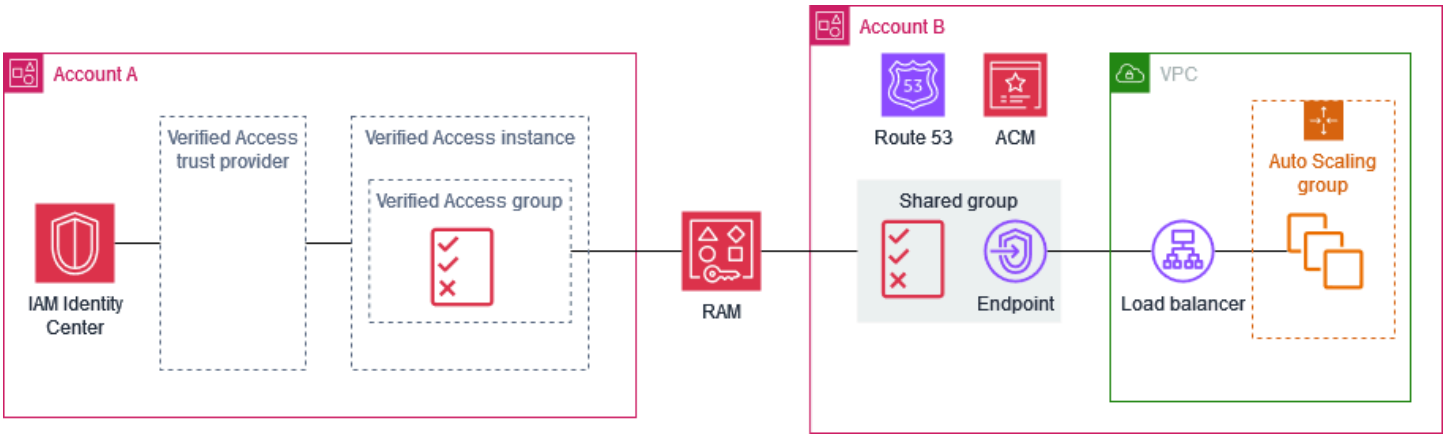
modify-verified-access-group-policy 명령을 사용합니다.

Verified Access 그룹을 다른와 공유 AWS 계정

소유한 Verified·Access 그룹을 다른 AWS 계정과 공유하는 경우 해당 계정이 그룹에서 Verified·Access 엔드포인트를 생성하도록 활성화합니다. Verified Access 그룹을 생성한 계정을 소유자 계정이라고 합니다. 공유 그룹을 사용하는 계정을 소비자 계정이라고 합니다.

다음 다이어그램은 Verified Access 그룹을 공유하는 것의 이점을 보여줍니다. 중앙 보안 팀은 계정 A를 소유합니다. 중앙 보안 팀은의 사용자 및 그룹을 관리하고 Verified·Access 신뢰 공급자 AWS IAM Identity Center, Verified·Access 인스턴스, Verified·Access 그룹 및 Verified·Access 정책과 같은 내부 애플리케이션에 대한 액세스를 제공하는 데 필요한 Verified·Access 리소스를 관리합니다. 애플리케이션 팀은 계정 B를 소유합니다. 로드 밸런서, Auto Scaling 그룹, Amazon Route 53의 DNS 구성, AWS Certificate Manager (ACM)의 TLS 인증서 등 내부 애플리케이션을 실행하는 데 필요한 리소스를 관리합니다. 중앙 보안 팀이 Verified Access 그룹을 계정 B와 공유한 후 애플리케이션 팀은 공유 그룹을 사

용하여 Verified Access 엔드포인트를 생성할 수 있습니다. 중앙 보안 팀이 Verified Access 그룹에 대해 생성한 정책에 따라 애플리케이션에 대한 액세스가 허용되거나 거부됩니다.



고려 사항

공유 Verified Access 그룹에는 다음 고려 사항이 적용됩니다.

소유자

- Verified Access 그룹을 공유하려면 사용자에게 `ec2:PutResourcePolicy` 및 `ec2:DeleteResourcePolicy` 권한이 있어야 합니다.
- Verified Access 그룹을 소유해야만 공유할 수 있습니다. 본인과 공유된 Verified Access 그룹은 공유할 수 없습니다.
- 조직의 계정과 공유를 활성화하면 초대를 사용하지 않고 Verified Access 그룹과 같은 리소스를 공유할 수 있습니다. 그렇지 않으면 소비자가 초대를 받고 초대를 수락해야만 공유 그룹에 액세스할 수 있습니다. 공유를 활성화하려면 조직의 관리 계정에서 AWS RAM 콘솔의 설정 페이지를 열고 공유 활성화를 AWS Organizations 선택합니다.
- 연결된 Verified Access 엔드포인트가 있는 경우 그룹을 삭제할 수 없습니다. 계정의 Verified Access 엔드포인트 페이지에서 소비자 계정에 의해 생성된 엔드포인트를 볼 수 있습니다. 엔드포인트 소유자의 계정 ID는 엔드포인트에 대한 인증서의 Amazon 리소스 이름(ARN)에 반영됩니다.

소비자

- 본인과 공유된 Verified Access 그룹을 보려면 콘솔에서 Verified Access 그룹 페이지를 열거나 describe-verified-access-groups를 직접적으로 호출합니다. 소유자의 계정 ID는 소유자 필드와 그룹의 Amazon 리소스 이름(ARN)에 반영됩니다.
- Verified Access 엔드포인트를 생성할 때 본인과 공유된 Verified Access 그룹을 지정할 수 있습니다.

- 공유 그룹과 연결되어 있지만 본인이 소유하지 않은 엔드포인트는 볼 수 없습니다.
- Verified Access 그룹의 소유자가 리소스 공유를 삭제하면 그룹에서 새 Verified Access 엔드포인트를 생성할 수 없습니다. 리소스 공유 삭제 전에 생성한 Verified Access 엔드포인트는 리소스 공유 삭제의 영향을 받지 않습니다. 그러나 공유 그룹의 소유자는 이러한 엔드포인트를 삭제할 수 있습니다.

리소스 공유

Verified·Access 그룹을 공유하려면 리소스 공유에 추가해야 합니다. 리소스 공유는 공유할 리소스와 공유 리소스를 사용할 수 있는 소비자를 지정합니다.

콘솔을 사용하여 Verified Access 그룹을 공유하려면

1. https://console.aws.amazon.com/ram/home AWS RAM 콘솔을 엽니다.
2. 조직에 대한 리소스 공유가 없는 경우 리소스 공유를 생성합니다. 보안 주체의 경우 전체 조직, 조직 단위 또는 특정 AWS 계정을 선택할 수 있습니다.
3. 리소스 공유를 선택한 다음 수정을 선택합니다.
4. `Resources`에서 리소스 유형으로 Verified Access 그룹을 선택한 다음 공유할 리소스 그룹을 선택합니다.
5. 건너뛰기: 검토 및 업데이트를 선택합니다.
6. 리소스 공유 업데이트를 선택합니다.

자세한 내용은 AWS RAM 사용 설명서의 리소스 공유 생성을 참조하세요.

Verified·Access 그룹 삭제

Verified·Access 그룹 사용을 마치면 이를 삭제할 수 있습니다. 연결된 Verified Access 엔드포인트가 있는 경우 그룹을 삭제할 수 없습니다.

콘솔을 사용하여 Verified Access 그룹을 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 그룹을 선택합니다.
3. 그룹을 선택합니다.
4. 작업, Verified·Access 그룹 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **`delete`**를 입력한 다음 삭제(Delete)를 선택합니다.

를 사용하여 Verified Access 그룹을 삭제하려면 AWS CLI

delete-verified-access-group 명령을 사용합니다.

Verified·Access 엔드포인트

Verified·Access 엔드포인트는 애플리케이션을 나타냅니다. 각 엔드포인트는 Verified Access 그룹과 연결되며 그룹에 대한 액세스 정책을 상속합니다. 필요에 따라 애플리케이션별 엔드포인트 정책을 각 엔드포인트에 연결할 수 있습니다.

내용

- Verified·Access 엔드포인트 유형
- 공유 VPC 및 서브넷에서 Verified Access의 작동 방식
- Verified Acces에 대한 로드 밸런서 엔드포인트 생성
- Verified·Access에 대한 네트워크 인터페이스 엔드포인트 생성
- Verified·Access를 위한 네트워크 CIDR 엔드포인트 생성
- Verified·Access를 위한 Amazon Relational Database Service 엔드포인트 생성
- Verified·Access 엔드포인트에서 발생하는 트래픽 허용
- Verified·Access 엔드포인트 수정
- Verified·Access 엔드포인트 정책 수정
- Verified·Access 엔드포인트 삭제

Verified·Access 엔드포인트 유형

가능한 Verified Access 엔드포인트 유형은 다음과 같습니다.

- 로드 밸런서 – 애플리케이션 요청이 로드 밸런서로 전송되어 애플리케이션에 배포됩니다. 자세한 내용은 로드 밸런서 엔드포인트 생성 단원을 참조하십시오.
- 네트워크 인터페이스 - 애플리케이션 요청은 지정된 프로토콜 및 포트를 사용하여 네트워크 인터페이스로 전송됩니다. 자세한 내용은 네트워크 인터페이스 엔드포인트 생성 단원을 참조하십시오.
- 네트워크 CIDR - 애플리케이션 요청이 지정된 CIDR 블록으로 전송됩니다. 자세한 내용은 네트워크 CIDR 엔드포인트 생성 단원을 참조하십시오.
- Amazon Relational Database Service(RDS) - 애플리케이션 요청은 RDS 인스턴스, RDS 클러스터 또는 RDS DB 프록시로 전송됩니다. 자세한 내용은 Amazon Relational Database Service 엔드포인트 생성 단원을 참조하십시오.

공유 VPC 및 서브넷에서 Verified Access의 작동 방식

공유 VPC 서브넷과 관련된 동작은 다음과 같습니다.

- Verified·Access 엔드포인트는 VPC 서브넷 공유를 통해 지원됩니다. 참여자는 공유 서브넷에서 Verified Access 엔드포인트를 생성할 수 있습니다.
- 엔드포인트를 생성한 참여자가 엔드포인트 소유자이며, 엔드포인트를 수정할 수 있는 유일한 당사자가 됩니다. VPC 소유자는 엔드포인트를 수정할 수 없습니다.
- Verified·Access 엔드포인트는 AWS 로컬 영역에서 생성할 수 없으므로 로컬 영역을 통해 공유할 수 없습니다.

자세한 내용은 Amazon VPC 사용 설명서의 다른 계정과 VPC 공유하기를 참조하십시오.

Verified Acces에 대한 로드 밸런서 엔드포인트 생성

다음 절차에 따라 Verified Access용 로드 밸런서 엔드포인트를 생성하세요. 로드 밸런서에 대한 자세한 내용은 Elastic Load Balancing 사용 설명서를 참조하십시오.

요구 사항

- IPv4 트래픽만 지원됩니다.
- WebSocket 연결과 같은 수명이 긴 HTTPS 연결은 TCP를 통해서만 지원됩니다.
- 로드 밸런서는 Application Load Balancer 또는 Network Load Balancer여야 하며, 내부 로드 밸런서여야 합니다.
- 로드 밸런서와 서브넷은 동일한 Virtual Private Cloud(VPC)에 속해야 합니다.
- HTTPS 로드 밸런서는 자체 서명 또는 공용 TLS 인증서를 사용할 수 있습니다. 키 길이가 1,024 또는 2,048인 RSA 인증서를 사용합니다.
- Verified·Access 엔드포인트를 생성하기 전에 Verified·Access 그룹을 생성해야 합니다. 자세한 내용은 the section called “Verified·Access 그룹 생성” 단원을 참조하십시오.
- 애플리케이션에 사용할 도메인 이름을 제공해야 합니다. 이 필드는 사용자가 애플리케이션에 액세스하는 데 사용하는 공용 DNS 이름입니다. 또한 이 도메인 이름과 일치하는 CN이 포함된 공용 SSL 인증서를 제공해야 합니다. 를 사용하여 인증서를 생성하거나 가져올 수 있습니다 AWS Certificate Manager.

콘솔을 사용하여 로드 밸런서 엔드포인트를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. Verified·Access 엔드포인트 생성을 선택합니다.
4. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
5. Verified·Access 그룹에서 Verified·Access 그룹을 선택합니다.
6. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 프로토콜에서 프로토콜을 선택합니다.
   b. 연결 유형(Attachment type)에서 VPC를 선택합니다.
   c. 엔드포인트 유형에서 로드 밸런서를 선택합니다.
   d. (HTTP/HTTPS) 포트에 포트 번호를 입력합니다. (TCP) 포트 범위에 포트 범위를 입력하고 포트 추가를 선택합니다.
   e. 로드 밸런서 ARN에서 로드 밸런서를 선택합니다.
   f. 서브넷에서 서브넷을 선택합니다. 가용 영역당 1개의 서브넷만 지정할 수 있습니다.
   g. 보안 그룹(Security groups)에서 엔드포인트의 보안 그룹을 선택합니다. 이러한 보안 그룹은 Verified Access 엔드포인트의 인바운드 및 아웃바운드 트래픽을 제어합니다.
   h. 엔드포인트 도메인 접두사의 경우 Verified Access가 엔드포인트에 대해 생성하는 DNS 이름 앞에 사용자 지정 식별자를 입력합니다.
7. (HTTP/HTTPS) 애플리케이션 세부 정보에서 다음을 수행합니다.
   a. 애플리케이션 도메인에 애플리케이션의 DNS 이름을 입력합니다.
   b. 도메인 인증서 ARN에서 퍼블릭 TLS 인증서를 선택합니다.
8. (선택 사항) 정책 정의에는 엔드포인트에 대한 Verified·Access 정책을 입력합니다.
9. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
10. Verified·Access 엔드포인트 생성을 선택합니다.

를 사용하여 Verified Access 엔드포인트를 생성하려면 AWS CLI

create-verified-access-endpoint 명령을 사용합니다.

Verified·Access에 대한 네트워크 인터페이스 엔드포인트 생성

다음 절차에 따라 네트워크 인터페이스 엔드포인트를 생성하십시오.

요구 사항

- IPv4 트래픽만 지원됩니다.
- 네트워크 인터페이스는 보안 그룹과 동일한 Virtual Private Cloud(VPC)에 속해야 합니다.
- 네트워크 인터페이스의 프라이빗 IP를 사용하여 트래픽을 전달합니다.
- Verified·Access 엔드포인트를 생성하기 전에 Verified·Access 그룹을 생성해야 합니다. 자세한 내용은 the section called “Verified·Access 그룹 생성” 단원을 참조하십시오.
- 애플리케이션에 사용할 도메인 이름을 제공해야 합니다. 이 필드는 사용자가 애플리케이션에 액세스하는 데 사용하는 공용 DNS 이름입니다. 또한 이 도메인 이름과 일치하는 CN이 포함된 공용 SSL 인증서를 제공해야 합니다. 를 사용하여 인증서를 생성하거나 가져올 수 있습니다 AWS Certificate Manager.

콘솔을 사용하여 네트워크 인터페이스 엔드포인트를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. Verified·Access 엔드포인트 생성을 선택합니다.
4. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
5. Verified·Access 그룹에서 Verified·Access 그룹을 선택합니다.
6. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 프로토콜에서 프로토콜을 선택합니다.
   b. 연결 유형(Attachment type)에서 VPC를 선택합니다.
   c. Endpoint type(엔드포인트 유형)에서 Network interface(네트워크 인터페이스)를 선택합니다.
   d. (HTTP/HTTPS) 포트에 포트 번호를 입력합니다. (TCP) 포트 범위에 포트 범위를 입력하고 포트 추가를 선택합니다.
   e. 네트워크 인터페이스에서 네트워크 인터페이스를 선택합니다.
   f. 보안 그룹(Security groups)에서 엔드포인트의 보안 그룹을 선택합니다. 이러한 보안 그룹은 Verified Access 엔드포인트의 인바운드 및 아웃바운드 트래픽을 제어합니다.
   g. 엔드포인트 도메인 접두사의 경우 Verified Access가 엔드포인트에 대해 생성하는 DNS 이름 앞에 사용자 지정 식별자를 입력합니다.

7. (HTTP/HTTPS) 애플리케이션 세부 정보에서 다음을 수행합니다.
   a. 애플리케이션 도메인에 애플리케이션의 DNS 이름을 입력합니다.
   b. 도메인 인증서 ARN에서 퍼블릭 TLS 인증서를 선택합니다.
8. (선택 사항) 정책 정의에는 엔드포인트에 대한 Verified·Access 정책을 입력합니다.
9. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
10. Verified·Access 엔드포인트 생성을 선택합니다.

