



AWS한국사용자모임
#Beginner
소모임



AWS한국사용자모임
#Beginner
소모임

IAM IDENTITY CENTER 도입

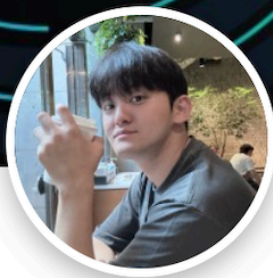
팀 간 경계를 허무는 효율적이고 안전한 개발환경 제공하기

이한섭 | AWSKRUG

발표자 소개



AWS한국사용자모임
#Beginner
소모임



hanseob lee

DevOps Engineer | AWS Community Builder | AWSKRUG
Organizer

대한민국 서울 · [연락처](#)



밀리의 서재

DevOps Engineer @ 백엔드개발본부 DevOps팀

“시스템의 수명은 **observability**와 함께한다.”

0. 목차



AWS한국사용자모임
#Beginner
소모임

- 배경지식 (IAM Organization, IAM Identity Center)
- IAM Identity Center 훑아보기
- 기존의 접근 방식
- 변경된 접근 방식
- 도입 효과
- 앞으로의 과제
- 후기

오늘 다루는 이야기



AWS한국사용자모임
#Beginner
소모임

- IAM Identity Center서비스 훑아보기 ○
- 이전까지의 환경과 놓여진 상황에 대해 ○
- 데브옵스의 수 많은 역할 중 일부에 대해 ○
- 보안적인 혹은 거버넌스적인 내용에 대해 ✖
- 상세 서비스 아키텍처에 대해 ✖





AWS한국사용자모임
#Beginner
소모임

배경 지식

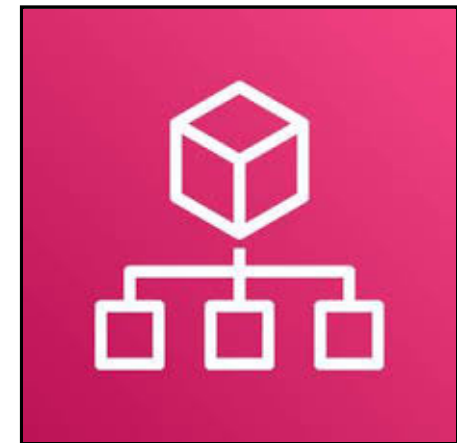
- AWS organization
- AWS IAM identity Center (SSO)

배경지식 (AWS Organizations)



AWS한국사용자모임
#Beginner
소모임

- Multi Account Centralize Control (OU)
- Protect the environment to ensure compliance



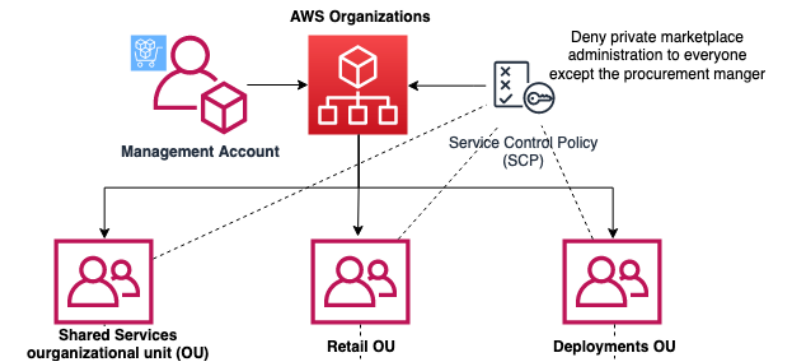
AWS Organizations

배경지식 (AWS Organizations)



AWS한국사용자모임
#Beginner
소모임

- Multi Account Centralize Control (OU)
- Protect the environment to ensure compliance



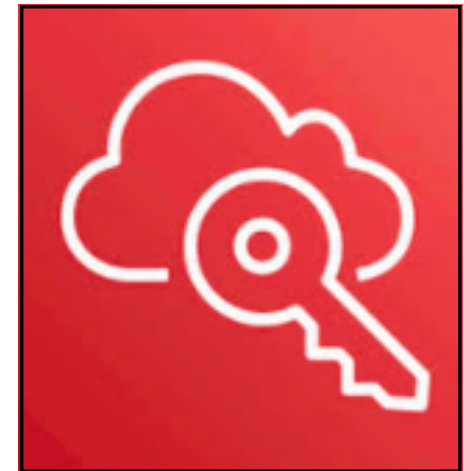
"계정을 그룹화하여 단일 단위로 관리할 수 있음"

