



SEOUL | MARCH 20, 2024

AWS Unicorn Day

Blux의 AI SaaS 솔루션 구축 사례

Sunhong Min (he/him)

MLOps Engineer

Blux

Agenda

Introducing Blux

Technological Considerations for Building Blux's SaaS Solution

How were the Technological Considerations Addressed?

The Results of Blux's SaaS Solution Enhancement



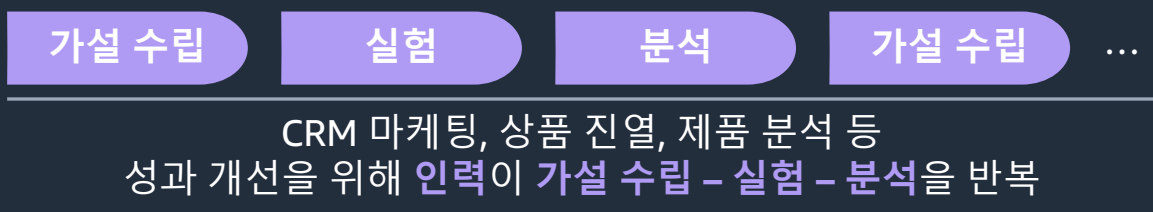


Go beyond your potential.

AI를 통해 온라인 비즈니스의 고객 가치를 극대화합니다.

Introducing Blux

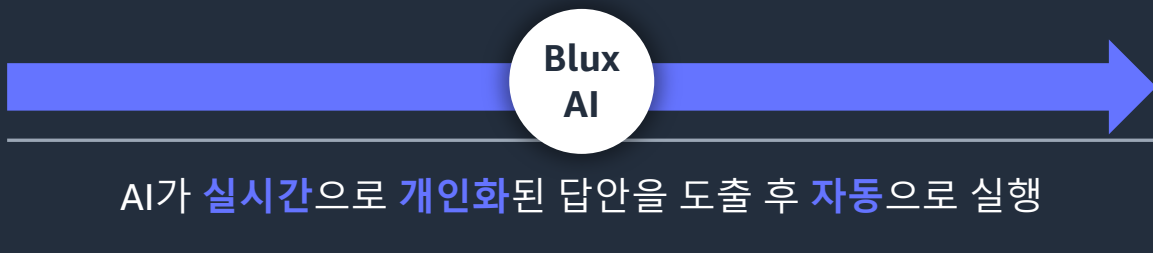
기존 앱/웹 서비스의 '가설 기반 성과 개선'



가설 수립 - 실험 - 분석을 무한 반복하기 위해
막대한 인적 리소스 사용

전문 인력이 요구되며 성과가 인력의 전문성에 의존적

Blux가 제시하는 '성과 개선의 혁신'