를 사용하여 Verified Access 엔드포인트를 생성하려면 AWS CLI

create-verified-access-endpoint 명령을 사용합니다.

Verified·Access를 위한 네트워크 CIDR 엔드포인트 생성

다음 절차에 따라 네트워크 CIDR 엔드포인트를 생성합니다. 예를 들어 네트워크 CIDR 엔드포인트를 사용하여 포트 22(SSH)를 통해 특정 서브넷의 EC2 인스턴스에 대한 액세스를 활성화할 수 있습니다.

요구 사항

- TCP 프로토콜만 지원됩니다.
- Verified·Access는 리소스에서 사용하는 CIDR 범위의 각 IP 주소에 대한 DNS 레코드를 제공합니다. 리소스를 삭제하면 해당 IP 주소가 더 이상 사용되지 않고 Verified Access가 해당 DNS 레코드를 삭제합니다.
- 사용자 지정 하위 도메인을 지정하는 경우 Verified·Access는 하위 도메인에서 사용 중인 각 IP 주소에 대한 DNS 레코드를 제공하고 DNS 서버의 IP 주소를 제공합니다. Verified Access DNS 서버를 가리키도록 하위 도메인에 대한 전달 규칙을 구성할 수 있습니다. 도메인의 레코드에 대한 모든 요청은 Verified Access DNS 서버에서 요청된 리소스의 IP 주소로 확인됩니다.
- Verified·Access 엔드포인트를 생성하기 전에 Verified·Access 그룹을 생성해야 합니다. 자세한 내용은 the section called “Verified·Access 그룹 생성” 단원을 참조하십시오.
- 엔드포인트를 생성한 다음를 사용하여 애플리케이션에 연결합니다연결 클라이언트.

콘솔을 사용하여 네트워크 CIDR 엔드포인트를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.

3. Verified·Access 엔드포인트 생성을 선택합니다.
4. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
5. Verified·Access 그룹에서 엔드포인트의 Verified·Access 그룹을 선택합니다.
6. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 프로토콜에서 TCP를 선택합니다.
   b. 연결 유형(Attachment type)에서 VPC를 선택합니다.
   c. 엔드포인트 유형에서 네트워크 CIDR을 선택합니다.
   d. 포트 범위에 포트 범위를 입력하고 포트 추가를 선택합니다.
   e. 서브넷에서 서브넷을 선택합니다.
   f. 보안 그룹(Security groups)에서 엔드포인트의 보안 그룹을 선택합니다. 이러한 보안 그룹은 Verified Access 엔드포인트의 인바운드 및 아웃바운드 트래픽을 제어합니다.
   g. (선택 사항) 엔드포인트 도메인 접두사에 Verified Access가 엔드포인트에 대해 생성하는 DNS 이름 앞에 추가할 사용자 지정 식별자를 입력합니다.
7. (선택 사항) 정책 정의에는 엔드포인트에 대한 Verified·Access 정책을 입력합니다.
8. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
9. Verified·Access 엔드포인트 생성을 선택합니다.

를 사용하여 Verified Access 엔드포인트를 생성하려면 AWS CLI

create-verified-access-endpoint 명령을 사용합니다.

Verified·Access를 위한 Amazon Relational Database Service 엔드포인트 생성

다음 절차에 따라 Amazon Relational Database Service(RDS) 엔드포인트를 생성합니다.

요구 사항

- TCP 프로토콜만 지원됩니다.
- RDS 인스턴스, RDS 클러스터 또는 RDS DB 프록시를 생성합니다.
- Verified·Access 엔드포인트를 생성하기 전에 Verified·Access 그룹을 생성해야 합니다. 자세한 내용은 the section called “Verified·Access 그룹 생성” 단원을 참조하십시오.

- 엔드포인트를 생성한 다음를 사용하여 애플리케이션에 연결합니다연결 클라이언트.

콘솔을 사용하여 Amazon Relational Database Service 엔드포인트를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. Verified·Access 엔드포인트 생성을 선택합니다.
4. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
5. Verified·Access 그룹에서 엔드포인트의 Verified·Access 그룹을 선택합니다.
6. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 프로토콜에서 TCP를 선택합니다.
   b. 연결 유형(Attachment type)에서 VPC를 선택합니다.
   c. 엔드포인트 유형에서 Amazon Relational Database Service(RDS)를 선택합니다.
   d. RDS 대상 유형의 경우 다음 중 하나를 수행합니다.
      - RDS 인스턴스를 선택한 다음 RDS 인스턴스에서 RDS 인스턴스를 선택합니다.
      - RDS 클러스터를 선택한 다음 RDS 클러스터에서 RDS 클러스터를 선택합니다.
      - RDS DB 프록시를 선택한 다음 RDS DB 프록시에서 RDS DB 프록시를 선택합니다.
   e. RDS 엔드포인트에서 이전 단계에서 선택한 RDS 리소스와 관련된 RDS 엔드포인트를 선택합니다.
   f. 포트에 포트 번호를 입력합니다.
   g. 서브넷에서 서브넷을 선택합니다. 가용 영역당 1개의 서브넷만 지정할 수 있습니다.
   h. 보안 그룹(Security groups)에서 엔드포인트의 보안 그룹을 선택합니다. 이러한 보안 그룹은 Verified Access 엔드포인트의 인바운드 및 아웃바운드 트래픽을 제어합니다.
   i. (선택 사항) 엔드포인트 도메인 접두사에 Verified Access가 엔드포인트에 대해 생성하는 DNS 이름 앞에 추가할 사용자 지정 식별자를 입력합니다.
7. (선택 사항) 정책 정의에는 엔드포인트에 대한 Verified·Access 정책을 입력합니다.
8. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
9. Verified·Access 엔드포인트 생성을 선택합니다.

를 사용하여 Verified Access 엔드포인트를 생성하려면 AWS CLI

create-verified-access-endpoint 명령을 사용합니다.

Verified·Access 엔드포인트에서 발생하는 트래픽 허용

Verified Access 엔드포인트에서 발생하는 트래픽을 허용하도록 애플리케이션의 보안 그룹을 구성할 수 있습니다. 엔드포인트의 보안 그룹을 소스로 지정하는 인바운드 규칙을 추가하면 됩니다. 애플리케이션이 Verified Access 엔드포인트로부터만 트래픽을 수신하도록 추가 인바운드 규칙을 제거하는 것이 좋습니다.

기존 아웃바운드 규칙을 유지하는 것이 좋습니다.

콘솔을 사용하여 애플리케이션의 보안 그룹 규칙을 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. Verified·Access 엔드포인트를 선택하고, 세부 정보 탭에서 보안 그룹 ID를 찾은 다음, 엔드포인트의 보안 그룹 ID를 복사합니다.
4. 탐색 창에서 Security groups를 선택합니다.
5. 대상과 관련된 보안 그룹에 대한 상자를 선택한 다음 작업, 인바운드 규칙 편집을 선택합니다.
6. Verified·Access 엔드포인트에서 발생하는 트래픽을 허용하는 보안 그룹 규칙을 추가하려면 다음을 수행하십시오.
   a. 규칙 추가를 선택합니다.
   b. 유형에서 모든 트래픽 또는 허용할 특정 트래픽을 선택합니다.
   c. 소스로 사용자 지정을 선택하고 엔드포인트에 보안 그룹의 ID를 붙여넣습니다.
7. (선택 사항) 트래픽이 Verified Access 엔드포인트에서만 발생하도록 하려면 다른 인바운드 보안 그룹 규칙을 삭제합니다.
8. 규칙 저장을 선택합니다.

를 사용하여 애플리케이션의 보안 그룹 규칙을 업데이트하려면 AWS CLI

describe-verified-access-endpoints 명령을 사용하여 보안 그룹의 ID를 가져온 다음 authorize-security-group-ingress 명령을 사용하여 인바운드 규칙을 추가합니다.

Verified·Access 엔드포인트 수정

다음 절차에 따라 Verified Access 엔드포인트를 수정하세요.

콘솔을 사용하여 Verified Access 엔드포인트를 수정하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. 엔드포인트를 선택합니다.
4. 작업, Verified·Access 엔드포인트 수정을 선택합니다.
5. 필요에 따라 엔드포인트 세부 사항을 수정합니다.
6. Verified·Access 엔드포인트 수정을 선택합니다.

를 사용하여 Verified Access 엔드포인트를 수정하려면 AWS CLI

modify-verified-access-endpoint 명령을 사용합니다.

Verified·Access 엔드포인트 정책 수정

다음 절차에 따라 Verified Access 엔드포인트의 정책을 수정하세요. 변경한 후에는 변경 사항이 적용되기까지 몇 분 정도 걸립니다.

콘솔을 사용하여 Verified Access 엔드포인트 정책을 수정하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. 엔드포인트를 선택합니다.
4. 작업, Verified·Access 엔드포인트 정책 수정을 선택합니다.
5. (선택 사항) 필요에 따라 정책 활성화를 켜거나 끕니다.
6. (선택 사항) 정책에는 엔드포인트에 적용할 Verified Access 정책을 입력합니다.
7. Verified·Access 엔드포인트 정책 수정을 선택합니다.

를 사용하여 Verified Access 엔드포인트 정책을 수정하려면 AWS CLI

modify-verified-access-endpoint-policy 명령을 사용합니다.

Verified·Access 엔드포인트 삭제

Verified·Access 엔드포인트 사용을 마치면 엔드포인트를 삭제할 수 있습니다.

콘솔을 사용하여 Verified Access 엔드포인트를 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. 엔드포인트를 선택합니다.
4. 작업, Verified·Access 엔드포인트 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **`delete`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 Verified Access 엔드포인트를 삭제하려면 AWS CLI

delete-verified-access-endpoint 명령을 사용합니다.

신뢰 공급자에서 Verified Access로 전송되는 신뢰 데이터

신뢰 데이터는 신뢰 공급자 AWS Verified Access 에서 로 전송되는 데이터입니다. 신뢰 데이터는 '사용자 클레임' 또는 '신뢰 컨텍스트'라고도 합니다. 데이터에는 일반적으로 사용자 또는 디바이스에 대한 정보가 포함됩니다. 신뢰 데이터의 예로는 사용자 이메일, 그룹 멤버십, 디바이스 운영 체제 버전, 디바이스 보안 상태 등이 있습니다. 전송되는 정보는 신뢰 공급자에 따라 다르므로 전체 및 업데이트된 신뢰 데이터 목록은 신뢰 공급자의 설명서를 참조해야 합니다.

그러나 Verified·Access 로깅 기능을 사용하면 신뢰 공급자로부터 어떤 신뢰 데이터가 전송되고 있는지도 확인할 수 있습니다. 이는 애플리케이션에 대한 액세스를 허용하거나 거부하는 정책을 정의할 때 유용할 수 있습니다. 로그에 신뢰 컨텍스트를 포함하는 방법에 대한 정보는 Verified Access 신뢰 컨텍스트 활성화 또는 비활성화를 참조하십시오.

이 섹션에는 정책 작성을 시작하는 데 도움이 되는 샘플 신뢰 데이터와 예제가 포함되어 있습니다. 여기에 제공된 정보는 설명을 위한 용도로만 사용되며 공식적인 참고 자료는 아닙니다.

내용

- Verified Access 신뢰 데이터의 기본 컨텍스트
- AWS IAM Identity Center Verified·Access 신뢰 데이터에 대한 컨텍스트
- Verified Access 신뢰 데이터의 서드 파티 신뢰 공급자 컨텍스트
- Verified Access의 사용자 클레임 전달 및 서명 확인

Verified Access 신뢰 데이터의 기본 컨텍스트

AWS Verified Access 에는 구성된 신뢰 공급자에 관계없이 모든 Cedar 평가에서 기본적으로 현재 요청에 대한 일부 요소가 포함됩니다. 원하는 경우 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다.

다음은 평가에 포함된 데이터의 예입니다.

예시

- HTTP 요청
- TCP 흐름

HTTP 요청

정책이 평가되면 Verified Access는 `context.http_request` 키 아래의 Cedar 컨텍스트에 현재 HTTP 요청에 대한 데이터를 포함합니다.

```
{
    "title": "HTTP Request data included by Verified Access",
    "type": "object",
    "properties": {
        "http_method": {
            "type": "string",
            "description": "The HTTP method",
            "example": "GET"
        },
        "hostname": {
            "type": "string",
            "description": "The host subcomponent of the authority component of the
 URI",
            "example": "example.com"
        },
        "path": {
            "type": "string",
            "description": "The path component of the URI",
            "example": "app/images"
        },
        "query": {
            "type": "string",
            "description": "The query component of the URI",
            "example": "value1=1&value2=2"
        },
        "x_forwarded_for": {
            "type": "string",
            "description": "The value of the X-Forwarded-For request header",
            "example": "17.7.7.1"
        },
        "port": {
           "type": "integer",
           "description": "The endpoint port",
           "example": 443
        },
        "user_agent": {
            "type": "string",
            "description": "The value of the User-Agent request header",
```

```
            "example": "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0)
 Gecko/20100101 Firefox/47.0"
        },
        "client_ip": {
            "type": "string",
            "description": "The IP address connecting to the endpoint",
            "example": "15.248.6.6"
        }
    }
}
```

정책 예제

다음은 HTTP 요청 데이터를 사용하는 Cedar 정책의 예입니다.

```
forbid(principal, action, resource) when {
   context.http_request.http_method == "POST"
   && !(context.identity.roles.contains("Administrator"))
 };
```

TCP 흐름

정책이 평가되면 Verified·Access는 `context.tcp_flow` 키 아래의 Cedar 컨텍스트에 현재 TCP 흐름에 대한 데이터를 포함합니다.

```
{
    "title": "TCP flow data included by Verified Access",
    "type": "object",
    "properties": {
        "destination_ip": {
            "type": "string",
            "description": "The IP address of the target",
            "example": "192.100.1.3"
        },
        "destination_port": {
            "type": "string",
            "description": "The target port",
            "example": 22
        },
        "client_ip": {
            "type": "string",
            "description": "The IP address connecting to the endpoint",
            "example": "172.154.16.9"
```

```
        }
    }
}
```

AWS IAM Identity Center Verified·Access 신뢰 데이터에 대한 컨텍스트

정책을 평가할 때를 신뢰 공급자 AWS IAM Identity Center 로 정의하면 AWS Verified Access 는 신뢰 공급자 구성에서 “정책 참조 이름”으로 지정한 키 아래에 Cedar 컨텍스트의 신뢰 데이터를 포함합니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다.

> **Note**
>
> 신뢰 공급자의 컨텍스트 키는 신뢰 공급자를 생성할 때 구성한 정책 참조 이름에서 가져옵니다. 예를 들어, 정책 참조 이름을 “idp123"으로 구성하면 컨텍스트 키는 “context.idp123"이 됩니다. 정책을 생성할 때 올바른 컨텍스트 키를 사용하고 있는지 확인합니다.

다음 JSON 스키마는 평가에 포함되는 데이터를 보여줍니다.

```
{
   "title": "AWS IAM Identity Center context specification",
   "type": "object",
   "properties": {
     "user": {
       "type": "object",
       "properties": {
         "user_id": {
           "type": "string",
           "description": "a unique user id generated by AWS IdC"
         },
         "user_name": {
           "type": "string",
           "description": "username provided in the directory"
         },
         "email": {
           "type": "object",
           "properties": {
             "address": {
               "type": "email",
```

```
                "description": "email address associated with the user"
              },
              "verified": {
                "type": "boolean",
                "description": "whether the email address has been verified by AWS IdC"
              }
            }
          }
        }
      },
      "groups": {
        "type": "object",
        "description": "A list of groups the user is a member of",
        "patternProperties": {
          "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{12}$": {
            "type": "object",
            "description": "The Group ID of the group",
            "properties": {
              "group_name": {
                "type": "string",
                "description": "The customer-provided name of the group"
              }
            }
          }
        }
      }
    }
  }
```

다음은 AWS IAM Identity Center에서 제공하는 신뢰 데이터를 기준으로 평가하는 정책의 예입니다.

```
permit(principal, action, resource) when {
   context.idc.user.email.verified == true
   // User is in the "sales" group with specific ID
   && context.idc.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
 };
```

> **Note**
>
> 그룹 이름은 변경될 수 있으므로 IAM Identity Center는 그룹 ID를 사용하여 그룹을 참조합니다. 이렇게 하면 그룹 이름을 변경할 때 정책 설명이 위반되는 것을 방지할 수 있습니다.

Verified Access 신뢰 데이터의 서드 파티 신뢰 공급자 컨텍스트

이 섹션에서는 타사 신뢰 공급자가 AWS Verified Access 에 제공하는 신뢰 데이터를 설명합니다.