배경지식 (IAM Identity Center)



AWS한국사용자모임
#Beginner
소모임

- Single Sign-On, SSO
- Centralized User Management
- Integration with External Identity Providers
- Permission Sets for Fine-Grained Access Control



AWS Identity Center (SSO)

“사용자 액세스를 단순화하는 단일 통합 지점”



AWS한국사용자모임
#Beginner
소모임

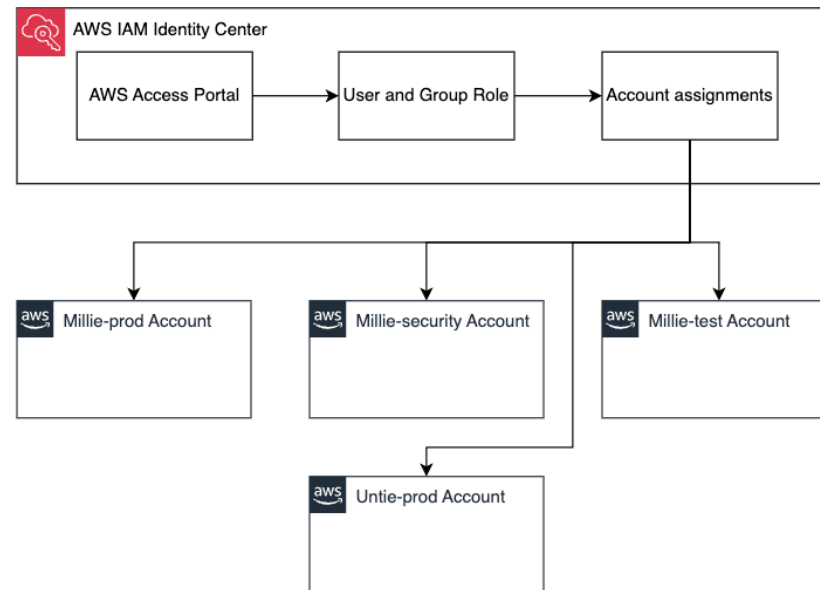
AWS IAM Identity Center 톺아보기

AWS IAM Identity Center 톺아보기



AWS한국사용자모임
#Beginner
소모임

Okta, Google Workspace, Microsoft Entra ID(구 Microsoft Active Directory), 기본 제공 IAM Identity Center 디렉터리 등 선택한 ID 소스를 사용하여 서비스를 구성하고 인력 사용자 및 그룹에 대한 공통된 이해를 바탕으로 모든 AWS 서비스를 제공할 수 있음.