인적 리소스 투입 최소화

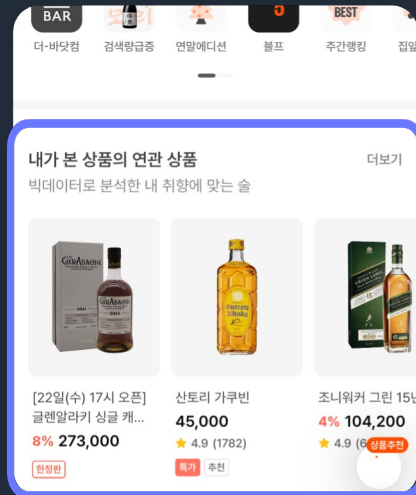
기존 방식 대비 2 ~ 7배의 성과 개선

Introducing Blux

BLUX의 상품 추천 엔진을 도입한 테넌트의 앱/웹 화면 예시

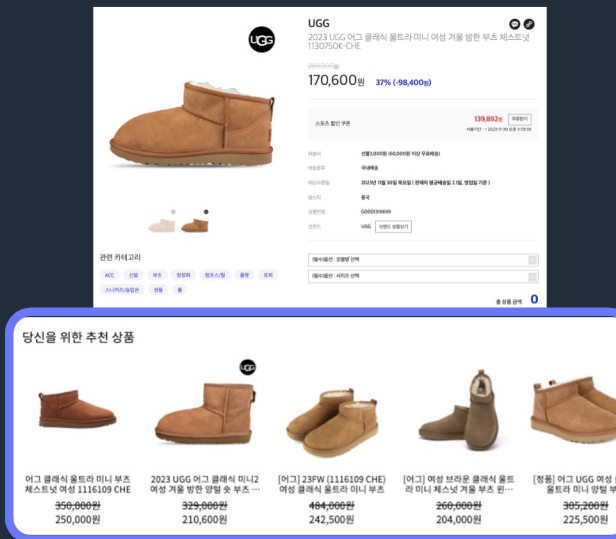
홈화면

구매할 확률이 가장 높은 상품
추천을 통해 구매 전환에 기여



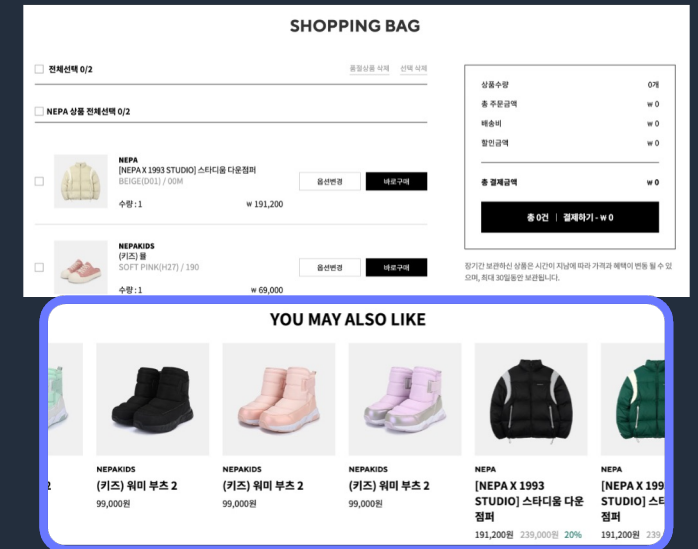
상품 상세 페이지

비슷한 상품 노출을 통해
지속적인 탐색 유도



장바구니

같이 살만한 상품 노출을 통해
Upselling & Cross-selling 유도



Introducing Blux

BLUX CRM 제품 소개

유저 세그멘테이션

꼭 보내야 할 사람에게만 보내는 카카오 친구톡, MMS



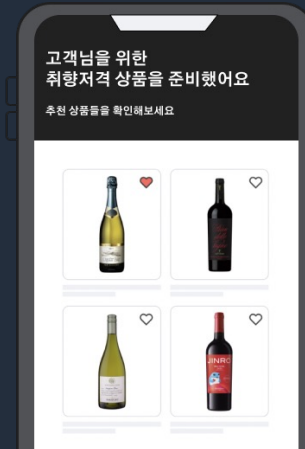
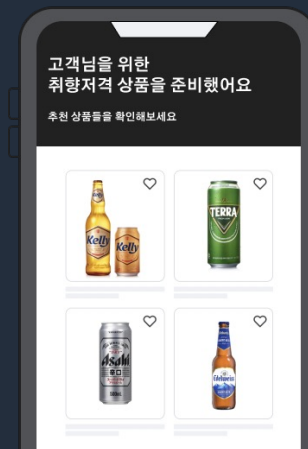
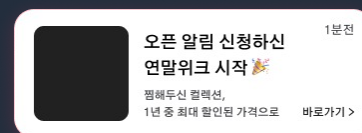
A브랜드 상품에
관심 있는 유저군