> **Note**
>
> 신뢰 공급자의 컨텍스트 키는 신뢰 공급자를 생성할 때 구성한 정책 참조 이름에서 가져옵니다. 예를 들어, 정책 참조 이름을 “idp123"으로 구성하면 컨텍스트 키는 “context.idp123"이 됩니다. 정책을 생성할 때 올바른 컨텍스트 키를 사용하고 있는지 확인합니다.

내용

- 브라우저 확장
- Jamf
- CrowdStrike
- JumpCloud

브라우저 확장

디바이스 신뢰 컨텍스트를 액세스 정책에 통합하려는 경우 AWS Verified Access 브라우저 확장 프로그램 또는 다른 파트너의 브라우저 확장 프로그램이 필요합니다. Verified Access는 현재 Google Chrome과 Mozilla Firefox 브라우저를 지원합니다.

우리는 현재 Jamf(macOS 장치 지원), CrowdStrike(Windows 11 및 Windows 10 장치 지원), JumpCloud(Windows 및 MacOS 모두 지원)의 세 가지 디바이스 신뢰 공급자를 지원합니다.

- 정책에서 Jamf 신뢰 데이터를 사용하는 경우 사용자는 Chrome 웹 스토어 또는 Firefox 추가 기능 사이트에서 AWS Verified Access 브라우저 확장 프로그램을 다운로드하여 디바이스에 설치해야 합니다.
- 정책에 CrowdStrike 신뢰 데이터를 사용하는 경우 먼저 사용자가 AWS Verified Access 네이티브 메시징 호스트(직접 다운로드 링크)를 설치해야 합니다. 이 구성 요소는 사용자 디바이스에서 실행되는 CrowdStrike 에이전트로부터 신뢰 데이터를 가져오는 데 필요합니다. 그런 다음이 구성 요소를 설치한 후 사용자는 Chrome 웹 스토어 또는 Firefox 추가 기능 사이트에서 디바이스에 AWS Verified Access 브라우저 확장 프로그램을 설치해야 합니다.
- JumpCloud를 사용하는 경우 사용자는 기기에 Chrome 웹 스토어 또는 Firefox 추가 기능 사이트의 JumpCloud 브라우저 확장 프로그램을 설치해야 합니다.

Jamf

Jamf는 타사 신뢰 공급자입니다. 정책을 평가할 때 Jamf를 신뢰 공급자로 정의하면 Verified·Access는 신뢰 공급자 구성에서 “정책 참조 이름(Policy Reference Name)”으로 지정한 키 아래에 Cedar 컨텍스트의 신뢰 데이터를 포함시킵니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다. 다음 JSON 스키마는 평가에 포함되는 데이터를 보여줍니다.

Verified Access와 Jamf를 사용하는 방법에 대한 자세한 내용은 Jamf 웹사이트의 AWS Verified Access를 Jamf 디바이스 자격 증명과 통합 섹션을 참조하세요.

```
{
    "title": "Jamf device data specification",
    "type": "object",
    "properties": {
        "iss": {
            "type": "string",
            "description": "\"Issuer\" - the Jamf customer ID"
        },
        "iat": {
            "type": "integer",
            "description": "\"Issued at Time\" - a unixtime (seconds since epoch) value
 of when the device information data was generated"
        },
        "exp": {
            "type": "integer",
            "description": "\"Expiration\" - a unixtime (seconds since epoch) value for
 when this device information is no longer valid"
        },
        "sub": {
            "type": "string",
            "description": "\"Subject\" - either the hardware UID or a value generated
 based on device location"
        },
        "groups": {
            "type": "array",
            "description": "Group IDs from UEM connector sync",
            "items": {
                "type": "string"
            }
        },
        "risk": {
            "type": "string",
            "enum": [
```

```
                "HIGH",
                "MEDIUM",
                "LOW",
                "SECURE",
                "NOT_APPLICABLE"
            ],
            "description": "a Jamf-reported level of risk associated with the device."
        },
        "osv": {
            "type": "string",
            "description": "The version of the OS that is currently running, in Apple
 version number format (https://support.apple.com/en-us/HT201260)"
        }
    }
}
```

다음은 Jamf에서 제공하는 신뢰 데이터를 기준으로 평가하는 정책의 예입니다.

```
permit(principal, action, resource) when {
   context.jamf.risk == "LOW"
};
```

Cedar는 Jamf의 위험 점수와 같은 Enum을 처리하는 데 유용한 `.contains()` 함수를 제공합니다.

```
permit(principal, action, resource) when {
   ["LOW", "SECURE"].contains(context.jamf.risk)
};
```

CrowdStrike

CrowdStrike는 타사 신뢰 공급자입니다. 정책을 평가할 때 CrowdStrike를 신뢰 공급자로 정의하면 Verified·Access는 신뢰 공급자 구성에서 “정책 참조 이름(Policy Reference Name)”으로 지정한 키 아래에 Cedar 컨텍스트의 신뢰 데이터를 포함합니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다. 다음 JSON 스키마는 평가에 포함되는 데이터를 보여줍니다.

Verified Access와 CrowdStrike를 사용하는 방법에 대한 자세한 내용은 GitHub 웹사이트의 CrowdStrike 및 AWS Verified Access를 통한 비공개 애플리케이션 보호를 참조하세요.

```
{
  "title": "CrowdStrike device data specification",
  "type": "object",
  "properties": {
```

```
    "assessment": {
      "type": "object",
      "description": "Data about CrowdStrike's assessment of the device",
      "properties": {
        "overall": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts as a weighted
 average of the OS and and Sensor Config scores"
        },
        "os": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the OS-
specific settings monitored on the host"
        },
        "sensor_config": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the
 different sensor policies monitored on the host"
        },
        "version": {
          "type": "string",
          "description": "The version of the scoring algorithm being used"
        }
      }
    },
    "cid": {
      "type": "string",
      "description": "Customer ID (CID) unique to the customer's environment"
    },
    "exp": {
      "type": "integer",
      "description": "unixtime, The expiration time of the token"
    },
    "iat": {
      "type": "integer",
      "description": "unixtime, The issued time of the token"
    },
    "jwk_url": {
      "type": "string",
      "description": "URL that details the JWT signing"
    },
    "platform": {
      "type": "string",
      "enum": ["Windows 10", "Windows 11", "macOS"],
```

```
      "description": "Operating system of the endpoint"
    },
    "serial_number": {
      "type": "string",
      "description": "The serial number of the device derived by unique system
 information"
    },
    "sub": {
      "type": "string",
      "description": "Unique CrowdStrike Agent ID (AID) of machine"
    },
    "typ": {
      "type": "string",
      "enum": ["crowdstrike-zta+jwt"],
      "description": "Generic name for this JWT media. Client MUST reject any other
 type"
    }
  }
}
```

다음은 CrowdStrike에서 제공하는 신뢰 데이터를 기준으로 평가하는 정책의 예입니다.

```
permit(principal, action, resource) when {
   context.crowdstrike.assessment.overall > 50
};
```

JumpCloud

JumpCloud는 타사 신뢰 공급자입니다. 정책을 평가할 때 JumpCloud를 신뢰 공급자로 정의하면 Verified Access는 사용자가 신뢰 공급자 구성에서 “정책 참조 이름(Policy Reference Name)”으로 지정한 키 아래에 Cedar 컨텍스트의 신뢰 데이터를 포함합니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다. 다음 JSON 스키마는 평가에 포함되는 데이터를 보여줍니다.

JumpCloud를 AWS Verified·Access와 함께 사용하는 방법에 대한 자세한 내용은 JumpCloud 웹 사이트의 JumpCloud 및 AWS Verified·Access 통합을 참조하세요. JumpCloud

```
{
  "title": "JumpCloud device data specification",
  "type": "object",
  "properties": {
    "device": {
      "type": "object",
```

```
      "description": "Properties of the device",
      "properties": {
        "is_managed": {
          "type": "boolean",
          "description": "Boolean to indicate if the device is under management"
        }
      }
    },
    "exp": {
      "type": "integer",
      "description": "Expiration. Unixtime of the token's expiration."
    },
    "durt_id": {
      "type": "string",
      "description": "Device User Refresh Token ID. Unique ID that represents the
 device + user."
    },
    "iat": {
      "type": "integer",
      "description": "Issued At. Unixtime of the token's issuance."
    },
    "iss": {
      "type": "string",
      "description": "Issuer. This will be 'go.jumpcloud.com'"
    },
    "org_id": {
      "type": "string",
      "description": "The JumpCloud Organization ID"
    },
    "sub": {
      "type": "string",
      "description": "Subject. The managed JumpCloud user ID on the device."
    },
    "system": {
      "type": "string",
      "description": "The JumpCloud system ID"
    }
  }
}
```

다음은 JumpCloud에서 제공하는 신뢰 컨텍스트를 기준으로 평가하는 정책의 예입니다.

```
permit(principal, action, resource) when {
```

```
   context.jumpcloud.org_id == 'Unique_organization_identifier'
};
```

Verified Access의 사용자 클레임 전달 및 서명 확인

AWS Verified Access 인스턴스가 사용자를 성공적으로 인증하면 IdP에서 수신한 사용자 클레임을 Verified Access 엔드포인트로 보냅니다. 사용자 클레임은 서명되어 애플리케이션에서 서명을 확인하고 클레임이 Verified Access에서 전송되었음을 확인할 수 있습니다. 이 프로세스 중에 다음 HTTP 헤더가 추가됩니다.

`x-amzn-ava-user-context`

이 헤더는 JSON 웹 토큰(JWT) 형식의 사용자 클레임을 포함합니다. JWT 형식에는 base64 URL 방식으로 인코딩된 헤더, 페이로드 및 서명이 포함됩니다. Verified·Access는 ES384(SHA-384 해시 알고리즘을 사용하는 ECDSA 서명 알고리즘)를 사용하여 JWT 서명을 생성합니다.

애플리케이션은 이러한 클레임을 개인화 또는 기타 사용자별 경험에 사용할 수 있습니다. 애플리케이션 개발자는 사용하기 전에 자격 증명 공급자가 제공하는 각 클레임의 고유성 및 검증 수준에 대해 스스로 학습해야 합니다. 일반적으로 `sub` 클레임은 특정 사용자를 식별하는 가장 좋은 방법입니다.

내용

- 예: OIDC 사용자 클레임용으로 서명된 JWT
- 예: IAM Identity Center 사용자 클레임용으로 서명된 JWT
- 퍼블릭 키
- 예: JWT 검색 및 디코딩

예: OIDC 사용자 클레임용으로 서명된 JWT

다음 예에서는 OIDC 사용자 클레임의 헤더와 페이로드가 JWT 형식에서 어떻게 보일지 보여줍니다.

헤더의 예:

```
{
   "alg": "ES384",
   "kid": "12345678-1234-1234-1234-123456789012",
   "signer": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/vai-
abc123xzy321a2b3c",
   "iss": "OIDC Issuer URL",
```

```
   "exp": "expiration" (120 secs)
}
```

페이로드의 예:

```
{
   "sub": "xyzsubject",
   "email": "xxx@amazon.com",
   "email_verified": true,
   "groups": [
      "Engineering",
      "finance"
   ],
   "additional_user_context": {
      "aud": "xxx",
      "exp": 1000000000,
      "groups": [
         "group-id-1",
         "group-id-2"
      ],
      "iat": 1000000000,
      "iss": "https://oidc-tp.com/",
      "sub": "xyzsubject",
      "ver": "1.0"
   }
}
```

예: IAM Identity Center 사용자 클레임용으로 서명된 JWT

다음 예에서는 IAM Identity Center 사용자 클레임의 헤더와 페이로드가 JWT 형식에서 어떻게 보일지 보여줍니다.

> **Note**
>
> IAM Identity Center의 경우 사용자 정보만 클레임에 포함됩니다.

헤더의 예:

```
{
   "alg": "ES384",
```

```
   "kid": "12345678-1234-1234-1234-123456789012",
   "signer": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/vai-
abc123xzy321a2b3c",
   "iss": "arn:aws:ec2:us-east-1:123456789012:verified-access-trust-provider/vatp-
abc123xzy321a2b3c",
   "exp": "expiration" (120 secs)
}
```

페이로드의 예:

```
{
    "user": {
        "user_id": "f478d4c8-a001-7064-6ea6-12423523",
        "user_name": "test-123",
        "email": {
            "address": "test@amazon.com",
            "verified": false
        }
    }
}
```

퍼블릭 키

Verified·Access 인스턴스는 사용자 클레임을 암호화하지 않기 때문에 HTTPS를 사용하도록 Verified·Access 엔드포인트를 구성하는 것이 좋습니다. HTTP를 사용하도록 Verified·Access 엔드포인트를 구성하는 경우 반드시 보안 그룹을 사용하여 트래픽을 엔드포인트로 제한해야 합니다.

보안을 보장하려면 클레임을 기반으로 권한을 부여하기 전에 서명을 확인하고 JWT 헤더의 `signer` 필드에 필요한 Verified Access 인스턴스 ARN이 포함되어 있는지 확인해야 합니다.

퍼블릭 키를 가져오려면 JWT 헤더에서 키 ID를 가져오고 이 정보를 사용하여 엔드포인트에서 퍼블릭 키를 조회합니다.

각의 엔드포인트 AWS 리전 는 다음과 같습니다.

`https://public-keys.prod.verified-access.<region>.amazonaws.com/<key-id>`

예: JWT 검색 및 디코딩

다음 코드 예에서는 Python 3.9로 키 ID, 퍼블릭 키 및 페이로드를 가져오는 방법을 보여줍니다.

```
import jwt
```

```
import requests
import base64
import json

# Step 1: Validate the signer
expected_verified_access_instance_arn = 'arn:aws:ec2:region-code:account-id:verified-
access-instance/verified-access-instance-id'

encoded_jwt = headers.dict['x-amzn-ava-user-context']
jwt_headers = encoded_jwt.split('.')[0]
decoded_jwt_headers = base64.b64decode(jwt_headers)
decoded_jwt_headers = decoded_jwt_headers.decode("utf-8")
decoded_json = json.loads(decoded_jwt_headers)
received_verified_access_instance_arn = decoded_json['signer']

assert expected_verified_access_instance_arn == received_verified_access_instance_arn,
 "Invalid Signer"

# Step 2: Get the key id from JWT headers (the kid field)
kid = decoded_json['kid']

# Step 3: Get the public key from regional endpoint
url = 'https://public-keys.prod.verified-access.' + region + '.amazonaws.com/' + kid
req = requests.get(url)
pub_key = req.text

# Step 4: Get the payload
payload = jwt.decode(encoded_jwt, pub_key, algorithms=['ES384'])
```

Verified·Access 정책

AWS Verified Access 정책을 사용하면에서 호스팅되는 애플리케이션에 액세스하기 위한 규칙을 정의할 수 있습니다 AWS. 정책 AWS 언어인 Cedar로 작성됩니다. Cedar를 사용하면 Verified·Access와 함께 사용하도록 구성한 ID 또는 디바이스 기반 신뢰 공급자로부터 전송된 신뢰 데이터를 기준으로 평가되는 정책을 생성할 수 있습니다.

Cedar 정책 언어에 대한 더 자세한 내용은 Cedar 참조 안내서를 참조하십시오.

Verified·Access 그룹을 생성하거나 Verified·Access 엔드포인트를 생성할 때 Verified·Access 정책을 정의할 수 있는 옵션이 있습니다. Verified Access 정책을 정의하지 않고 그룹 또는 엔드포인트를 생성할 수 있지만 정책을 정의할 때까지 모든 액세스 요청이 차단됩니다. 또는 기존 Verified Access 그룹 또는 엔드포인트를 생성한 후 정책을 추가하거나 변경할 수도 있습니다.

내용

- Verified Access 정책 문 구조
- Verified Access 정책의 내장 연산자
- Verified Access 정책 평가
- Verified Access 정책 논리 단락
- Verified Access 정책 예제
- Verified Access 정책 도우미

Verified Access 정책 문 구조

다음 표는 Verified Access 정책의 구조를 보여줍니다.

구성 요소	구문
effect	`permit | forbid`
scope	`(principal, action, resource)`
조건 절	`when { context.policy-reference-name .attribute-name };`

정책 구성 요소

Verified Access 정책에는 다음 구성 요소가 포함되어 있습니다.

- 효과 - 액세스를 `permit`(허용) 또는 `forbid`(거부)합니다.
- 범위 – 효과가 적용되는 위탁자, 작업 및 리소스입니다. 특정 위탁자, 작업 또는 리소스를 식별하지 않음으로써 Cedar의 범위를 정의하지 않은 상태로 둘 수 있습니다. 이 경우 가능한 모든 주체, 작업 및 리소스에 정책이 적용됩니다.
- 조건 절 - 효과가 적용되는 컨텍스트입니다.