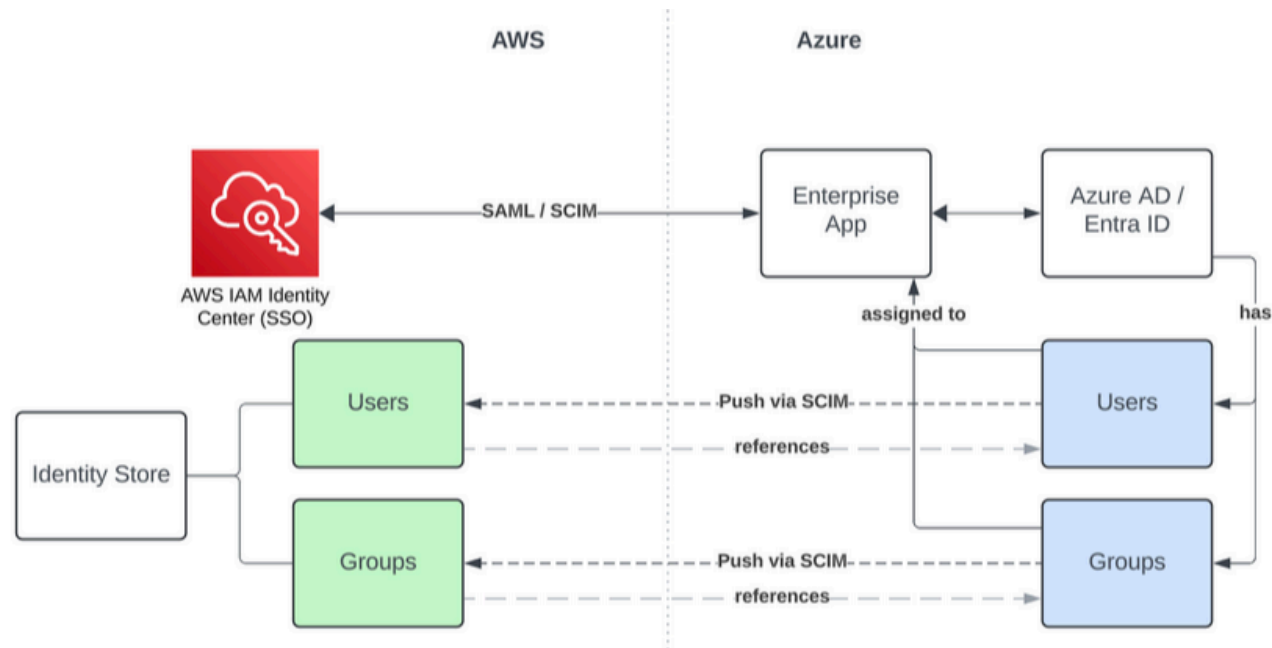


AWS IAM Identity Center 톺아보기



AWS한국사용자모임
#Beginner
소모임

Okta, Google Workspace, Microsoft Entra ID(구 Microsoft Active Directory), 기본 제공 IAM Identity Center 디렉터리 등 선택한 ID 소스를 사용하여 서비스를 구성하고 인력 사용자 및 그룹에 대한 공통된 이해를 바탕으로 모든 AWS 서비스를 제공할 수 있음.





AWS한국사용자모임
#Beginner
소모임

기존의 접근 방식

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

입사 초반

- 제어할 수 없는 Accesskey들의 향연

| 액세스 키 ID | | 활성 키 수명 | 생성 시간 |
|----------|--------------|---------|-------|
| | | | |
| 2년 전 | Active - AKI | 1393일 | 5년 전 |
| 2년 전 | Inactive - A | 1330일 | 5년 전 |
| 1년 전 | Active - AKI | 1148일 | 4년 전 |
| 1년 전 | Active - AKI | 1080일 | 4년 전 |
| 1년 전 | Active - AKI | 898일 | 3년 전 |
| 1년 전 | Active - AKI | 703일 | 3년 전 |
| 1년 전 | Active - AKI | 686일 | 2년 전 |
| 1년 전 | Active - AKI | 480일 | 2년 전 |
| 1년 전 | Active - AKI | 443일 | 2년 전 |
| 1년 전 | Active - AKI | 423일 | 2년 전 |
| 1년 전 | Active - AKI | 384일 | 2년 전 |
| 1년 전 | Active - AKI | 384일 | 2년 전 |

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

제어할 수 없는 Accesskey들의 향연
(서버, 개발자 로컬, 어딘가 ...etc...)



사용 목적과 유저 확인



Accesskey 제거



사용 목적과 유저 확인 -> 불가

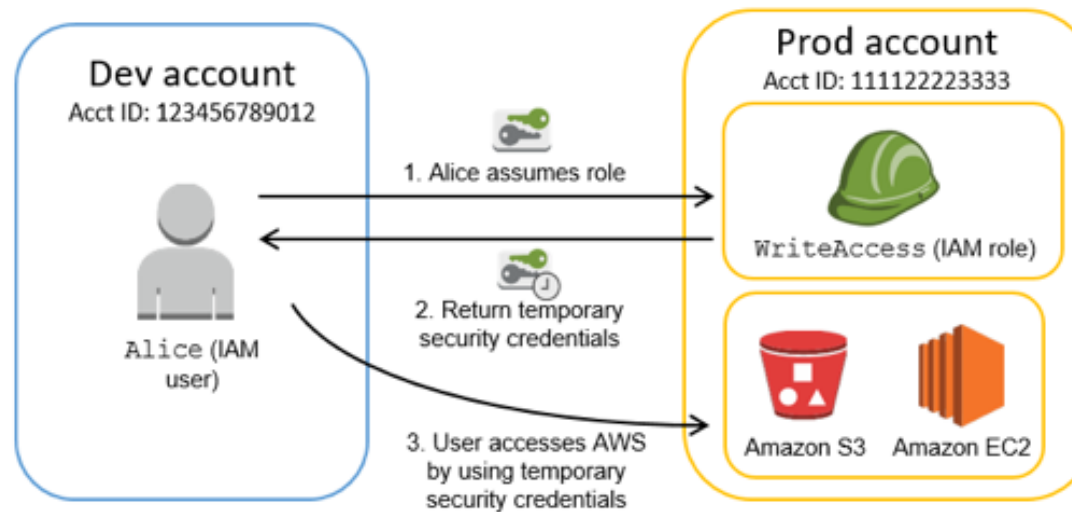
1. 레거시 소스코드 내 statics value로 들어간 키 식별 불가
2. 본인들이 쓰고 있는 키가 어떤 키인지 식별 불가