이탈 위험 유저군에
할인 쿠폰 발행

맞춤형 상품 제안

하나의 푸시 알림으로 개인별로 맞춤화된 상품 제안



Introducing Blux

여러 테넌트의 사례로 입증된 BLUX 기술의 비즈니스 임팩트



Introducing Blux

여러 테넌트의 사례로 입증된 BLUX 기술의 비즈니스 임팩트



Introducing Blux

여러 테넌트의 사례로 입증된 BLUX 기술의 비즈니스 임팩트



Introducing Blux

여러 테넌트의 사례로 입증된 BLUX 기술의 비즈니스 임팩트



Tech Considerations for Building Blux's SaaS Solution

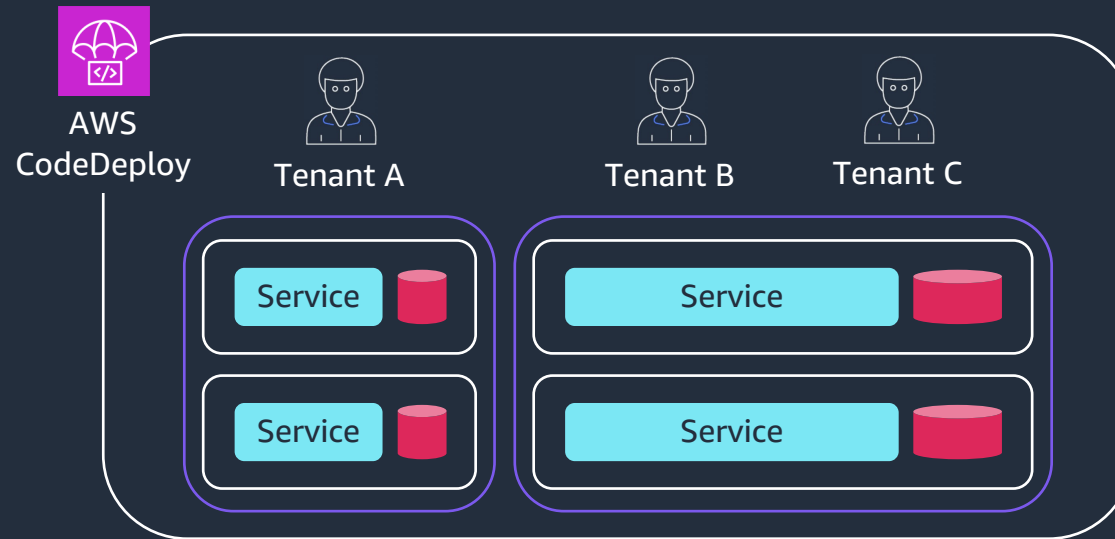
BLUX의 SAAS 솔루션 구축 시 기술 고려사항

1. 모든 테넌트를 일괄적으로 관리할 수 있는 공통된 운영 환경

Tech Considerations for Building Blux's SaaS Solution

SaaS 기업이 공통된 운영 환경을 갖췄을 때의 장점

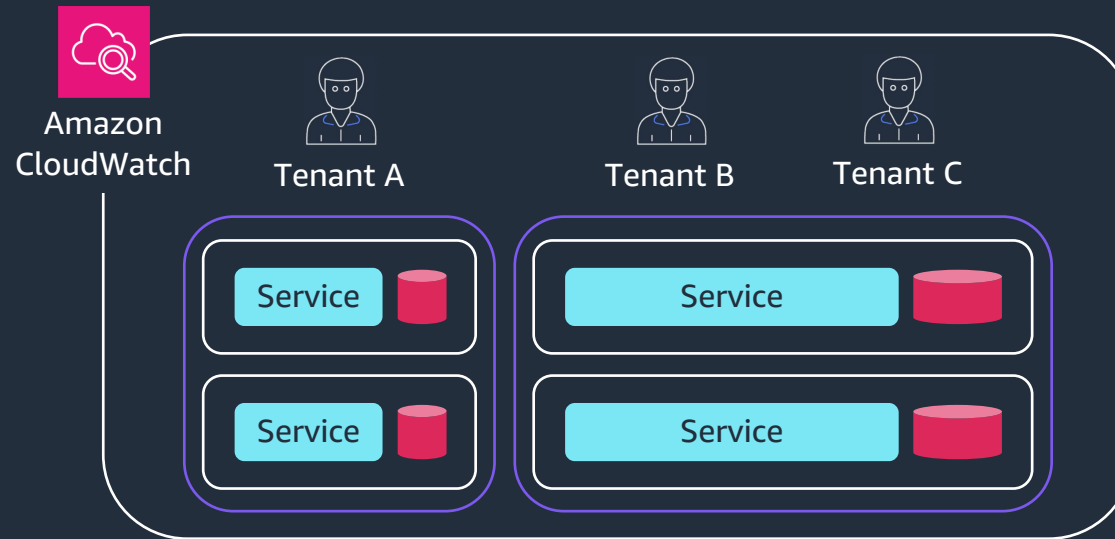
- 단일 공유 프로세스를 통해 모든 테넌트에게 새 기능을 배포할 수 있습니다.