> **Important**
>
> Verified Access의 경우 조건 조항의 신뢰 데이터를 참조하여 정책을 완전히 표현합니다. 정책 범위는 항상 정의되지 않은 상태로 유지해야 합니다. 그런 다음 조건 조항에서 ID 및 디바이스 신뢰 컨텍스트를 사용하여 액세스를 지정할 수 있습니다.

설명

AWS Verified Access 정책에 설명을 포함할 수 있습니다. 주석은 `//`로 시작하고 줄바꿈 문자로 끝나는 줄로 정의됩니다.

다음 예는 정책의 주석을 보여줍니다.

```
// grants access to users in a specific domain using trusted devices
permit(principal, action, resource)
when {
  // the user's email address is in the @example.com domain
  context.idc.user.email.address.contains("@example.com")
  // Jamf thinks the user's computer is low risk or secure.
  && ["LOW", "SECURE"].contains(context.jamf.risk)
};
```

여러 절

&& 연산자를 사용하여 정책 문에 조건 절을 두 개 이상 사용할 수 있습니다.

```
permit(principal,action,resource)
when{
```

```
 context.policy-reference-name.attribute1 &&
 context.policy-reference-name.attribute2
};
```

추가 예제는 다음(Verified Access 정책 예제)을 참조하십시오.

예약 문자

다음 예제는 컨텍스트 속성이 정책 언어의 예약 문자인 :(세미콜론)을 사용하는 경우 정책을 작성하는 방법을 보여줍니다.

```
permit(principal, action, resource)
when {
    context.policy-reference-name["namespace:groups"].contains("finance")
};
```

Verified Access 정책의 내장 연산자

에서 설명한 대로 다양한 조건을 사용하여 AWS Verified Access 정책의 컨텍스트를 생성할 때 && 연산자를 사용하여 추가 조건을 추가할 Verified Access 정책 문 구조수 있습니다. 정책 조건에 추가적인 표현력을 추가하는 데 사용할 수 있는 다른 내장 연산자도 많이 있습니다. 다음 표에는 참조용으로 제공되는 모든 내장 연산자가 나와 있습니다.

연산자	유형 및 오버로드	설명
!	Boolean → Boolean	논리 not.
==	any → any	대등. 유형이 일치하지 않는 경우에도 모든 유형의 값에서 작동합니다. 서로 다른 유형의 값은 결코 서로 같을 수 없습니다.
!=	any → any	부등, 대등의 정반대(위 참조).
<	(long, long) → Boolean	보다 작은 배장 정수.
<=	(long, long) → Boolean	작거나 같은 배장 정수.
>	(long, long) → Boolean	보다 큰 배장 정수.

연산자	유형 및 오버로드	설명
>=	(long, long) → Boolean	크거나 같은 배장 정수.
in	(entity, entity) → Boolean	계층 멤버십(재귀적: A의 A는 항상 참임).
	(entity, set(entity)) → Boolean	계층 멤버십: (A와 B) || (C의 A) ||이면 [B, C,...] 의 A는 참이며 ... 집합에 개체가 아닌 항목이 포함된 경우 오류입니다.
&&	(Boolean, Boolean) → Boolean	논리 및 (단락 평가).
||	(Boolean, Boolean) → Boolean	논리 또는 (단락 평가).
.exists()	entity → Boolean	엔터티 존재.
has	(entity, attribute) → Boolean	중위 연산자. `e has f`은(는) 레코드 또는 엔터티 `e`에 속성 `f`에 대한 바인딩이 있는지 테스트합니다. `e`가 존재하지 않는 경우 또는 `e`가 존재하지만 속성 `f`가 없는 경우 `false`를 반환합니다. 속성은 식별자 또는 문자열로 표현할 수 있습니다.
like	(string, string) → Boolean	중위 연산자. `t like p`은(는) 텍스트 `t`가 패턴 `p`과 일치하는지 확인합니다. 패턴에는 0개 이상의 문자와 일치하는 와일드카드 문자 `*`가 포함될 수 있습니다. `t`에서 문자 그대로의 별표 문자를 일치시키려면 `p`의 `\*`에서 특수 문자열을 사용할 수 있습니다.

연산자	유형 및 오버로드	설명
.contains()	(set, any) → Boolean	멤버십을 설정합니다(B는 A의 요소인지).
.containsAll()	(set, set) → Boolean	A 집합에 B 집합의 모든 요소가 포함되어 있는지 테스트합니다.
.containsAny()	(set, set) → Boolean	A 집합에 B 집합의 요소가 포함되어 있는지 테스트합니다.

Verified Access 정책 평가

정책 문서는 하나 이상의 정책 설명(`permit` 또는 `forbid` 설명)의 집합입니다. 정책은 조건부 조항(`when` 진술)이 참일 경우 적용됩니다. 정책 문서에 액세스를 허용하려면 문서에 있는 하나 이상의 허가 정책이 적용되어야 하며 금지 정책은 적용할 수 없습니다. 허가 정책이 적용되지 않고/않거나 하나 이상의 금지 정책이 적용되는 경우에는 정책 문서가 액세스를 거부합니다. Verified Access 그룹과 Verified Access 엔드포인트 모두에 대해 정책 문서를 정의한 경우 두 문서 모두 액세스를 허용해야 합니다. Verified Access 엔드포인트에 대한 정책 문서를 정의하지 않은 경우 Verified Access 그룹 정책에만 액세스가 필요합니다.

AWS Verified Access 는 정책을 생성할 때 구문을 검증하지만 조건부 절에 입력한 데이터는 검증하지 않습니다.

Verified Access 정책 논리 단락

주어진 컨텍스트에 존재할 수도 있고 존재하지 않을 수도 있는 데이터를 평가하는 AWS Verified Access 정책을 작성할 수도 있습니다. 존재하지 않는 컨텍스트의 데이터를 참조하는 경우 Cedar는 오류를 생성하고 사용자의 의도와 상관없이 정책을 평가하여 액세스를 거부합니다. 예를 들어, 이 컨텍스트에 `fake_provider`와(과) `bogus_key`이(가) 존재하지 않으므로 거부로 이어질 수 있습니다.

```
permit(principal, action, resource) when {
  context.fake_provider.bogus_key > 42
};
```

이러한 상황을 방지하려면 has 연산자를 사용하여 키가 있는지 확인할 수 있습니다. has 연산자가 거짓을 반환하면 연결된 문장에 대한 추가 평가가 중단되고 Cedar는 존재하지 않는 항목을 참조하려고 시도하면서 오류를 발생하지 않습니다.

```
permit(principal, action, resource) when {
  context.identity.user has "some_key" && context.identity.user.some_key > 42
};
```

이는 서로 다른 두 신뢰 공급자를 참조하는 정책을 지정할 때 가장 유용합니다.

```
permit(principal, action, resource) when {
  // user is in an allowed group
  context.aws_idc.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
  &&(
    (
      // if CrowdStrike data is present,
      // permit if CrowdStrike's overall assessment is over 50
      context has "crowdstrike" && context.crowdstrike.assessment.overall > 50
    )
    ||
    (
      // if Jamf data is present,
      // permit if Jamf's risk score is acceptable
      context has "jamf" && ["LOW", "NOT_APPLICABLE", "MEDIUM",
 "SECURE"].contains(context.jamf.risk)
    )
  )
};
```

Verified Access 정책 예제

Verified Access 정책을 사용하여 특정 사용자 및 디바이스에 애플리케이션에 대한 액세스 권한을 부여할 수 있습니다.

정책 예제

- 예제 1: IAM Identity Center에서 그룹에 액세스 권한 부여
- 예제 2: 서드 파티 공급자에서 그룹에 액세스 권한 부여
- 예제 3: CrowdStrike를 사용하여 액세스 권한 부여
- 예제 4: 특정 IP 주소 허용 또는 거부

예제 1: IAM Identity Center에서 그룹에 액세스 권한 부여

를 사용할 때는 IDs를 사용하여 그룹을 참조하는 AWS IAM Identity Center것이 좋습니다. 이렇게 하면 그룹 이름을 변경할 경우 정책 문의 실행이 종료되는 것을 방지할 수 있습니다.

다음 예제 정책은 확인된 이메일 주소가 있는 지정된 그룹의 사용자에게만 액세스를 허용합니다. 그룹 ID는 c242c5b0-6081-1845-6fa8-6e0d9513c107입니다.

```
permit(principal,action,resource)
when {
    context.policy-reference-name.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
    && context.policy-reference-name.user.email.verified == true
};
```

다음 예제 정책은 사용자가 지정된 그룹에 속하고 확인된 이메일 주소를 갖고 있으며 Jamf 디바이스 위험 점수가 LOW인 경우에만 액세스를 허용합니다.

```
permit(principal,action,resource)
when {
    context.policy-reference-name.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
    && context.policy-reference-name.user.email.verified == true
    && context.jamf.risk == "LOW"
};
```

신뢰 데이터에 대한 자세한 내용은 the section called “AWS IAM Identity Center 컨텍스트” 섹션을 참조하세요.

예제 2: 서드 파티 공급자에서 그룹에 액세스 권한 부여

다음 예제 정책은 사용자가 지정된 그룹에 속하고 확인된 이메일 주소를 갖고 있으며 Jamf 디바이스 위험 점수가 LOW인 경우에만 액세스를 허용합니다. 그룹의 이름은 “finance”입니다.

```
permit(principal,action,resource)
when {
     context.policy-reference-name.groups.contains("finance")
     && context.policy-reference-name.email_verified == true
     && context.jamf.risk == "LOW"
};
```

신뢰 데이터에 대한 자세한 내용은 the section called “서드 파티 컨텍스트” 섹션을 참조하세요.

예제 3: CrowdStrike를 사용하여 액세스 권한 부여

다음 예제 정책은 전체 평가 점수가 50점을 넘을 때 액세스를 허용합니다.

```
permit(principal,action,resource)
when {
    context.crwd.assessment.overall > 50
};
```

예제 4: 특정 IP 주소 허용 또는 거부

다음 예제 정책은 지정된 IP 주소의 요청만 허용합니다.

```
permit(principal, action, resource)
when {
    context.http_request.client_ip == "192.0.2.1"
};
```

다음 예제 정책은 지정된 IP 주소의 요청을 거부합니다.

```
forbid(principal,action,resource)
when {
    ip(context.http_request.client_ip).isInRange(ip("192.0.2.1/32"))
};
```

Verified Access 정책 도우미

Verified Access 정책 도우미는 정책을 테스트하고 개발하는 데 사용할 수 있는 검증된 액세스 콘솔의 도구입니다. 엔드포인트 정책, 그룹 정책 및 신뢰 컨텍스트를 한 화면에 표시하여 정책을 테스트하고 편집할 수 있습니다.

신뢰 컨텍스트 형식은 신뢰 공급자마다 다르며 Verified Access 관리자는 특정 신뢰 제공자가 사용하는 정확한 형식을 모를 수도 있습니다. 따라서 테스트 및 개발 목적으로 한 곳에서 신뢰 컨텍스트와 그룹 및 엔드포인트 정책을 모두 확인하는 것이 매우 유용할 수 있습니다.

다음 섹션에서는 환경의 기능에 대해 설명합니다.

업무

- 1단계: 리소스 지정
- 2단계: 정책 테스트 및 편집
- 3단계: 변경 사항 검토 및 적용

1단계: 리소스 지정

다음 섹션에서는, 사용하려는 AMI를 선택합니다. 또한 사용자(이메일 주소로 식별)를 지정하고 선택적으로 사용자 이름 및/또는 장치 식별자를 지정합니다. 기본적으로 가장 최근의 승인 결정은 지정된 사용자의 Verified Access 로그에서 추출됩니다. 선택적으로 가장 최근의 허용 또는 거부 결정을 구체적으로 선택할 수 있습니다.

마지막으로 신뢰 컨텍스트, 권한 부여 결정, 엔드포인트 정책 및 그룹 정책이 모두 다음 화면에 표시됩니다.

정책 도우미를 열고 리소스를 지정하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified Access 인스턴스를 선택한 다음 사용할 인스턴스의 Verified Access 인스턴스 ID를 클릭합니다.
3. 정책 지원 시작을 선택합니다.
4. 사용자 이메일 주소에 사용자의 이메일 주소를 입력합니다.
5. Verified Access 엔드포인트에서 정책을 편집하고 테스트하려는 엔드포인트를 선택합니다.
6. (선택 사항) 이름에는 사용자 이름을 입력합니다.
7. (선택 사항) 장치 식별자에 고유한 장치 식별자를 입력합니다.
8. (선택 사항) 인증 결과에서 사용하려는 최근 인증 결과의 유형을 선택합니다. 기본적으로 최신 권한 부여 결과가 사용됩니다.
9. Next(다음)를 선택합니다.

2단계: 정책 테스트 및 편집

이 페이지에는 작업할 때 사용할 수 있는 다음과 같은 정보가 표시됩니다.

- 신뢰 제공자가 사용자에게 보낸 신뢰 컨텍스트 및 (선택적으로) 이전 단계에서 지정한 장치입니다.
- 이전 단계에서 지정된 Verified Access 엔드포인트에 대한 Cedar 정책입니다.
- 엔드포인트가 속한 Verified Access 그룹에 대한 Cedar 정책입니다.

Verified Access 엔드포인트 및 그룹에 대한 Cedar 정책은 이 페이지에서 편집할 수 있지만 신뢰 컨텍스트는 정적입니다. 이제 이 페이지에서 Cedar 정책과 함께 신뢰 컨텍스트를 볼 수 있습니다.

정책 테스트 버튼을 선택하여 신뢰 컨텍스트와 비교하여 정책을 테스트하면 권한 부여 결과가 화면에 표시됩니다. 정책을 편집하고 변경 사항을 다시 테스트하여 필요에 따라 프로세스를 반복할 수 있습니다.

정책 변경에 만족하면 다음을 선택하여 정책 지원의 다음 화면으로 계속 진행합니다.

3단계: 변경 사항 검토 및 적용

정책 도우미의 마지막 페이지에서 쉽게 검토할 수 있도록 강조 표시된 정책 변경 내용을 확인할 수 있습니다. 이제 마지막으로 검토하고 변경 사항 적용을 선택하여 변경 사항을 적용할 수 있습니다.

이전을 선택하여 이전 페이지로 돌아가거나 취소를 선택하여 정책 지원을 완전히 취소할 수도 있습니다.

용 연결 클라이언트 AWS Verified Access

AWS Verified Access 는 사용자 디바이스와 비 HTTP 애플리케이션 간의 연결을 활성화할 수 있도록 연결 클라이언트를 제공합니다. 클라이언트는 사용자 트래픽을 안전하게 암호화하고, 사용자 자격 증명 정보와 디바이스 컨텍스트를 추가하고, 정책 적용을 위해 Verified Access로 라우팅합니다. 액세스 정책이 액세스를 허용하는 경우 사용자는 애플리케이션에 연결됩니다. Connectivity Client가 연결되어 있는 한 사용자 액세스 권한이 지속적으로 부여됩니다.

클라이언트는 시스템 서비스로 실행되며 충돌에 대한 복원력이 뛰어납니다. 연결이 불안정해지면 클라이언트가 연결을 다시 설정합니다.

클라이언트는 임시 OAuth 액세스 토큰을 사용하여 보안 터널을 설정합니다. 사용자가 클라이언트에서 로그아웃하면 터널 연결이 해제됩니다.

액세스 및 새로 고침 토큰은 사용자 디바이스의 암호화된 SQLite 데이터베이스에 로컬로 저장됩니다.

내용

- 사전 조건
- 연결 클라이언트 다운로드
- 클라이언트 구성 파일 내보내기
- 애플리케이션에 연결
- 클라이언트 제거
- 모범 사례
- 문제 해결
- 버전 기록

사전 조건

시작하기 전에 다음 필수 조건을 완료합니다.

- 신뢰 공급자를 사용하여 Verified·Access 인스턴스를 생성합니다.
- 애플리케이션에 대한 TCP 엔드포인트를 생성합니다.
- 라우팅 문제를 방지하려면 컴퓨터를 VPN 클라이언트에서 연결 해제합니다.
- 컴퓨터에서 IPv6를 활성화합니다. 지침은 컴퓨터에서 실행 중인 운영 체제에 대한 설명서를 참조하세요.

- Windows 컴퓨터에서 신뢰할 수 있는 플랫폼 모듈(TPM)이 지원되는지 확인하고 WebView2 런타임을 설치합니다.