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..



임시 상승된 액세스는 운영 기능을 방해하지 않고 인적 액세스와 관련된 위험을 줄이는 데 도움을 줌
실 서비스에 사용중인 AWS Accesskey를 점진적 줄여나가기 위한 첫 걸음

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

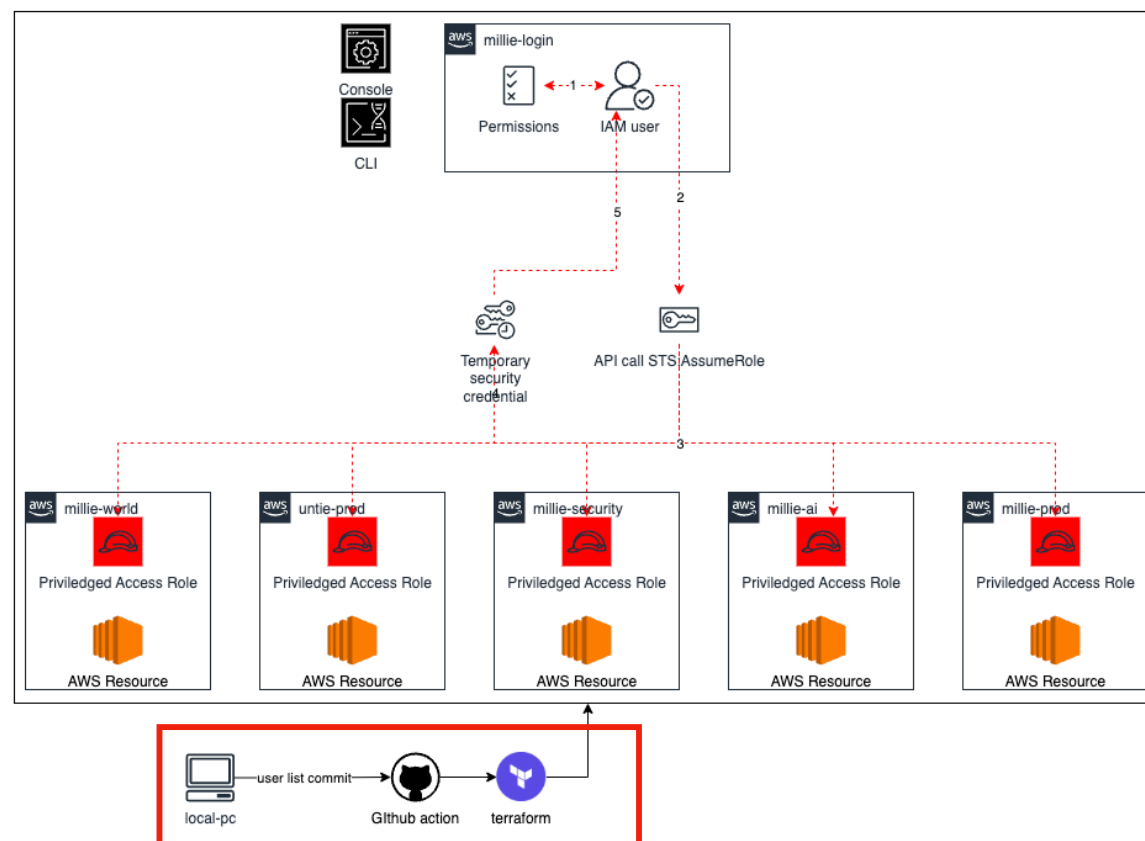
1. 사용자와 서비스 역할을 분리
2. 사용자 access 방식을 assume switch방식으로 변경
3. millie-login이라는 중앙 계정에서 각 계정의 팀 역할로 switching 하여 접근하도록 변경

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..