Tech Considerations for Building Blux's SaaS Solution

SaaS 기업이 공통된 운영 환경을 갖췄을 때의 장점

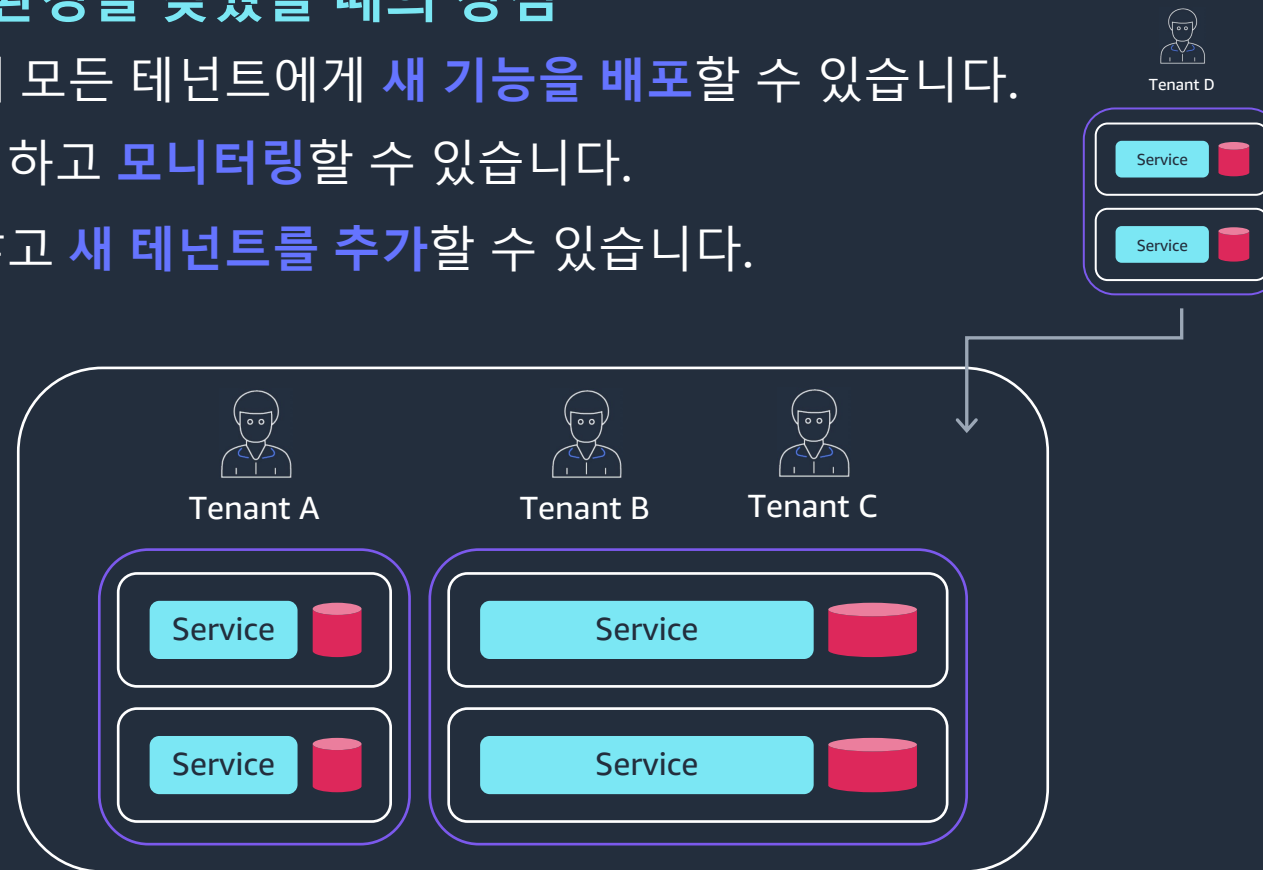
- 단일 공유 프로세스를 통해 모든 테넌트에게 새 기능을 배포할 수 있습니다.
- 모든 테넌트를 한눈에 관리하고 모니터링할 수 있습니다.