연결 클라이언트 다운로드

이전 버전의 클라이언트를 제거합니다. 클라이언트를 다운로드하고 설치 프로그램이 서명되었는지 확인한 다음 설치 프로그램을 실행합니다. 서명되지 않은 설치 관리자를 사용하여 클라이언트를 설치하지 마십시오.

- Mac용 Connectivity Client와 Apple Silicon 버전 1.0.2
- Mac용 Connectivity Client와 Intel 버전 1.0.2
- x64 버전 1.0.2를 사용하는 Windows용 연결 클라이언트

클라이언트 구성 파일 내보내기

다음 절차에 따라 Verified Access 인스턴스에서 클라이언트에 필요한 구성 정보를 내보냅니다.

콘솔을 사용하여 클라이언트 구성 파일을 내보내려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 작업, 클라이언트 구성 파일 내보내기를 선택합니다.

를 사용하여 클라이언트 구성 파일을 내보내려면 AWS CLI

export-verified-access-instance-client-configuration 명령을 사용합니다. 출력을 .json 파일에 저장합니다. 파일 이름은 `ClientConfig-` 접두사로 시작해야 합니다.

애플리케이션에 연결

다음 절차에 따라 클라이언트를 사용하여 애플리케이션에 연결합니다.

클라이언트를 사용하여 애플리케이션에 연결하려면

1. 다음 위치에 있는 사용자의 디바이스에 클라이언트 구성 파일을 배포합니다.

- Windows - `C:\ProgramData\Connectivity Client`
- macOS – `/Library/Application\ Support/Connectivity\ Client`

2. 클라이언트 구성 파일이 루트(macOS) 또는 관리자(Windows) 소유인지 확인합니다.
3. 연결 클라이언트를 시작합니다.
4. 연결 클라이언트가 로드되면 사용자는 IdP에 의해 인증됩니다.
5. 인증 후 사용자는 선택한 클라이언트를 사용하여 Verified Access에서 제공하는 DNS 이름을 사용하여 애플리케이션에 액세스할 수 있습니다.

클라이언트 제거

연결 클라이언트 사용을 마치면 제거할 수 있습니다.

macOS

버전 1.0.1 이상

`/Applications/Connectivity Client`로 이동한 후 `Connectivity Client Uninstaller.app`를 실행합니다.

버전 1.0.0

Mac with Apple Silicon 또는 Mac with Intel용 `connectivity_client_cleanup.sh` 스크립트를 다운로드하고 스크립트에 대한 실행 권한을 설정한 다음 다음과 같이 스크립트를 실행합니다.

```
sudo ./connectivity_client_cleanup.sh
```

Windows

Windows에서 클라이언트를 제거하려면 설치 관리자를 실행하고 제거를 선택합니다.

모범 사례

다음 모범 사례를 고려하세요.

- 최신 버전의 클라이언트를 설치합니다.
- 서명되지 않은 설치 관리자를 사용하여 클라이언트를 설치하지 마십시오.

- IT 관리자가 제공하는 신뢰할 수 있는 구성이 아닌 한 사용자는 구성을 사용해서는 안 됩니다. 신뢰할 수 없는 구성은 피싱 페이지로 리디렉션될 수 있습니다.
- 사용자는 워크스테이션을 유휴 상태로 두기 전에 클라이언트에서 로그아웃해야 합니다.
- OIDC 구성에 `offline_access` 범위를 추가합니다. 이렇게 하면 사용자가 재인증할 필요 없이 더 많은 액세스 토큰을 얻는 데 사용되는 새로 고침 토큰에 대한 요청이 허용됩니다.

문제 해결

다음 정보는 클라이언트 관련 문제를 해결하는 데 도움이 될 수 있습니다.

문제

- 로그인할 때 브라우저가 열려 IdP에 의한 인증을 완료하지 않습니다.
- 인증 후 클라이언트 상태는 "연결되지 않음"입니다.
- Chrome 또는 Edge 브라우저를 사용하여 연결할 수 없음

로그인할 때 브라우저가 열려 IdP에 의한 인증을 완료하지 않습니다.

가능한 원인: 구성 파일이 누락되었거나 형식이 잘못되었습니다.

해결 방법: 시스템 관리자에게 문의하여 업데이트된 구성 파일을 요청합니다.

인증 후 클라이언트 상태는 "연결되지 않음"입니다.

가능한 원인: , AWS Client VPN Cisco AnyConnect 또는 OpenVPN Connect와 같은 다른 VPN 소프트웨어 실행.

해결 방법: 다른 VPN 소프트웨어와의 연결을 해제합니다. 그래도 연결할 수 없는 경우 진단 보고서를 생성하고 시스템 관리자와 공유합니다.

가능한 원인: Windows 플랫폼에서 클라이언트는 컨트롤 플레인 통신을 위해 포트 80에서 HTTP를 사용합니다. TCP 포트 80을 차단하는 방화벽 규칙은 컨트롤 플레인 통신을 방지합니다.

해결 방법: Windows 방화벽 규칙에서 포트 80의 TCP를 차단하는 명시적 아웃바운드 규칙을 확인하고 비활성화합니다.

Chrome 또는 Edge 브라우저를 사용하여 연결할 수 없음

가능한 원인: Chrome 또는 Edge 브라우저를 사용하여 웹 애플리케이션에 연결할 때 브라우저가 IPv6 도메인 이름을 확인하지 못합니다.

해결 방법:에 문의하세요AWS Support.

버전 기록

다음 표에는 클라이언트의 버전 기록이 나와 있습니다.

버전	변경 사항	다운로드	날짜
1.0.2	macOS • 버그 수정 및 안정성 개선 • UI 개선 사항 Windows • 버그 수정 및 안정성 개선 • UI 개선 사항	• Mac과 Apple Silicon • Mac과 Intel • x64가 있는 Windows	2025년 6월 9일
1.0.1	macOS • 안정성 개선 • 제거 프로그램 Windows • 안정성 개선	• Mac과 Apple Silicon • Mac과 Intel • x64가 있는 Windows	2025년 2월 5일
1.0.0	공개 평가판	• Mac과 Apple Silicon • Mac과 Intel • x64가 있는 Windows	2024년 12월 1일

Verified·Access의 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. AWS Verified·Access에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스규정 준수 프로그램 .
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Verified·Access 사용 시 책임 분담 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 Verified·Access를 구성하는 방법을 보여줍니다. 또한 Verified Access 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

내용

- Verified·Access의 데이터 보호
- Verified·Access의 ID 및 액세스 관리
- Verified·Access의 규정 준수 확인
- Verified·Access의 복원성

Verified·Access의 데이터 보호

AWS 공동 책임 모델 AWS Verified Access의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하

세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션과 내부의 모든 기본 보안 제어를 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Verified Access 또는 기타 AWS 서비스 에서 콘솔, API AWS CLI또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

전송 중 암호화

Verified·Access는 전송 계층 보안(TLS) 1.2 이상을 사용하여 인터넷을 통해 최종 사용자에서 Verified·Access 엔드포인트로 전송되는 모든 데이터를 암호화합니다.

인터네트워크 트래픽 개인 정보 보호

VPC의 특정 리소스에 대한 액세스를 제한하도록 Verified·Access를 구성할 수 있습니다. 사용자 기반 인증의 경우 엔드포인트에 액세스하는 사용자 그룹을 기반으로 네트워크 일부에 대한 액세스를 제한할 수도 있습니다. 자세한 내용은 Verified·Access 정책 단원을 참조하십시오.

AWS Verified Access에 대한 저장 데이터 암호화

AWS Verified Access는 기본적으로 AWS 소유 KMS 키를 사용하여 저장 데이터를 암호화합니다. 저장 데이터를 기본적으로 암호화하면 민감한 데이터를 보호하는 데 수반되는 운영 오버헤드와 복잡성을 줄이는 데 도움이 됩니다. 동시에 엄격한 암호화 규정 준수 및 규제 요구 사항을 충족하는 안전한 애플리케이션을 구축할 수 있습니다. 다음 섹션에서는 Verified·Access가 저장된 데이터 암호화에 KMS 키를 사용하는 방법에 대한 세부 정보를 제공합니다.

내용

- Verified·Access 및 KMS 키
- 개인 식별 정보
- AWS Verified Access가에서 권한 부여를 사용하는 방법 AWS KMS
- Verified·Access로 고객 관리형 키 사용
- Verified·Access 리소스에 대한 고객 관리형 키 지정
- AWS Verified Access 암호화 컨텍스트
- AWS Verified Access에 대한 암호화 키 모니터링

Verified·Access 및 KMS 키

AWS 소유 키

Verified·Access는 KMS 키를 사용하여 개인 식별 정보(PII)를 자동으로 암호화합니다. 이는 기본적으로 발생하며 AWS 소유 키의 사용을 직접 확인, 관리, 사용 또는 감사할 수 없습니다. 하지만 데이터를 암호화하는 키를 보호하기 위해 어떤 조치를 취하거나 어떤 프로그램을 변경할 필요가 없습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS 소유 키를 참조하세요.

이 암호화 계층을 비활성화하거나 대체 암호화 유형을 선택할 수는 없지만 Verified Access 리소스를 생성할 때 고객 관리형 키를 선택하여 기존 AWS 소유 암호화 키에 두 번째 암호화 계층을 추가할 수 있습니다.

고객 관리형 키

Verified·Access는 생성 및 관리하는 대칭 고객 관리형 키를 사용하여 기존 기본 암호화에 두 번째 암호화 계층을 추가할 수 있도록 지원합니다. 이 암호화 계층을 완전히 제어할 수 있으므로 다음과 같은 작업을 수행할 수 있습니다.

- 키 정책 수립 및 유지

- IAM 정책 및 권한 부여 수립 및 유지
- 키 정책 활성화 및 비활성화
- 키 암호화 자료 교체
- 태그 추가
- 키 별칭 만들기
- 삭제를 위한 스케줄 키

자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키를 참조하십시오.

> **Note**
>
> Verified Access는 AWS 소유 키를 사용하여 유휴 시 암호화를 자동으로 활성화하여 개인 식별 데이터를 무료로 보호합니다.
> 그러나 고객 관리형 키를 사용하는 경우 AWS KMS 요금이 적용됩니다. 요금에 대한 자세한 내용은 AWS Key Management Service 요금을 참조하십시오.

개인 식별 정보

다음 표는 Verified·Access에서 사용하는 개인 식별 정보(PII) 및 암호화 방법을 요약합니다.

데이터 유형	AWS 소유 키 암호화	고객 관리형 키 암호화(선택 사항)
`Trust provider (user-type)` 사용자 유형 신뢰 공급자에는 AuthorizationEndpoint, UserInfoEndpoint, ClientId, ClientSecret 등과 같이 PII로 간주되는 OIDC 옵션이 포함되어 있습니다.	활성화됨	활성화됨
`Trust provider (device-type)`	활성화됨	활성화됨

데이터 유형	AWS 소유 키 암호화	고객 관리형 키 암호화(선택 사항)
디바이스 유형 신뢰 공급자에는 PII로 간주되는 TenantId가 포함되어 있습니다.		
`Group policy` Verified·Access 그룹을 생성하거나 수정하는 동안 제공됩니다. 액세스 요청 승인에 대한 규칙을 포함합니다. 사용자 이름, 이메일 주소 등과 같은 PII를 포함할 수 있습니다.	활성화됨	활성화됨
`Endpoint policy` Verified·Access 엔드포인트를 생성하거나 수정하는 동안 제공됩니다. 액세스 요청 승인에 대한 규칙을 포함합니다. 사용자 이름, 이메일 주소 등과 같은 PII를 포함할 수 있습니다.	활성화됨	활성화됨

AWS Verified Access가에서 권한 부여를 사용하는 방법 AWS KMS

Verified·Access는 고객 관리형 키를 사용할 수 있는 권한이 필요합니다.

고객 관리형 키로 암호화된 Verified Access 리소스를 생성하면 Verified Access는 CreateGrant 요청을에 전송하여 사용자를 대신하여 권한을 생성합니다 AWS KMS. 의 권한 부여 AWS KMS 는 Verified Access에 계정의 고객 관리형 키에 대한 액세스 권한을 부여하는 데 사용됩니다.

Verified·Access는 다음 내부 작업에 대해 고객 관리형 키를 사용할 수 있는 권한이 필요합니다.

- AWS KMS 에 복호화 요청을 보내 암호화된 데이터 키를 복호화하여 데이터를 복호화하는 데 사용할 수 있도록 합니다.
- 권한 부여를 삭제 AWS KMS 하려면에 RetireGrant 요청을 보냅니다.

언제든지 권한 부여에 대한 액세스 권한을 취소하거나 고객 관리형 키에 대한 서비스 액세스를 제거할 수 있습니다. 그렇게 하면 Verified·Access는 고객 관리형 키로 암호화된 데이터에 액세스할 수 없으며, 이는 해당 데이터에 의존하는 작업에 영향을 미칩니다.

Verified·Access로 고객 관리형 키 사용

AWS Management Console또는 AWS KMS APIs. AWS Key Management Service 개발자 안내서의 대칭 암호화 키 생성 단계를 따릅니다.

키 정책

키 정책에서는 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 만들 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 정책을 참조하세요.

Verified·Access 리소스에서 고객 관리형 키를 사용하려면 키 정책에서 다음 API 작업을 허용해야 합니다.

- `kms:CreateGrant` - 고객 관리형 키에 권한 부여를 추가합니다. 지정된 KMS 키에 제어 액세스 권한을 부여하여 Verified·Access에 필요한 작업을 허용할 수 있는 액세스 권한을 부여합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 권한 부여를 참조하세요.

  이를 통해 Verified·Access는 다음을 수행할 수 있습니다.
  - 데이터 키가 암호화에 즉시 사용되지 않으므로 암호화된 데이터 키를 생성하고 저장하려면 `GenerateDataKeyWithoutPlainText`를 호출합니다.
  - 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스하려면 `Decrypt`를 호출합니다.
  - `RetireGrant`에 대한 서비스를 허용하도록 사용 중지 주체를 설정합니다.
- `kms:DescribeKey` – Verified·Access에서 키를 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.
- `kms:GenerateDataKey` - Verified·Access가 키를 사용하여 데이터를 암호화할 수 있도록 허용합니다.
- `kms:Decrypt` - Verified·Access가 암호화된 데이터 키를 해독할 수 있도록 허용합니다.

다음은 Verified·Access에 사용할 수 있는 키 정책의 예입니다.

```
"Statement" : [
    {
```

```
      "Sid" : "Allow access to principals authorized to use Verified Access",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "*"
      },
      "Action" : [
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "kms:ViaService" : "verified-access.region.amazonaws.com",
          "kms:CallerAccount" : "111122223333"
        }
    },
    {
      "Sid": "Allow access for key administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
       },
      "Action" : [
        "kms:*"
       ],
      "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    {
      "Sid" : "Allow read-only access to key metadata to the account",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "kms:RevokeGrant"
      ],
      "Resource" : "*"
    }
```

```
]
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 정책 생성 및 키 액세스 문제 해결을 참조하세요.

Verified·Access 리소스에 대한 고객 관리형 키 지정

고객 관리형 키를 지정하여 다음 리소스에 2차 계층 암호화를 제공할 수 있습니다.

- Verified·Access 그룹
- Verified·Access 엔드포인트
- Verified·Access 신뢰 공급자

를 사용하여 이러한 리소스를 생성할 때 추가 암호화 -- 선택 사항 섹션에서 고객 관리형 키를 지정할 AWS Management Console수 있습니다. 프로세스 중에 암호화 설정 사용자 지정(고급) 확인란을 선택한 다음 사용하려는 AWS KMS 키 ID를 입력합니다. 기존 리소스를 수정하거나 AWS CLI를 사용하여 이 작업을 수행할 수도 있습니다.

> **Note**
>
> 위의 리소스에 암호화를 더 추가하는 데 사용되는 고객 관리형 키가 손실될 경우 리소스의 구성 값에 더 이상 액세스할 수 없습니다. 그러나 AWS Management Console 또는를 사용하여 새 고객 관리형 키를 AWS CLI적용하고 구성 값을 재설정하여 리소스를 수정할 수 있습니다.

AWS Verified Access 암호화 컨텍스트

암호화 컨텍스트는 데이터에 대한 추가 컨텍스트 정보가 포함된 선택적 키-값 페어 세트입니다.는 암호화 컨텍스트를 인증된 추가 데이터로 AWS KMS 사용하여 인증된 암호화를 지원합니다. 데이터 암호화 요청에 암호화 컨텍스트를 포함하면는 암호화 컨텍스트를 암호화된 데이터에 AWS KMS 바인딩합니다. 요청에 동일한 암호화 컨텍스트를 포함해야 이 데이터를 해독할 수 있습니다.

AWS Verified Access 암호화 컨텍스트

Verified Access는 모든 암호화 작업에서 동일한 암호화 컨텍스트를 사용합니다. 여기서 키는 AWS KMS `aws:verified-access:arn`이고 값은 리소스 Amazon 리소스 이름(ARN)입니다. 다음은 Verified·Access 리소스의 암호화 컨텍스트입니다.

Verified·Access 신뢰 공급자

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessTrustProviderId"
}
```

Verified·Access 그룹

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessGroupId"
}
```

Verified·Access 엔드포인트

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessEndpointId"
}
```

AWS Verified Access에 대한 암호화 키 모니터링

AWS Verified Access 리소스와 함께 고객 관리형 KMS 키를 사용하는 경우 AWS CloudTrail를 사용하여 Verified Access가 보내는 요청을 추적할 수 있습니다 AWS KMS.