기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

```
membership = {  
  "DevOps-team" = {  
    user_list = ["A", "B", "C"]  
  }  
  "Backend-Team" = {  
    user_list = ["D", "E", "F"]  
  }  
  :  
  :  
  :  
  "Client-Team" = {  
    user_list = ["H", "I", "J"]  
  }  
}
```

| | |
|--|------|
| ✓ Update membership.tf Terraform #10: Commit e39d527 pushed by Shawn-Millie-DevOps | main |
| add Terraform #9: Commit 73ddd5f pushed by Dominic-Millie-DevOps | main |
| ✓ usergroup 변경 Terraform #8: Commit 886bba5 pushed by Dominic-Millie-DevOps | main |
| ✓ clientTeam mark,lyla,brian add Terraform #7: Commit 6695308 pushed by Dominic-Millie-DevOps | main |
| ✓ membership 그룹정리 Terraform #6: Commit b2a8f90 pushed by Dominic-Millie-DevOps | main |
| ✓ Update membership.tf Terraform #5: Commit 8c7d90f pushed by Dominic-Millie-DevOps | main |
| ✓ Update membership.tf Terraform #4: Commit b86f95b pushed by Dominic-Millie-DevOps | main |
| ✓ Update membership.tf Terraform #3: Commit 088ac37 pushed by Dominic-Millie-DevOps | main |
| ✓ fix: tatum name변경 Terraform #2: Commit 12d83f8 pushed by Dominic-Millie-DevOps | main |

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

- (필수). AWS 계정 생성 요청 결제 상신 링크
- 0. AWS 역할전환이란
- 1.AWS 역할전환의 이점
- 2.AWS 사용자 최초 접속 및 MFA 설정 방법
- 3.AWS 계정 간 전환 사용방법
- [별첨1]AWS 계정 및 역할 리스트
- [별첨2]참고사항

(필수). AWS 계정 생성 요청 결제 상신 링크

0. AWS 역할전환이란

AWS 역할전환이 필요한 이유는 보안 강화, 권한 관리 용이성, 비용 최적화와 간편한 운영을 위함입니다. 또한, 다중 계정 환경에서 특히 유용하게 활용됩니다.

계정 해킹 방어를 위해 로그인 모니터링 및 실 서비스에 사용중인 AWS Accesskey를 점진적 줄여나가기 위한 첫 걸음입니다.

더 자세한 내용은 아래 문서를 참고해주시고, 추가로 구조나 구성에 대한 궁금증이 있으면 입구에서 @dominic 을 찾아주세요.

 IAM의 임시 보안 자격 증명 - AWS Identity and Access Management

 IAM 역할 - AWS Identity and Access Management

 사용자에서 IAM 역할로 전환(콘솔) - AWS Identity and Access Management

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

1. 모든 서비스의 코드에서 Accesskey를 도려내는 작업을 동시에 진행
 - Boto3.client, resource, session 함수들에서 accesskey를 서버 profile role 을 가져가도록 점진적 제거 작전에 돌입
2. Managed Policy를 Custom하는 작업 + 로직이 사용하는 AWS 서비스 권한 추적
 - IAM의 액세스 관리자 기능을 유용하게 사용하였음

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

| 서비스 | 권한 부여 정책 | 마지막 액세스 날짜 |
|------------------------------------|----------|-----------------|
| Amazon S3 | | 어제 |
| AWS CloudFormation | | 추적 기간에 액세스되지 않음 |
| Amazon S3 Object Lambda | | 추적 기간에 액세스되지 않음 |