Tech Considerations for Building Blux's SaaS Solution

SaaS 기업이 공통된 운영 환경을 갖췄을 때의 장점

- 단일 공유 프로세스를 통해 모든 테넌트에게 **새 기능을 배포**할 수 있습니다.
- 모든 테넌트를 한눈에 관리하고 **모니터링**할 수 있습니다.
- 운영 오버헤드를 늘리지 않고 **새 테넌트를 추가**할 수 있습니다.



Tech Considerations for Building Blux's SaaS Solution

BLUX의 SAAS 솔루션 구축 시 기술 고려사항

1. 모든 테넌트를 일괄적으로 관리할 수 있는 공통된 운영 환경
2. 민첩하고 자동화된 온보딩 프로세스

Tech Considerations for Building Blux's SaaS Solution

온보딩이 중요한 이유

- 온보딩이 **첫인상**을 결정합니다.
- 너무 복잡하면 쓰고 싶은 마음도 사라집니다.
- 온보딩 경험이 좋으면 **제품에 대한 믿음**도 커집니다.

Tech Considerations for Building Blux's SaaS Solution

온보딩이 중요한 이유 (내부편)

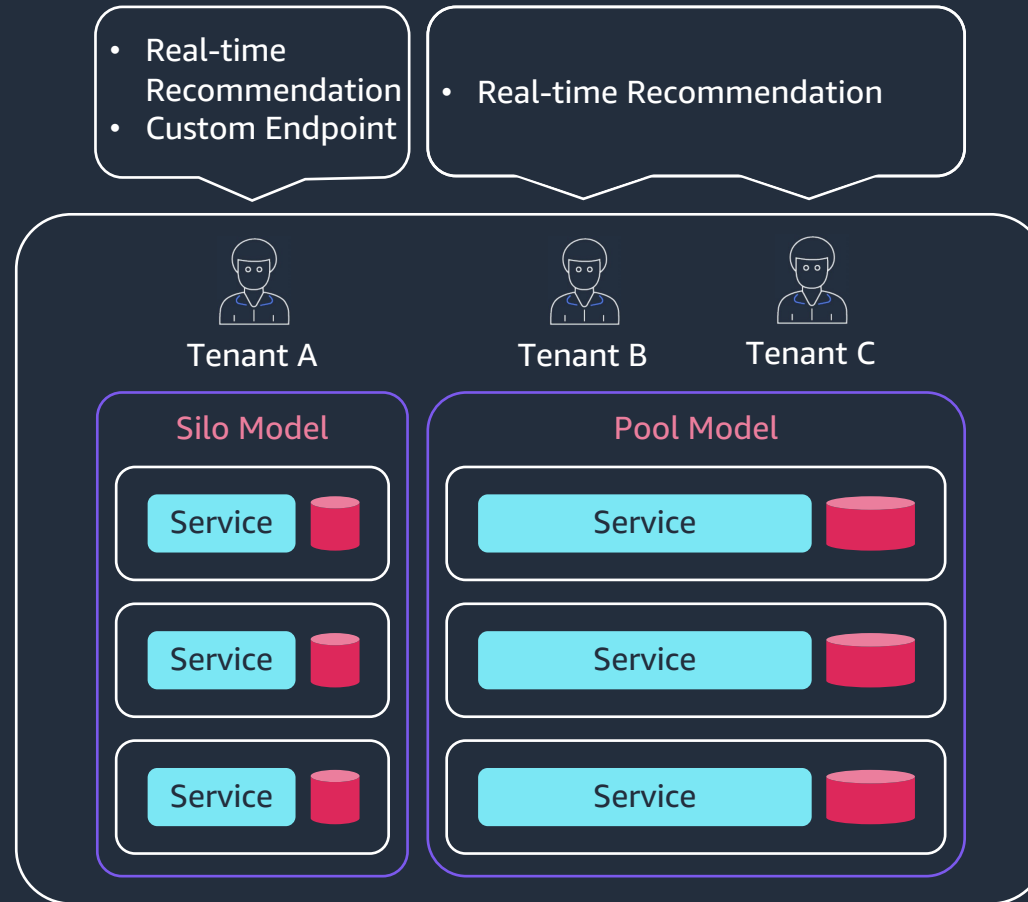
- 불필요한 CS 비용을 줄일 수 있습니다.
- 반복적인 작업을 자동화함으로써 실수를 줄이고 시간을 절약할 수 있습니다.