다음 예제는 고객 관리형 KMS 키로 암호화된 데이터에 액세스하기 위해 Verified Access에서 호출한 KMS 작업을 `GenerateDataKey`모니터링하는 , `CreateGrant` `RetireGrantDecrypt`, `DescribeKey`, 및에 대한 AWS CloudTrail 이벤트입니다.

CreateGrant

고객 관리형 키를 사용하여 리소스를 암호화하는 경우 Verified·Access는 사용자를 대신하여 AWS 계정의 키에 액세스하라는 `CreateGrant` 요청을 보냅니다. Verified·Access에서 생성하는 권한 부여는 고객 관리형 키와 연결된 리소스에만 적용됩니다.

다음 예제 이벤트는 `CreateGrant` 작업을 기록합니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T16:27:12Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T16:41:42Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "operations": [
            "Decrypt",
            "RetireGrant",
            "GenerateDataKey"
        ],
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae",
        "constraints": {
            "encryptionContextSubset": {
                "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-0e54f581e2e5c97a2"
            }
        },
        "granteePrincipal": "verified-access.ca-central-1.amazonaws.com",
        "retiringPrincipal": "verified-access.ca-central-1.amazonaws.com"
    },
    "responseElements": {
```

```
        "grantId":
 "e5a050fff9893ba1c43f83fddf61e5f9988f579beaadd6d4ad6d1df07df6048f",
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
    },
    "requestID": "0faa837e-5c69-4189-9736-3957278e6444",
    "eventID": "1b6dd8b8-cbee-4a83-9b9d-d95fa5f6fd08",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RetireGrant

Verified·Access는 리소스를 삭제할 때 `RetireGrant` 작업을 사용하여 권한 부여를 제거합니다.

다음 예제 이벤트는 `RetireGrant` 작업을 기록합니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
```

```
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T16:42:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T16:47:53Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RetireGrant",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": null,
    "responseElements": {
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
    },
    "additionalEventData": {
        "grantId":
 "b35e66f9bacb266cec214fcaa353c9cf750785e28773e61ba6f434d8c5c7632f"
    },
    "requestID": "7d4a31c2-d426-434b-8f86-336532a70462",
    "eventID": "17edc343-f25b-43d4-bbff-150d8fff4cf8",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Decrypt

Verified·Access는 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스하는 `Decrypt` 작업을 호출합니다.

다음 예제 이벤트는 Decrypt 작업을 기록합니다.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:47:05Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e",
        "encryptionContext": {
            "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-00f20a4e455e9340f",
            "aws-crypto-public-key": "AkK+vi1W/
acBKv7OR8p2DeUrA8EgpTffSrjBqNucODuBYhyZ3hlMuYYJz9x7CwQWZw=="
        }
    },
    "responseElements": null,
```

```
    "requestID": "2e920fd3-f2f6-41b2-a5e7-2c2cb6f853a9",
    "eventID": "3329e0a3-bcfb-44cf-9813-8106d6eee31d",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DescribeKey

Verified·Access는 `DescribeKey` 작업을 사용하여 리소스와 연결된 고객 관리형 키가 계정 및 리전에 존재하는지 확인합니다.

다음 예제 이벤트는 `DescribeKey` 작업을 기록합니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
```

```
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:46:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
    },
    "responseElements": null,
    "requestID": "5b127082-6691-48fa-bfb0-4d40e1503636",
    "eventID": "ffcfc2bb-f94b-4c00-b6fb-feac77daff2a",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateDataKey

다음 예제 이벤트는 `GenerateDataKey` 작업을 기록합니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:46:49Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-00f20a4e455e9340f",
            "aws-crypto-public-key": "A/ATGxaYatPUlOtM+l/mfDndkzHUmX5Hav+29IlIm
+JRBKFuXf24ulztmOIsqFQliw=="
        },
        "numberOfBytes": 32,
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
    },
    "responseElements": null,
    "requestID": "06535808-7cce-4ae1-ab40-e3afbf158a43",
    "eventID": "1ce79601-5a5e-412c-90b3-978925036526",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Verified·Access의 ID 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 누가 Verified·Access 리소스를 사용하도록 인증되고(로그인됨) 권한이 부여되는지(권한 있음)를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- Verified·Access가 IAM과 함께 작동하는 방식
- Verified·Access에 대한 자격 증명 기반 정책 예제
- Verified·Access 자격 증명 및 액세스 문제 해결
- Verified·Access에 대한 서비스 연결 역할 사용
- AWS Verified·Access에 대한 관리형 정책

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 Verified·Access에서 수행하는 작업에 따라 다릅니다.

서비스 사용자 - Verified·Access 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 Verified·Access 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다. Verified·Access의 기능에 액세스할 수 없는 경우 Verified·Access 자격 증명 및 액세스 문제 해결를 참조하십시오.

서비스 관리자 - 회사에서 Verified·Access 리소스를 책임지고 있는 경우 Verified·Access에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 Verified·Access 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사가 Verified·Access에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 Verified·Access가 IAM과 함께 작동하는 방식를 참조하십시오.

IAM 관리자 - IAM 관리자라면 Verified·Access에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 Verified·Access 자격 증명 기반 정책 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. , AWS 계정 루트 사용자 IAM 사용자 또는 IAM 역할을 수임하여 인증(로그인 AWS)되어야 합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 자격 증명 AWS 으로에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의에 로그인하는 방법을 AWS참조하세요. AWS 계정

AWS 프로그래밍 방식으로에 액세스하는 경우는 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 자세한 방법은 IAM 사용 설명서에서 API 요청용AWS Signature Version 4를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화할 것을 AWS 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 다중 인증 및 IAM 사용 설명서에서 IAM의AWS 다중 인증을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명을 AWS 계정 루트 사용자라고 하며 계정을 생성하는 데 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지

않을 것을 강력히 권장합니다. 루트 사용자 자격 증명을 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하세요.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스 에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, AWS Directory Service, Identity Center 디렉터리 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스 에 액세스하는 모든 사용자의 사용자입니다. 페더레이션 자격 증명에 액세스할 때 역할을 AWS 계정수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을(를) 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 및 애플리케이션에서 사용할 수 있도록 자체 ID 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇인가요?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 사용자 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우, 정기적으로 액세스 키 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 자격 증명을 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서에서 IAM 사용자 사용 사례를 참조하세요.

IAM 역할

IAM 역할은 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 에서 IAM 역할을 일시적으로 수임하려면 사용자에서 IAM 역할(콘솔)로 전

환할 AWS Management Console수 있습니다. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-console.html 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 AWS CLI 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 관련 역할에 대한 자세한 내용은 IAM 사용 설명서의 Create a role for a third-party identity provider (federation)를 참조하세요. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 집합을 IAM의 역할과 연관짓습니다. 권한 집합에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 집합을 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 교차 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 에서는 정책을 리소스에 직접 연결할 AWS 서비스수 있습니다(역할을 프록시로 사용하는 대신). 교차 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하세요.
- 교차 서비스 액세스 - 일부는 다른에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 예를 들어, 서비스에서 호출하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 위탁자의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 완료하기 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 수신할 때만 이루어집니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 Create a role to delegate permissions to an AWS 서비스를 참조하세요.

- 서비스 연결 역할 - 서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 애플리케이션 - IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로필에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결된 AWS 경우 권한을 정의하는의 객체입니다.는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자 및 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console AWS CLI, 또는 API에서 역할 정보를 가져올 수 있습니다 AWS .

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은의 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 위탁자가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 위탁자를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는이 포함될 수 있습니다 AWS 서비스.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3 AWS WAF및 Amazon VPC는 ACLs. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 ID 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 객체의 ID 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔티티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책(SCPs) - SCPs는의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책입니다 AWS Organizations. AWS Organizations 는 비즈니스가 소유한 여러 AWS 계정 을 그

룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각각을 포함하여 멤버 계정의 엔터티에 대한 권한을 제한합니다 AWS 계정 루트 사용자. 조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 Service control policies을 참조하세요.

- 리소스 제어 정책(RCP) - RCP는 소유한 각 리소스에 연결된 IAM 정책을 업데이트하지 않고 계정의 리소스에 대해 사용 가능한 최대 권한을 설정하는 데 사용할 수 있는 JSON 정책입니다. RCP는 멤버 계정의 리소스에 대한 권한을 제한하며 조직에 속하는지 여부에 AWS 계정 루트 사용자관계없이를 포함한 자격 증명에 대한 유효 권한에 영향을 미칠 수 있습니다. RCP를 AWS 서비스 지원하는 목록을 포함하여 조직 및 RCPs에 대한 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCPs)을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 ID 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

Verified·Access가 IAM과 함께 작동하는 방식

IAM을 사용하여 Verified·Access에 대한 액세스를 관리하기 전에 Verified·Access와 함께 사용할 수 있는 IAM 기능을 알아보십시오.

IAM 기능	Verified·Access 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예

IAM 기능	Verified·Access 지원
정책 조건 키	예
ACLs	아니요
ABAC(정책 내 태그)	부분
임시 자격 증명	예
보안 주체 권한	예
서비스 역할	아니요
서비스 링크 역할	예

Verified·Access 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방식을 개괄적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

Verified·Access에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 위탁자가 연결된 사용자 또는 역할에 적용되므로 위탁자를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하십시오.

Verified·Access에 대한 자격 증명 기반 정책 예제

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

Verified·Access 내 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 위탁자가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 위탁자를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는이 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 위탁자로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 보안 주체와 리소스가 다른 경우 신뢰할 수 AWS 계정있는 계정의 IAM 관리자는 보안 주체 엔터티(사용자 또는 역할)에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 위탁자에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

Verified·Access에 대한 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 위탁자가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

Verified·Access 작업 목록을 보려면 서비스 승인 참조의 Amazon EC2에서 정의한 작업을 참조하십시오.

Verified·Access의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
ec2
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
    "ec2:action1",
    "ec2:action2"
]
```

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

Verified·Access에 대한 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 문에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

Verified·Access 리소스 유형 및 해당 ARN 목록을 보려면 서비스 승인 참조에서 Amazon EC2에서 정의한 리소스를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 Amazon EC2에서 정의한 작업을 참조하십시오.

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

Verified·Access에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 `OR` 작업을 사용하여 조건을 AWS 평가합니다. 문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

Verified·Access 조건 키 목록을 보려면 서비스 승인 참조의 Amazon EC2에 사용되는 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 Amazon EC2에서 정의한 작업을 참조하십시오.

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

Verified·Access의 ACL

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 위탁자(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Verified·Access 기능을 사용한 ABAC

ABAC 지원(정책의 태그): 부분적

속성 기반 액세스 제어(ABAC)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. 에서는 AWS이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연

결할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 위탁자의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

Verified·Access에서 임시 자격 증명 사용

임시 자격 증명 지원: 예

일부 AWS 서비스 는 임시 자격 증명을 사용하여 로그인할 때 작동하지 않습니다. 임시 자격 증명으로 AWS 서비스 작업하는를 포함한 추가 정보는 AWS 서비스 IAM 사용 설명서의 IAM으로 작업하는 섹션을 참조하세요.

사용자 이름 및 암호를 제외한 방법을 AWS Management Console 사용하여에 로그인하는 경우 임시 자격 증명을 사용합니다. 예를 들어 회사의 SSO(Single Sign-On) 링크를 AWS 사용하여에 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 자격 증명을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다. 그런 다음 이러한 임시 자격 증명을 사용하여 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 access AWS. AWS recommds에 액세스할 수 있습니다. 자세한 정보는 IAM의 임시 보안 자격 증명 섹션을 참조하십시오.

Verified·Access의 서비스 간 보안 주체 권한

전달 액세스 세션(FAS) 지원: 예

IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 완료하기 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청

을 수신할 때만 이루어집니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

Verified·Access를 위한 서비스 역할

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 Create a role to delegate permissions to an AWS 서비스를 참조하세요.

Verified·Access를 위한 서비스 연결 역할

서비스 링크 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

Verified·Access 서비스 연결 역할을 생성 또는 관리하는 방법에 대한 자세한 내용은 Verified·Access에 대한 서비스 연결 역할 사용 섹션을 참조하십시오.

Verified·Access에 대한 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할에는 Verified·Access 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 Verified·Access에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- Verified·Access 인스턴스 생성 정책
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Verified·Access 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 AWS CloudFormation. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

Verified·Access 인스턴스 생성 정책

Verified·Access 인스턴스를 생성하려면 IAM 보안 주체가 IAM 정책에 이 추가 설명을 추가해야 합니다.

```
{
```

```
    "Effect": "Allow",
    "Action": "verified-access:AllowVerifiedAccess",
    "Resource": "*"
}
```

> **Note**
>
> `verified-access:AllowVerifiedAccess`은(는) 작업 전용 가상 API 입니다. 리소스, 태그 또는 조건 키 기반 권한 부여는 지원하지 않습니다. `ec2:CreateVerifiedAccessInstance` API 작업에서 리소스, 태그 또는 조건 키 기반 인증을 사용합니다.

Verified·Access 인스턴스 생성에 대한 예제 정책입니다. 이 예제에서 `123456789012`는 AWS 계정 번호이고 `us-east-1`는 AWS 리전입니다.

JSON

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVerifiedAccessInstance",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:verified-access-
instance/*"
        },
        {
            "Effect": "Allow",
            "Action": "verified-access:AllowVerifiedAccess",
            "Resource": "*"
        }
    ]
}
```

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Verified·Access 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Verified·Access 및 IAM에서 발생할 수 있는 공통적인 문제를 진단하고 수정할 수 있습니다.