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

액세스 키를 점진적 제거

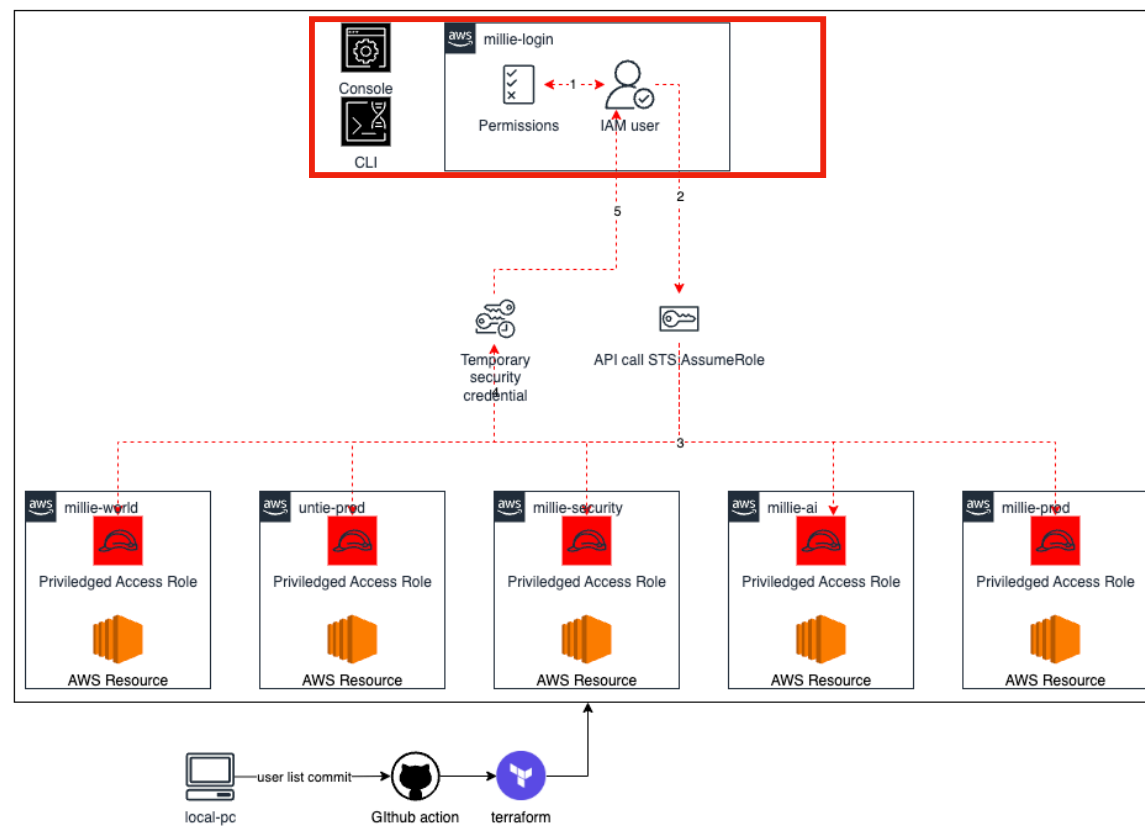
새로운 로컬 개발환경 내 키 발급방식을 변경

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..



기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

```
(my-venv-3.9) [redacted]Mac millie-infra % python assume_role.py tom
Available accounts:
1. [redacted]ie-login
2. [redacted]ie-prod
3. [redacted]e-prod
Choose an account (enter the number): 2
Available roles:
1. [redacted]
2. [redacted]
Choose a role (enter the number): 2
arn:aws:iam::2[redacted]:role/millie-prod@[redacted]
tom millie-prod@[redacted]n:aws:iam::[redacted]:mfa/Phone
Enter your MFA token: [redacted]
```

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

하여 23년에는..

AWS Assume Role with MFA

AWS TEAM List

TEAM List: DevOps

AWS Account List

Account List:

Insert MFA Serial

MFA Serial Number: *****

Sign in

기존의 접근 방식



AWS한국사용자모임
#Beginner
소모임

1 년간 운영하면서 또 다른 문제들이 감지됨

필요없는 혹은 과한 권한들을 제거하고 커스텀하기엔 문제를 확인하고 조치하기에 효율이 떨어지는 구조

ex. 권한에 문제가 발생했을 때 확인하는 절차

- Millie-login 내 그룹에 할당된 권한이 적절한가
- switch role를 사용할 권한이나 그룹을 제대로 할당하였는가
- switch role을 통해 접속한 임시 권한에 적절한 권한이 발급되어있는가
 - 토큰이 만료되지 않았는가
 - ip 범위 내 접근하였는가