Tech Considerations for Building Blux's SaaS Solution

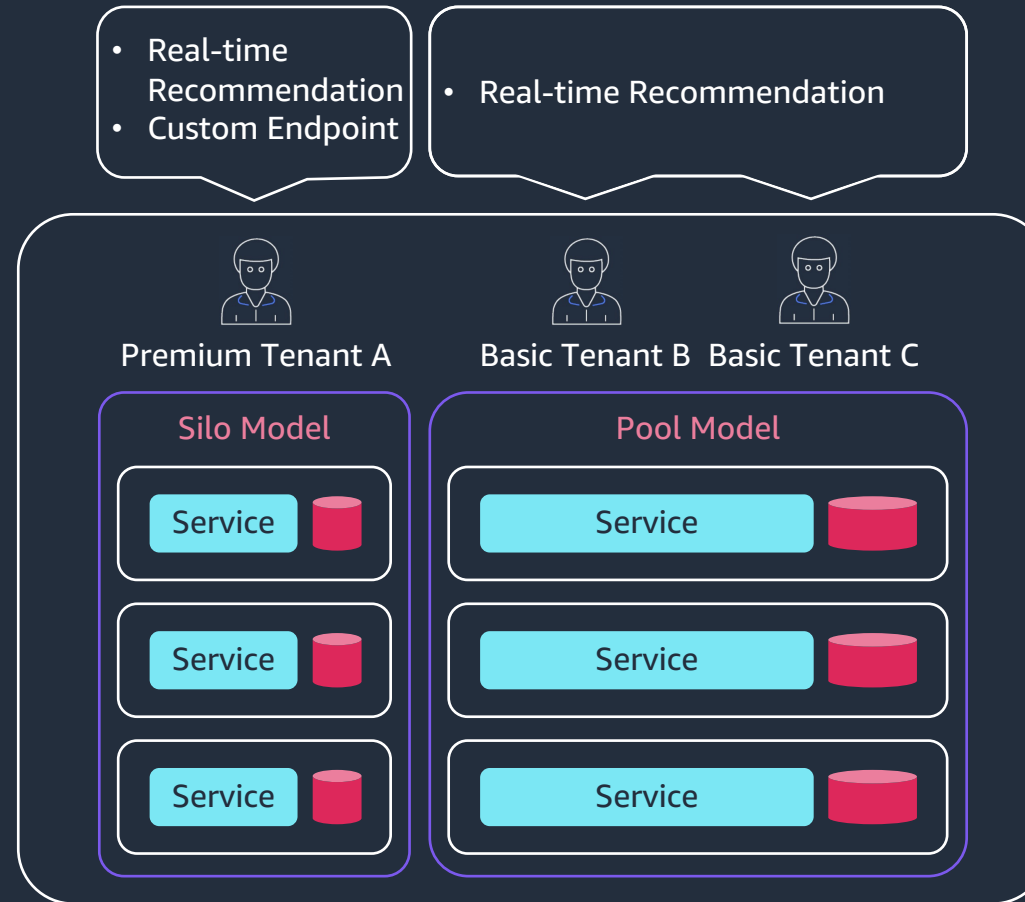
BLUX의 SAAS 솔루션 구축 시 기술 고려사항

1. 모든 테넌트를 일괄적으로 관리할 수 있는 공통된 운영 환경
2. 민첩하고 자동화된 온보딩 프로세스
3. 다중 티어 시스템 및 티어에 따른 배포 모델

Tech Considerations for Building Blux's SaaS Solution



Tech Considerations for Building Blux's SaaS Solution



Tech Considerations for Building Blux's SaaS Solution

BLUX의 SAAS 솔루션 구축 시 기술 고려사항

1. 모든 테넌트를 일괄적으로 관리할 수 있는 공통된 운영 환경
2. 민첩하고 자동화된 온보딩 프로세스
3. 다중 티어 시스템 및 티어에 따른 배포 모델
4. 테넌트 자원 격리

Tech Considerations for Building Blux's SaaS Solution

테넌트 격리란? *

- 자원에 대한 액세스를 엄격하게 제어하여, **다른 테넌트의 자원에 액세스하려는 시도를 차단**하는 메커니즘
- 테넌트를 멀티 테넌트 모델로 전환시킬수록, 테넌트는 한 테넌트가 다른 테넌트의 자원에 액세스할 가능성에 대해 더 우려하게 됩니다.
- 멀티 테넌트 모델이 공유 인프라에서 실행되는 경우에도, **각 테넌트의 자원은 격리**되어야 합니다.