문제

- Verified·Access에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 Verified Access 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

Verified·Access에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`* 리소스에 대한 세부 정보를 보려고 하지만 가상 `ec2:`*`GetWidget`* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:GetWidget on resource: my-example-widget
```

이 경우, `ec2:`*`GetWidget`* 작업을 사용하여 *`my-example-widget`* 리소스에 액세스할 수 있도록 `mateojackson` 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Verified·Access에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 `marymajor`(이)라는 IAM 사용자가 콘솔을 사용하여 Verified·Access에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 Verified Access 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하십시오.

- Verified·Access에서 이러한 기능을 지원하는지 여부를 알아보려면 Verified·Access가 IAM과 함께 작동하는 방식를 참조하십시오.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

Verified·Access에 대한 서비스 연결 역할 사용

AWS Verified Access 는 서비스에 직접 연결된 IAM 역할 유형인 IAM AWS 서비스 연결 역할을 사용합니다. Verified·Access에 대한 서비스 연결 역할은 Verified·Access에서 정의하며 서비스가 AWS 서비스 사용자를 대신하여 다른를 호출하는 데 필요한 모든 권한을 포함합니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할은 Verified·Access를 더 쉽게 설정할 수 있습니다. Verified·Access에서 서비스 연결 역할의 권한을 정의하므로 다르게 정의되지 않은 한,

Verified·Access만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 개체에 연결할 수 없습니다.

Verified·Access에 대한 서비스 연결 역할 권한

Verified·Access는 AWSServiceRoleForVPCVerifiedAccess라는 서비스 연결 역할을 사용하여 서비스를 사용하는 데 필요한 사용자 계정 내 리소스를 프로비저닝합니다.

AWSServiceRoleForVPCVerifiedAccess 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `verified-access.amazonaws.com`

이름이 AWSVPCVerifiedAccessServiceRolePolicy인 연결 권한 정책은 Verified·Access가 지정된 리소스에 대해 다음 작업을 수행하도록 허용합니다.

- 모든 서브넷 및 보안 그룹과 `VerifiedAccessManaged=true` 태그가 있는 모든 네트워크 인터페이스에 대한 조치 `ec2:CreateNetworkInterface`
- 생성 시 모든 네트워크 인터페이스에 대한 조치 `ec2:CreateTags`
- `VerifiedAccessManaged=true` 태그가 있는 모든 네트워크 인터페이스에 대한 조치 `ec2:DeleteNetworkInterface`
- 모든 보안 그룹 및 `VerifiedAccessManaged=true` 태그가 있는 모든 네트워크 인터페이스에 대한 조치 `ec2:ModifyNetworkInterfaceAttribute`

AWS 관리형 정책 참조 안내서에서이 정책에 대한 권한을 볼 수도 있습니다. AWSVPCVerifiedAccessServiceRolePolicy를 참조하세요.

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 링크 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하십시오.

Verified·Access에 대한 서비스 연결 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. AWS CLI, 또는 AWS API에서 CreateVerifiedAccessEndpoint AWS Management Console를 호출하면 Verified Access가 서비스 연결 역할을 생성합니다.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. CreateVerifiedAccessEndpoint를 다시 호출하면 Verified·Access에서 서비스 연결 역할을 다시 생성합니다.

Verified·Access에 대한 서비스 연결 역할 편집

Verified·Access에서는 AWSServiceRoleForVPCVerifiedAccess 서비스 연결 역할을 편집할 수 없습니다. 서비스 링크 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 설명 편집을 참조하세요.

Verified·Access에 대한 서비스 연결 역할 삭제

AWSServiceRoleForVPCVerifiedAccess 역할을 수동으로 삭제하지 않아도 됩니다. AWS CLI, 또는 AWS API에서 DeleteVerifiedAccessEndpoint AWS Management Console를 호출하면 Verified Access가 리소스를 정리하고 서비스 연결 역할을 삭제합니다.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 AWSServiceRoleForVPCVerifiedAccess 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하세요.

Verified·Access 서비스 연결 역할이 지원되는 리전

Verified·Access는 서비스를 사용할 수 AWS 리전 있는 모든에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 AWS 리전 및 엔드포인트 섹션을 참조하세요.

AWS Verified·Access에 대한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 줍니다. AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 AWS 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AWSVPCVerifiedAccessServiceRolePolicy

이 정책은 Verified·Access가 사용자를 대신하여 작업을 수행할 수 있도록 서비스 연결 역할에 연결됩니다. 자세한 내용은 서비스 연결 역할 사용 단원을 참조하십시오. 이 정책의 권한을 보려면 에서 AWSVPCVerifiedAccessServiceRolePolicy를 보거나 AWS 관리형 정책 참조 안내서에서 AWSVPCVerifiedAccessServiceRolePolicy 정책을 볼 AWS Management Console수 있습니다.

AWS 관리형 정책에 대한 Verified·Access 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 Verified·Access의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 Verified·Access 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
AWSVPCVerifiedAccessServiceRolePolicy - 정책 업데이트됨	Verified Access에서 ‘sid’ 필드 아래에 모든 작업에 대한 설명을 포함하도록 관리형 정책을 업데이트했습니다.	2023년 11월 17일
AWSVPCVerifiedAccessServiceRolePolicy - 정책 업데이트됨	Verified Access에서 `ec2:CreateNetworkInterface` 권한에 보안 그룹 리소스를 추가하도록 관리형 정책을 업데이트했습니다.	2023년 5월 31일
AWSVPCVerifiedAccessServiceRolePolicy - 새로운 정책	Verified·Access는 서비스를 사용하는 데 필요한 리소스를 계정에 프로비저닝할 수 있도록 새로운 정책을 추가했습니다.	2022년 11월 29일
Verified·Access, 변경 내용 추적 시작	Verified·Access가 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2022년 11월 29일

Verified·Access의 규정 준수 확인

AWS Verified Access 는 FIPS(연방 정보 처리 표준) 규정 준수를 지원하도록 구성할 수 있습니다. Verified·Access의 FIPS 규정 준수를 설정하는 방법에 대한 더 많은 정보 및 세부 사항은 Verified·Access의 FIPS 규정 준수를 참조하십시오.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 규정 준수 AWS 서비스 프로그램 범위규정 준수 섹션을 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다.는 규정 준수를 지원하기 위해 다음 리소스를 AWS 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- HIPAA 적격 서비스 참조 - HIPAA 적격 서비스가 나열되어 있습니다. 모든 AWS 서비스 가 HIPAA에 적합한 것은 아닙니다.
- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 업계 및 위치에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에는 여러 프레임워크(미국 국립표준기술연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제표준화기구(ISO) 포함)의 보안 제어에 대한 지침을 보호하고 AWS 서비스 매핑하는 모범 사례가 요약되어 있습니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 -이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub - 이를 AWS 서비스 통해 내 보안 상태를 포괄적으로 확인할 수 있습니다 AWS. Security Hub는 보안 컨트롤을 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 의심스러운 악의적인 활동이 있는지 환경을 모니터링하여 사용자, AWS 계정 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험과 규정 및 업계 표준 준수를 관리하는 방법을 간소화할 수 있습니다.

Verified·Access의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 기반으로 구축됩니다.는 지연 시간이 짧고 처리량이 많으며 중복성이 높은 네트워킹과 연결된 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공합니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

AWS 글로벌 인프라 외에도 Verified Access는 고가용성 요구 사항을 지원하는 데 도움이 되는 다음 기능을 제공합니다.

고가용성을 위한 다중 서브넷

로드 밸런서 유형의 Verified·Access 엔드포인트를 생성할 때 엔드포인트에 서브넷을 여러 개 연결할 수 있습니다. 엔드포인트와 연결하는 각 서브넷은 서로 다른 가용 영역에 속해야 합니다. 서브넷을 여러 개 연결하면 여러 가용 영역을 사용하여 고가용성을 보장할 수 있습니다.

모니터링 AWS Verified Access

모니터링은의 신뢰성, 가용성 및 성능을 유지하는 데 중요한 부분입니다 AWS Verified Access.는 Verified Access를 모니터링하고, 문제가 있을 때 보고하고, 적절한 경우 자동 조치를 취할 수 있는 다음과 같은 모니터링 도구를 AWS 제공합니다.

- 액세스 로그 - 애플리케이션 액세스 요청에 대한 세부 정보를 캡처합니다. 자세한 내용은 the section called “Verified·Access 로그” 단원을 참조하십시오.
- AWS CloudTrail -에 의해 또는를 대신하여 수행된 API 호출 및 관련 이벤트를 캡처 AWS 계정 하고 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 호출된 사용자 및 계정 AWS, 호출이 수행된 원본 IP 주소, 호출이 발생한 시기를 식별할 수 있습니다. 자세한 내용은 the section called “CloudTrail 로그” 단원을 참조하십시오.

Verified·Access 로그

는 각 액세스 요청을 AWS Verified Access 평가한 후 모든 액세스 시도를 기록합니다. 이를 통해 애플리케이션 액세스에 대한 중앙 집중식 가시성을 제공하고 보안 인시던트 및 감사 요청에 신속하게 대응할 수 있습니다. Verified·Access는 Open Cybersecurity Schema Framework(OCSF) 로깅 형식을 지원합니다.

로깅을 활성화하는 경우 로그를 전송할 대상을 구성해야 합니다. 로깅이 제대로 작동하려면 로깅 대상을 구성하는 데 사용되는 IAM 위탁자에게 특정 권한이 있어야 합니다. 각 로깅 대상에 필요한 IAM 권한은 Verified Access 로깅 권한 섹션에서 확인할 수 있습니다. Verified·Access는 액세스 로그를 게시하기 위한 다음 대상을 지원합니다.

- Amazon CloudWatch Logs 로그 그룹
- Amazon S3 버킷
- Amazon Data Firehose 전송 스트림

내용

- Verified Access 로깅 버전
- Verified Access 로깅 권한
- Verified Access 로그 활성화 또는 비활성화
- Verified Access 신뢰 컨텍스트 활성화 또는 비활성화

- Verified Access에 대한 OCSF 버전 0.1 로그 예제
- Verified Access에 대한 OCSF 버전 1.0.0-rc.2 로그 예제

Verified Access 로깅 버전

기본적으로 Verified·Access 로깅 시스템은 Open Cybersecurity Schema Framework(OCSF) 버전 0.1을 사용합니다. 버전 0.1을 사용하는 샘플 로그는 섹션을 참조하세요Verified Access에 대한 OCSF 버전 0.1 로그 예제.

최신 로깅 버전은 OCSF 버전 1.0.0-rc.2와 호환됩니다. 스키마에 대한 자세한 내용은 OCSF 스키마를 참조하세요. 버전 1.0.0-rc.2를 사용하는 샘플 로그는 섹션을 참조하세요Verified Access에 대한 OCSF 버전 1.0.0-rc.2 로그 예제.

Verified Access 엔드포인트가 TCP 프로토콜을 사용하는 경우 OCSF 버전 0.1을 사용할 수 없습니다.

콘솔을 사용하여 로깅 버전을 업그레이드하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 적절한 Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 업데이트 로그 버전 드롭다운 목록에서 ocsf-1.0.0-rc.2를 선택합니다.
6. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 로깅 버전을 업그레이드하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용합니다.

Verified Access 로깅 권한

로깅이 제대로 작동하려면 로깅 대상을 구성하는 데 사용되는 IAM 위탁자에게 특정 권한이 있어야 합니다. 다음 섹션에서는 각 로깅 대상에 필요한 권한을 보여줍니다.

CloudWatch Logs로 전송하려면:

- Verified·Access 인스턴스에서 `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`

- 모든 리소스에서 `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDelivery`, `logs:ListLogDeliveries`, 및 `logs:UpdateLogDelivery`
- 대상 로그 그룹에서 `logs:DescribeLogGroups`, `logs:DescribeResourcePolicies`, 및 `logs:PutResourcePolicy`

Amazon S3로 전송하려면:

- Verified·Access 인스턴스에서 `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`
- 모든 리소스에서 `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDelivery`, `logs:ListLogDeliveries`, 및 `logs:UpdateLogDelivery`
- 그리고 대상 버킷에서 `s3:GetBucketPolicy` 및 `s3:PutBucketPolicy`

Firehose로 전송하는 경우:

- Verified·Access 인스턴스에서 `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`
- 모든 리소스에서 `firehose:TagDeliveryStream`
- 모든 리소스에서 `iam:CreateServiceLinkedRole`
- 모든 리소스에서 `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDelivery`, `logs:ListLogDeliveries`, 및 `logs:UpdateLogDelivery`

Verified Access 로그 활성화 또는 비활성화

이 섹션의 절차에 따라 로깅을 활성화하거나 비활성화할 수 있습니다. 로깅을 활성화하는 경우 로그를 전송할 대상을 구성해야 합니다. 로깅이 제대로 작동하려면 로깅 대상을 구성하는 데 사용되는 IAM 위탁자에게 특정 권한이 있어야 합니다. 각 로깅 대상에 필요한 IAM 권한은 Verified Access 로깅 권한 섹션에서 확인할 수 있습니다.

내용

- 액세스 로그 활성화
- 액세스 로그 비활성화

액세스 로그 활성화

Verified·Access 로그를 활성화하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. (선택 사항) 신뢰 공급자가 보낸 신뢰 데이터를 로그에 포함하려면 다음을 수행하십시오.
   a. 업데이트 로그 버전 드롭다운 목록에서 ocsf-1.0.0-rc.2를 선택합니다.
   b. 신뢰 컨텍스트 포함을 선택합니다.
6. 다음 중 하나를 수행하십시오.
   - Amazon CloudWatch Logs로 전송을 활성화합니다. 대상 로그 그룹을 선택합니다.
   - Amazon S3로 전송을 활성화합니다. 대상 버킷의 이름, 소유자 및 접두사를 입력합니다.
   - Firehose로 전송을 켭니다. 대상 전송 스트림을 선택합니다.
7. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 Verified Access 로그를 활성화하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용합니다.

액세스 로그 비활성화

Verified·Access 인스턴스의 액세스 로그는 언제든지 비활성화할 수 있습니다. 액세스 로그를 비활성화하면 로그 데이터는 사용자가 삭제할 때까지 로그 대상에 남아 있습니다.

Verified·Access 로그를 비활성화하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 로그 전송을 끕니다.

6. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 Verified Access 로그를 비활성화하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용합니다.

Verified Access 신뢰 컨텍스트 활성화 또는 비활성화

신뢰 공급자가 보낸 신뢰 컨텍스트는 선택적으로 Verified Access 로그에 포함되도록 활성화할 수 있습니다. 이는 애플리케이션에 대한 액세스를 허용하거나 거부하는 정책을 정의할 때 유용할 수 있습니다. 활성화하면 `data` 필드 아래의 로그에서 신뢰 컨텍스트를 찾을 수 있습니다. 신뢰 컨텍스트가 비활성화되면 `data` 필드가 `null`로 설정됩니다. 로그에 신뢰 컨텍스트를 포함하도록 Verified Access를 구성하려면 다음 절차를 따르세요.

> **Note**
>
> Verified·Access 로그에 신뢰 컨텍스트를 포함하려면 최신 로깅 버전 `ocsf-1.0.0-rc.2`로 업그레이드해야 합니다. 다음 절차에서는 이미 로깅을 활성화했다고 가정합니다. 그렇지 않은 경우 전체 절차 액세스 로그 활성화를 참조하십시오.

내용

- 신뢰 컨텍스트 활성화
- 신뢰 컨텍스트 비활성화

신뢰 컨텍스트 활성화

콘솔을 사용하여 Verified·Access 로그에 신뢰 컨텍스트를 포함하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 적절한 Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 업데이트 로그 버전 드롭다운 목록에서 ocsf-1.0.0-rc.2를 선택합니다.
6. 신뢰 컨텍스트 포함을 활성화합니다.

7. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 Verified Access 로그에 신뢰 컨텍스트를 포함하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용합니다.

신뢰 컨텍스트 비활성화

로그에 더 이상 신뢰 컨텍스트를 포함하지 않으려면 다음 절차에 따라 신뢰 컨텍스트를 제거할 수 있습니다.

콘솔을 사용하여 Verified·Access 로그에서 신뢰 컨텍스트를 제거하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 적절한 Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 신뢰 컨텍스트 포함을 끕니다.
6. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 Verified Access 로그에서 신뢰 컨텍스트를 제거하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용합니다.

Verified Access에 대한 OCSF 버전 0.1 로그 예제

다음은 OCSF 버전 0.1을 사용하는 샘플 로그입니다.

예시

- OIDC를 통한 액세스 권한 부여
- OIDC와 JAMF를 통한 액세스 권한 부여
- OIDC와 CrowdStrike를 통한 액세스 권한 부여
- 쿠키 누락으로 인한 액세스 거부
- 정책으로 인한 액세스 거부
- 알 수 없는 로그 항목

OIDC를 통한 액세스 권한 부여

이 예제 로그 항목에서 Verified·Access는 OIDC 사용자 신뢰 공급자를 통해 엔드포인트에 대한 액세스를 허용합니다.


```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
```

```
        ],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
```

```
}
```

OIDC와 JAMF를 통한 액세스 권한 부여

이 예제 로그 항목에서 Verified·Access는 OIDC 및 JAMF 디바이스 신뢰 공급자를 통해 엔드포인트에 대한 액세스를 허용합니다.

```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0,
        "uid": "41b07859-4222-4f41-f3b9-97dc1EXAMPLE"
    },
    "duration": "0.347",
    "end_time": "1668804944086",
    "time": "1668804944086",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 304
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
```

```
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "oidc",
            "uid": "vatp-9778003bc2EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "4f040d0f96becEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-321318ce-6100d340adf4fb29dEXAMPLE",
        "logged_time": 1668805278555,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-18T20:55:44.086480Z",
    "proxy": {
        "ip": "10.5.192.96",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-3598f66575EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "192.168.20.246",
        "port": 61769
    },
    "start_time": "1668804943739",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
```

```
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```

OIDC와 CrowdStrike를 통한 액세스 권한 부여

이 예제 로그 항목에서 Verified·Access는 OIDC 및 CrowdStrike 디바이스 신뢰 공급자를 통해 엔드포인트에 대한 액세스를 허용합니다.

```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.173.3",
        "os": {
            "name": "Windows 11",
            "type": "Windows",
            "type_id": 100
        },
        "type": "Unknown",
        "type_id": 0,
        "uid": "122978434f65093aee5dfbdc0EXAMPLE",
        "hw_info": {
            "serial_number": "751432a1-d504-fd5e-010d-5ed11EXAMPLE"
        }
    },
    "duration": "0.028",
    "end_time": "1668816620842",
    "time": "1668816620842",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "test.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "h2",
            "text": "https://test.app.example.com:443/"
        },
```

```
        "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 304
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "oidc",
            "uid": "vatp-506d9753f6EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "23bb45b16a389EXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-c16c5a65-b641e4056cc6cb0eeEXAMPLE",
        "logged_time": 1668816977134,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-19T00:10:20.842295Z",
    "proxy": {
        "ip": "192.168.144.62",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-2f80f37e64EXAMPLE"
    },
```

```
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.14.173.3",
        "port": 55706
    },
    "start_time": "1668816620814",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```

쿠키 누락으로 인한 액세스 거부

이 예제 로그 항목에서 Verified·Access는 인증 쿠키가 누락되어 액세스를 거부합니다.