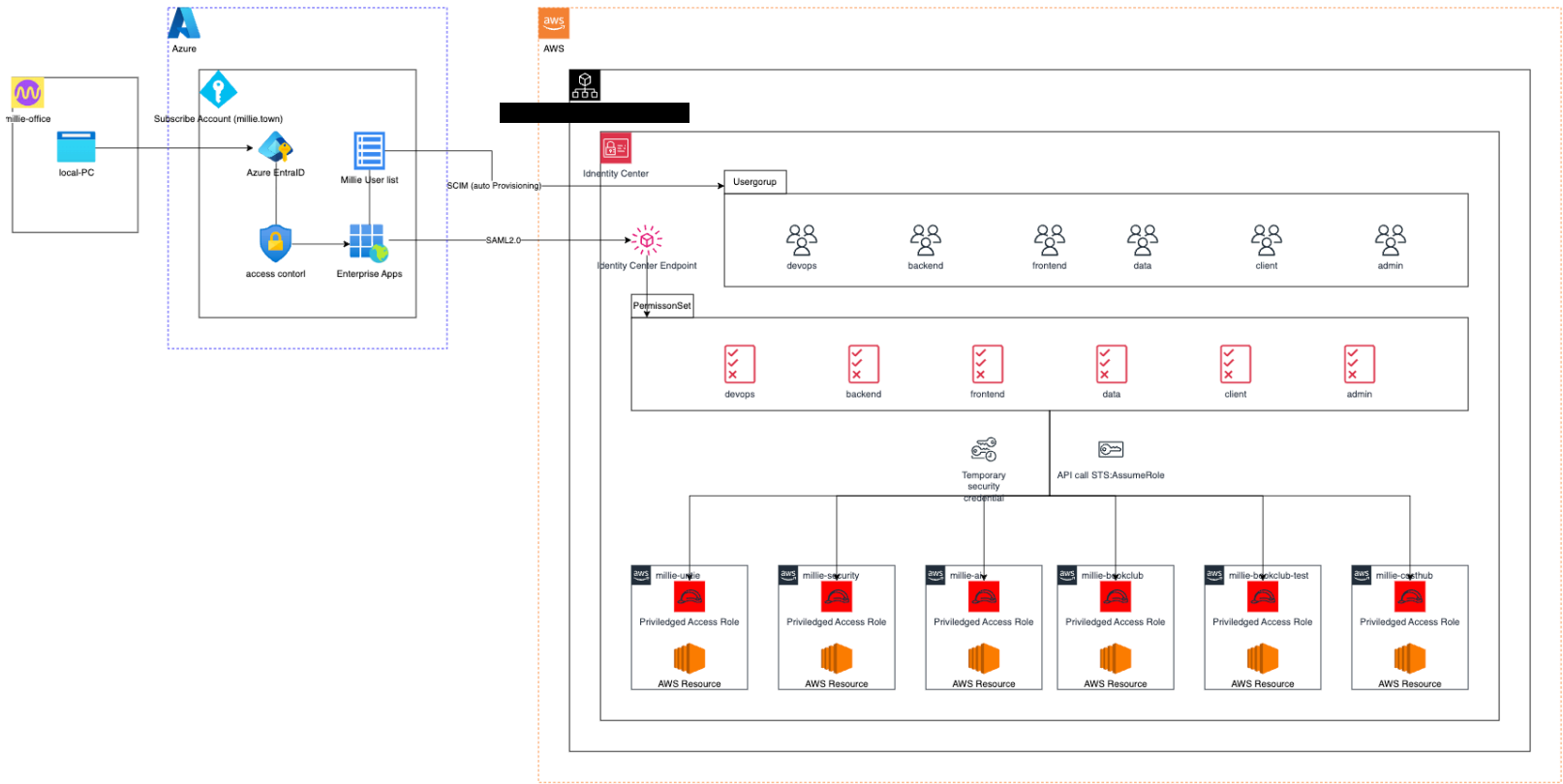
AWS한국사용자모임
#Beginner
소모임

변경된 접근 방식

변경된 접근 방식



AWS한국사용자모임
#Beginner
소모임



변경된 접근 방식



AWS한국사용자모임
#Beginner
소모임

1. 계정 관리 부분의 변경 사항

millie-login 내 iam 유저 생성 → millie 전사에서 사용중인 M365의 계정을 saml 2.0로 연결 (Azure AD)
신규 입사자 발생 시, 따로 유저를 추가하지 않아도 자동으로 프로비저닝되도록 연결 (개발조직 한정)