* AWS 백서 <SaaS 아키텍처 기초> 중 **테넌트 격리**: https://docs.aws.amazon.com/ko_kr/whitepapers/latest/saas-architecture-fundamentals/tenant-isolation.html

Tech Considerations for Building Blux's SaaS Solution

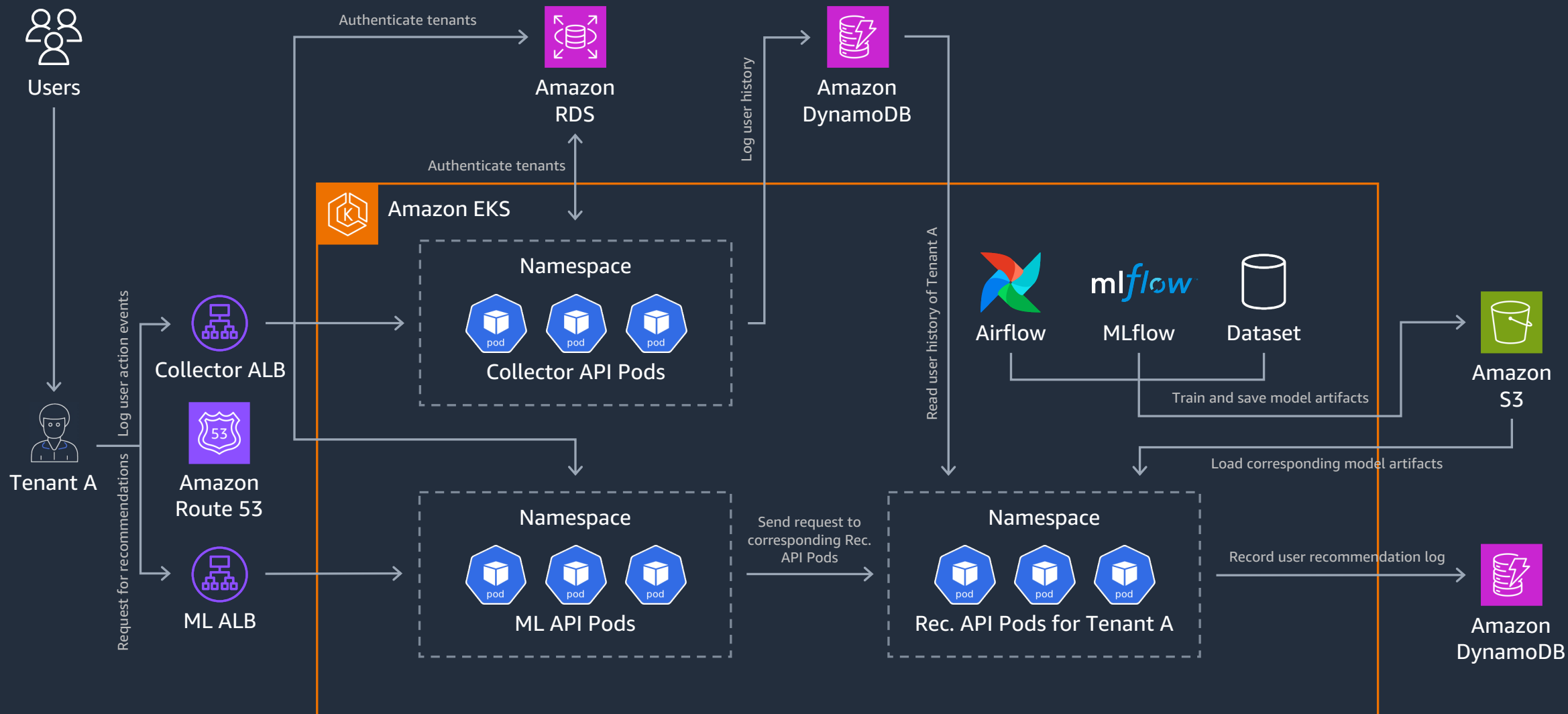