```
{
    "activity": "Access Denied",
    "activity_id": "2",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": null,
    "duration": "0.0",
    "end_time": "1668593568259",
    "time": "1668593568259",
    "http_request": {
        "http_method": "POST",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/dns-query",
            "port": 443,
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/dns-query"
        },
        "user_agent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML",
        "version": "HTTP/2.0"
    },
```

```
    "http_response": {
        "code": 302
    },
    "identity": null,
    "message": "",
    "metadata": {
        "uid": "Root=1-5cf1c832-a565309ce20cc7dafEXAMPLE",
        "logged_time": 1668593776720,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T10:12:48.259762Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-108ed7a672EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.7.178.16",
        "port": "46246"
    },
    "start_time": "1668593568258",
    "status_code": "200",
    "status_details": "Authentication Denied",
    "status_id": "2",
    "status": "Failure",
    "type_uid": "20800102",
    "type_name": "AccessLogs: Access Denied",
    "unmapped": null
}
```


정책으로 인한 액세스 거부

이 예제 로그 항목에서 Verified·Access는 액세스 정책에 의해 요청이 허용되지 않기 때문에 인증된 요청을 거부합니다.


```
{
```

```
    "activity": "Access Denied",
    "activity_id": "2",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.4.133.137",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.023",
    "end_time": "1668573630978",
    "time": "1668573630978",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 401
    },
    "identity": {
        "authorizations": [],
        "idp": {
            "name": "user",
            "uid": "vatp-e048b3e0f8EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "0e1281ad3580aEXAMPLE"
        }
    },
    "message": "",
```

```
    "metadata": {
        "uid": "Root=1-531a036a-09e95794c7b96aefbEXAMPLE",
        "logged_time": 1668573773753,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T04:40:30.978732Z",
    "proxy": {
        "ip": "3.223.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-021d5eaed2EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.4.133.137",
        "port": "31746"
    },
    "start_time": "1668573630955",
    "status_code": "300",
    "status_details": "Authorization Denied",
    "status_id": "2",
    "status": "Failure",
    "type_uid": "20800102",
    "type_name": "AccessLogs: Access Denied",
    "unmapped": null
}
```

알 수 없는 로그 항목

이 예제 로그 항목에서 Verified·Access는 전체 로그 항목을 생성할 수 없으므로 알 수 없는 로그 항목을 내보냅니다. 이렇게 하면 모든 요청이 액세스 로그에 표시됩니다.

```
{
    "activity": "Unknown",
    "activity_id": "0",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
```

```
    "class_uid": "208001",
    "device": null,
    "duration": "0.004",
    "end_time": "1668580207898",
    "time": "1668580207898",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "identity": null,
    "message": "",
    "metadata": {
        "uid": "Root=1-435eb955-6b5a1d529343f5adaEXAMPLE",
        "logged_time": 1668580579147,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:30:07.898344Z",
    "proxy": {
        "ip": "10.1.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-6c32b53b3cEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.28.57.68",
        "port": "47220"
    },
```

```
    "start_time": "1668580207893",
    "status_code": "000",
    "status_details": "Unknown",
    "status_id": "0",
    "status": "Unknown",
    "type_uid": "20800100",
    "type_name": "AccessLogs: Unknown",
    "unmapped": null
}
```

Verified Access에 대한 OCSF 버전 1.0.0-rc.2 로그 예제

다음은 OCSF 버전 1.0.0-rc.2를 사용하는 샘플 로그입니다.

예시

- 신뢰 컨텍스트가 포함된 액세스 권한 부여
- 신뢰 컨텍스트가 생략된 액세스 권한 부여
- 네트워크 CIDR 엔드포인트를 사용하여 권한 할당

신뢰 컨텍스트가 포함된 액세스 권한 부여

```
{
    "activity_name": "Access Grant",
    "activity_id": "1",
    "actor": {
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
        }],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "invoked_by": "",
        "process": {},
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
```

```
            "uuid": "00u6wj48lbxTAEXAMPLE"
        },
        "session": {}
    },
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Access Activity",
    "class_uid": "3006",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
```

```
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_detail": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300601",
    "type_name": "Access Activity: Access Grant",
    "data": {
        "context": {
            "oidc": {
                "family_name": "Last",
                "zoneinfo": "America/Los_Angeles",
                "exp": 1670631145,
                "middle_name": "Middle",
                "given_name": "First",
                "email_verified": true,
                "name": "Test User Display",
                "updated_at": 1666305953,
                "preferred_username": "johndoe-user@test.com",
                "profile": "http://www.example.com",
                "locale": "US",
                "nickname": "Tester",
                "email": "johndoe-user@test.com",
                "additional_user_context": {
                    "aud": "xxx",
                    "exp": 1000000000,
                    "groups": [
                        "group-id-1",
                        "group-id-2"
                    ],
                    "iat": 1000000000,
                    "iss": "https://oidc-tp.com/",
                    "sub": "xyzsubject",
                    "ver": "1.0"
```

```
                }
            },
            "http_request": {
                "x_forwarded_for": "1.1.1.1,2.2.2.2",
                "http_method": "GET",
                "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
                "port": "80",
                "hostname": "hostname.net"
            }
        }
    }
}
```

신뢰 컨텍스트가 생략된 액세스 권한 부여

```
{
    "activity_name": "Access Grant",
    "activity_id": "1",
    "actor": {
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
        }],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "invoked_by": "",
        "process": {},
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        },
        "session": {}
    },
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Access Activity",
```

```
    "class_uid": "3006",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
```

```
        "ip": "172.24.57.68",
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_detail": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300601",
    "type_name": "Access Activity: Access Grant",
    "data": null
}
```

네트워크 CIDR 엔드포인트를 사용하여 권한 할당

```
{
    "activity_id": "1",
    "activity_name": "Assign Privileges",
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Authorization",
    "class_uid": "3003",
    "data": {
        "endpoint_type": "cidr",
        "protocol": "tcp",
        "access_path": "public",
        "idp": {
            "name": "my-oidc-instance",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
        }],
        "context": {
            "oidc": {
                "family_name": "Last",
                "zoneinfo": "America/Los_Angeles",
                "exp": 1670631145,
                "middle_name": "Middle",
                "given_name": "First",
```

```
                "email_verified": true,
                "name": "Test User Display",
                "updated_at": 1666305953,
                "preferred_username": "johndoe-user@test.com",
                "profile": "http://www.example.com",
                "locale": "US",
                "nickname": "Tester",
                "email": "johndoe-user@test.com",
                "additional_user_context": {
                    "aud": "xxx",
                    "exp": 1000000000,
                    "groups": [
                        "group-id-1",
                        "group-id-2"
                    ],
                    "iat": 1000000000,
                    "iss": "https://oidc-tp.com/",
                    "sub": "xyzsubject",
                    "ver": "1.0"
                }
            },
            "tcp_flow": {
                "destination_ip": "10.0.0.1",
                "destination_port": 22,
                "client_ip": "10.2.7.68"
            }
        }
    },
    "device": {
        "ip":  "10.2.7.68",
        "port": 1002,
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "metadata": {
        "uid": "",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
```

```
        }
    },
    "severity": "Informational",
    "severity_id": "1",
    "start_time": "1668580194340",
    "status_code": "200",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300301",
    "type_name": "Authorization: Assign Privileges",
    "count": 1,
    "dst_endpoint": {
        "ip": "107.22.231.155",
        "port": 22
    },
    "privileges": [
        "vae-12345cbce2EXAMPLE"
    ],
    "user": {
        "email_addr": "johndoe-user@test.com",
        "uid": "johndoe-user",
        "uuid": "9bcce02a-fc15-4091-a0b7-874d157c67b8"
    }
}
```


를 사용하여 Verified Access API 호출 로깅 AWS CloudTrail

AWS Verified·Access는 Verified·Access AWS 서비스 에서 사용자, 역할 또는가 수행한 작업에 대한 레코드를 제공하는 AWS CloudTrail서비스와 통합됩니다. CloudTrail은 Verified Access에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 Verified·Access 콘솔로부터의 호출과 Verified·Access API 작업에 대한 코드 호출이 포함됩니다. CloudTrail에서 수집한 정보를 사용하여 Verified Access에 대한 요청, 요청이 수행된 IP 주소, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트 사용자로 했는지 사용자 보안 인증으로 했는지 여부.
- IAM Identity Center 사용자를 대신하여 요청이 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.

- 다른 AWS 서비스에서 요청했는지 여부.

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화되며 CloudTrail 이벤트 기록에 자동으로 액세스할 수 있습니다. CloudTrail 이벤트 기록은 지난 90일 간 AWS 리전의 관리 이벤트에 대해 보기, 검색 및 다운로드가 가능하고, 수정이 불가능한 레코드를 제공합니다. 자세한 설명은 AWS CloudTrail 사용 설명서의 CloudTrail 이벤트 기록 작업을 참조하세요. Event history(이벤트 기록) 보기는 CloudTrail 요금이 부과되지 않습니다.

AWS 계정 지난 90일 동안의 이벤트를 지속적으로 기록하려면 추적 또는 CloudTrail Lake 이벤트 데이터 스토어를 생성합니다.

CloudTrail 추적

CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 를 사용하여 생성된 모든 추적 AWS Management Console 은 다중 리전입니다. AWS CLI를 사용하여 단일 리전 또는 다중 리전 추적을 생성할 수 있습니다. 계정 AWS 리전 의 모든에서 활동을 캡처하므로 다중 리전 추적을 생성하는 것이 좋습니다. 단일 리전 추적을 생성하는 경우 추적의 AWS 리전에 로깅된 이벤트만 볼 수 있습니다. 추적에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 Creating a trail for your AWS 계정 및 Creating a trail for an organization을 참조하세요.

CloudTrail에서 추적을 생성하여 진행 중인 관리 이벤트의 사본 하나를 Amazon S3 버킷으로 무료로 전송할 수는 있지만, Amazon S3 스토리지 요금이 부과됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요. Amazon S3 요금에 대한 자세한 내용은 Amazon S3 요금을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어

CloudTrail Lake를 사용하면 이벤트에 대해 SQL 기반 쿼리를 실행할 수 있습니다. CloudTrail Lake는 행 기반 JSON 형식의 기존 이벤트를 Apache ORC 형식으로 변환합니다. ORC는 빠른 데이터 검색에 최적화된 열 기반 스토리지 형식입니다. 이벤트는 이벤트 데이터 스토어로 집계되며, 이벤트 데이터 스토어는 고급 이벤트 선택기를 적용하여 선택한 기준을 기반으로 하는 변경 불가능한 이벤트 컬렉션입니다. 이벤트 데이터 스토어에 적용하는 선택기는 어떤 이벤트가 지속되고 쿼리할 수 있는지 제어합니다. CloudTrail Lake에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 AWS CloudTrail Lake 작업을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어 및 쿼리에는 비용이 발생합니다. 이벤트 데이터 스토어를 생성할 때 이벤트 데이터 스토어에 사용할 요금 옵션을 선택합니다. 요금 옵션에 따라 이벤트 모으기 및 저장 비용과 이벤트 데이터 스토어의 기본 및 최대 보존 기간이 결정됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요.

Verified Access 관리 이벤트

관리 이벤트는의 리소스에서 수행되는 관리 작업에 대한 정보를 제공합니다 AWS 계정. 이를 컨트롤 플레인 작업이라고도 합니다. 기본적으로 CloudTrail은 관리 이벤트를 로깅합니다.

Verified Access는 컨트롤 플레인 작업을 관리 이벤트로 로깅합니다. 관련 목록은 Amazon EC2 API 참조를 참조하세요.

Verified Access 이벤트 예제

다음은 `CreateVerifiedAccessInstance` 작업을 설명하는 CloudTrail 이벤트를 보여 주는 예제입니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIKK4OOINJWEXAMPLE:jdoe",
        "arn": "arn:aws:iam::123456789012:user/jdoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "jdoe"
    },
    "eventTime": "2022-11-18T20:44:04Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "CreateVerifiedAccessInstance",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "CreateVerifiedAccessInstanceRequest": {
            "Description": "",
            "ClientToken": "85893b1e-49f6-4d24-97de-280c664edf1b"
        }
    },
    "responseElements": {
        "CreateVerifiedAccessInstanceResponse": {
            "verifiedAccessInstance": {
                "creationTime": "2022-11-18T20:44:04",
                "description": "",
                "verifiedAccessInstanceId": "vai-0d79d91875542c549",
                "verifiedAccessTrustProviderSet": ""
            },
```

```
            "requestId": "2eae195d-6bfd-46d7-b46e-a68cb791de09"
        }
    },
    "requestID": "2eae195d-6bfd-46d7-b46e-a68cb791de09",
    "eventID": "297d6529-1144-40f6-abf8-3a76f18d88f0",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

CloudTrail 레코드 콘텐츠에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail record contents를 참조하세요.

에 대한 할당량 AWS Verified Access

AWS 계정 에는 각에 대해 이전에 제한이라고 하는 기본 할당량이 있습니다 AWS 서비스. 다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다.

AWS 계정-레벨 할당량

AWS 계정 에는 Verified Access와 관련된 다음과 같은 할당량이 있습니다.

명칭	기본값	조정 가능	설명
Verified·Access 인스턴스	5	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 인스턴스의 최대 수입니다.
Verified·Access 그룹	10	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 그룹의 최대 수입니다.
Verified·Access 신뢰 공급자	15	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 신뢰 공급자의 최대 수입니다.
Verified·Access 엔드포인트	50	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 엔드포인트의 최대 수입니다.

HTTP 헤더

HTTP 헤더에는 다음과 같이 크기 제한이 있습니다.

명칭	기본값	조정 가능
요청 라인	16K	아니요
단일 헤더	16K	아니요
전체 응답 헤더	32K	아니요

명칭	기본값	조정 가능
전체 요청 헤더	64K	아니요

HTTP 트래픽

연결 유휴 제한 시간은 60초입니다. 애플리케이션이 HTTP 요청에 응답하는 데 60초 이상 걸리는 경우 클라이언트는 HTTP 504 게이트웨이 제한 시간 오류를 수신합니다. Verified Access 로그가 활성화된 경우 HTTP 504 오류를 로깅합니다.

OIDC 클레임 크기

다음은 OIDC 클레임 크기 제한입니다.

명칭	기본값	조정 가능
OIDC 클레임 크기	11K	아니요

IAM Identity Center

Verified Access는 최대 1,000개의 그룹에 할당된 IAM Identity Center의 사용자에게 액세스 권한을 제공할 수 있습니다.

Verified·Access 사용 설명서에 대한 문서 이력

다음 표에서는 Verified·Access에 대한 문서 릴리스를 소개합니다.

변경 사항	설명	날짜
신뢰 컨텍스트에서 액세스 토큰 지원	OIDC 사용자 클레임`additional_user_context` 에를 추가하도록 업데이트합니다.	2025년 2월 24일
비 HTTP 프로토콜을 통한 리소스 지원	HTTP가 아닌 프로토콜을 통해 리소스에 대한 액세스 권한 릴리스.	2025년 2월 5일
미리 보기 릴리스	HTTP가 아닌 프로토콜을 통해 리소스에 대한 액세스 릴리스를 미리 봅니다.	2024년 12월 1일
AWS 관리형 정책 업데이트	Verified·Access에 대한 AWS 관리형 IAM 정책을 업데이트했습니다.	2023년 11월 17일
유휴 시(저장된) 데이터 암호화	AWS Verified Access는 기본적으로 AWS 소유 KMS 키를 사용하여 저장 데이터를 암호화합니다.	2023년 9월 28일
FIPS 규정 준수 지원	FIPS 규정 준수에 대한 Verified·Access를 구성합니다.	2023년 9월 26일
향상된 로깅	로그에 신뢰 컨텍스트를 추가하는 로깅 기능 추가.	2023년 6월 19일
AWS 관리형 정책 업데이트	Verified·Access에 대한 AWS 관리형 IAM 정책을 업데이트했습니다.	2023년 5월 31일

GA 릴리스	Verified·Access 사용 설명서의 GA 릴리스. AWS WAF 통합을 포함합니다.	2023년 4월 27일
미리 보기 릴리스	Verified·Access 사용 안내서 미리 보기 릴리스	2022년 11월 29일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.