2. 권한 제어 부분의 변경 사항

중앙 계정에서 모든 계정의 권한을 제어 관리하고 감독할 수 있음

정적 할당된 액세스 권한이 아닌 JIT(Just In Time) 액세스를 통해 키의 수명주기를 조절하고 제어할 수 있음

3. 별도의 작업없이 임시토큰을 발급받은 액세스키를 사용자가 직접 할당 받을 수 있음 (중요)

변경된 접근 방식



AWS한국사용자모임
#Beginner
소모임

AWS access portal

Accounts Applications

AWS accounts (6) [Create shortcut](#)

Filter accounts by name, ID, or email address

- ▶ millie- [redacted] lie.town
- ▶ millie- [redacted] 626211110529 | login@millie.town
- ▼ Millie- [redacted] lie.town
 - RegionalAdmin | [Access keys](#)
 - backend | [Access keys](#)
 - client | [Access keys](#)
 - devops | [Access keys](#)
 - frontend | [Access keys](#)
- ▶ millie- [redacted] @millie.town
- ▶ mil [redacted] town
- ▶ vor [redacted] millie.town

변경된 접근 방식



AWS한국사용자모임
#Beginner
소모임

macOS and Linux | Windows | PowerShell

▼ AWS IAM Identity Center credentials (Recommended)

To extend the duration of your credentials, we recommend you configure the AWS CLI to retrieve them automatically using the `aws configure sso` command. [Learn more](#)

SSO start URL

SSO Region

▼ Option 1: Set AWS environment variables

Run the following commands in your terminal to set the AWS environment variables. [Learn more](#)

```
export AWS_ACCESS_KEY_ID="ASIAXXXXXXXXXXXXXX"
export AWS_SECRET_ACCESS_KEY="Yy/pRYNdXXXXXXXXXXXX"
export AWS_SESSION_TOKEN="IQoJb3JpZ2luZXN0XXXXXXXXXXXX"
```

Copy

▼ Option 2: Add a profile to your AWS credentials file

Copy and paste the following text in your AWS credentials file (`~/.aws/credentials`). [Learn more](#)

```
[269315701127_backend]
aws_access_key_id = ASIAXXXXXXXXXXXXXX
aws_secret_access_key = Yy/pRYNdXXXXXXXXXXXX
aws_session_token = IQoJb3JpZ2luZXN0XXXXXXXXXXXX
```

Copy

▼ Option 3: Use individual values in your AWS service client

Copy and paste these values into your code. [Learn more](#)

AWS access key ID


AWS secret access key


AWS session token

변경된 접근 방식



AWS한국사용자모임
#Beginner
소모임

**eyes-system** 앱 9월 11일 오후 1:35


 AWS Login 감지

로그인 상태:
Success




유저정보:
dean@millie.town

접속환경:
millie_bookclub


접근 IP:
[REDACTED]

 외부망 접근으로 AWS API차단

로그인 시간 : 2024-09-11 13:35:21 powered by : devops팀

 1  1 

1개의 댓글

**Dominic(DEVOPS)** 9월 11일 오후 1:36

이사람 연행해가세요



AWS한국사용자모임
#Beginner
소모임

도입 효과



1. **Accesskey** 관리 추적을 하지 않아도 되는 구조 (8시간 만료)
2. 개발자의 **access** 퍼널을 효과적으로 단축시킴
3. 중앙 집중식 사용자 관리와 접근제어를 통해 역할전환방식의 비효율성을 극복



AWS한국사용자모임
#Beginner
소모임

앞으로의 과제



1. Temporary Elevated Access Management 확장



사용자가 필요한 권한을 직접 요청하는 셀프서비스 (self-service) 구조

임시 권한 상승 구조 구현



AWS한국사용자모임
#Beginner
소모임

후기



1. DevOps 문화에 가치는 결국 무언인가
2. 신뢰 쌓기
3. 자동화라고 우기지말자
4. 커뮤니케이션 비용에 대해



AWS한국사용자모임
#Beginner
소모임

감사합니다