BLUX의 SAAS 솔루션 구축 시 기술 고려사항

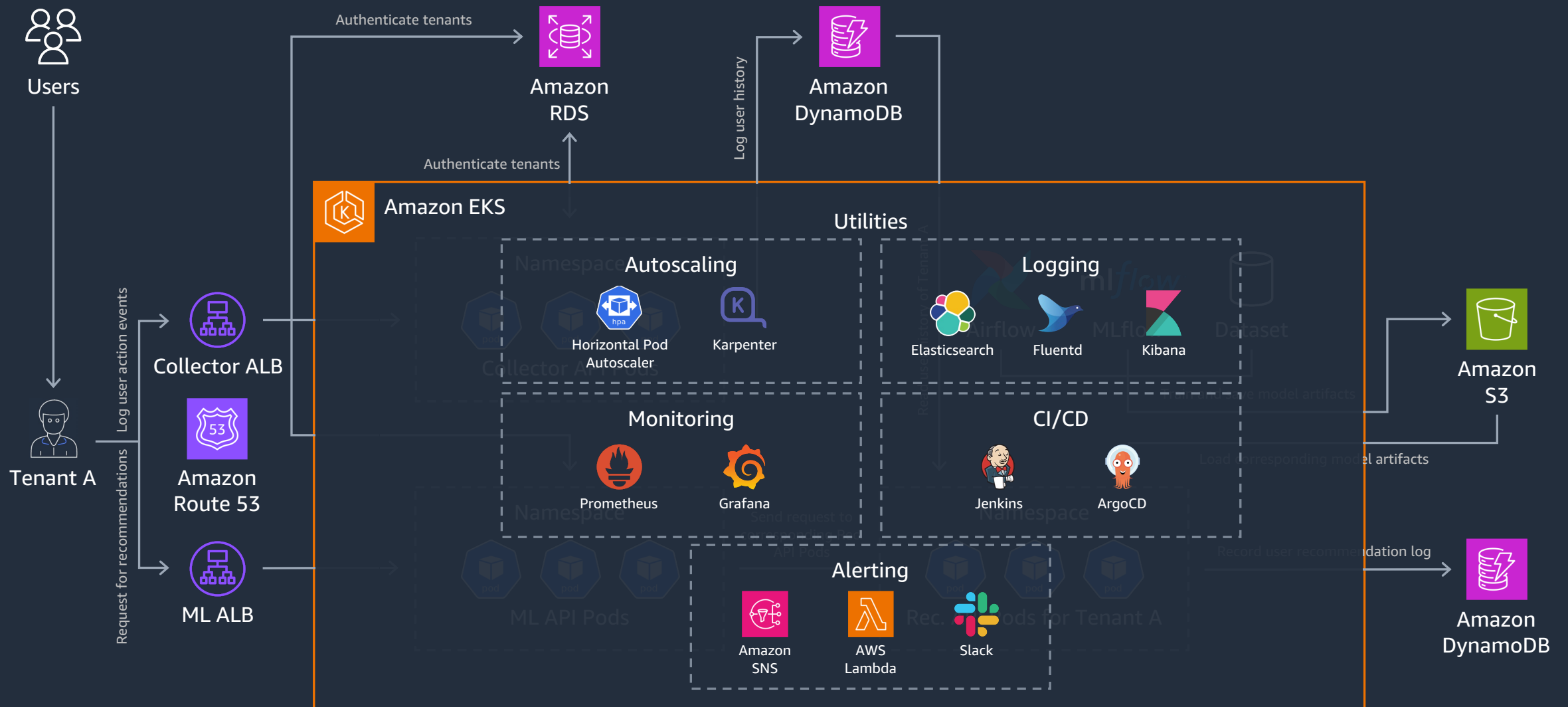
1. 모든 테넌트를 일괄적으로 관리할 수 있는 공통된 운영 환경
2. 민첩하고 자동화된 온보딩 프로세스
3. 다중 티어 시스템 및 티어에 따른 배포 모델
4. 테넌트 자원 격리

How were the Tech Considerations Addressed?

기술 고려사항이 어떻게 해결되었나요?

1. Amazon EKS 기반 SaaS 아키텍처 구축

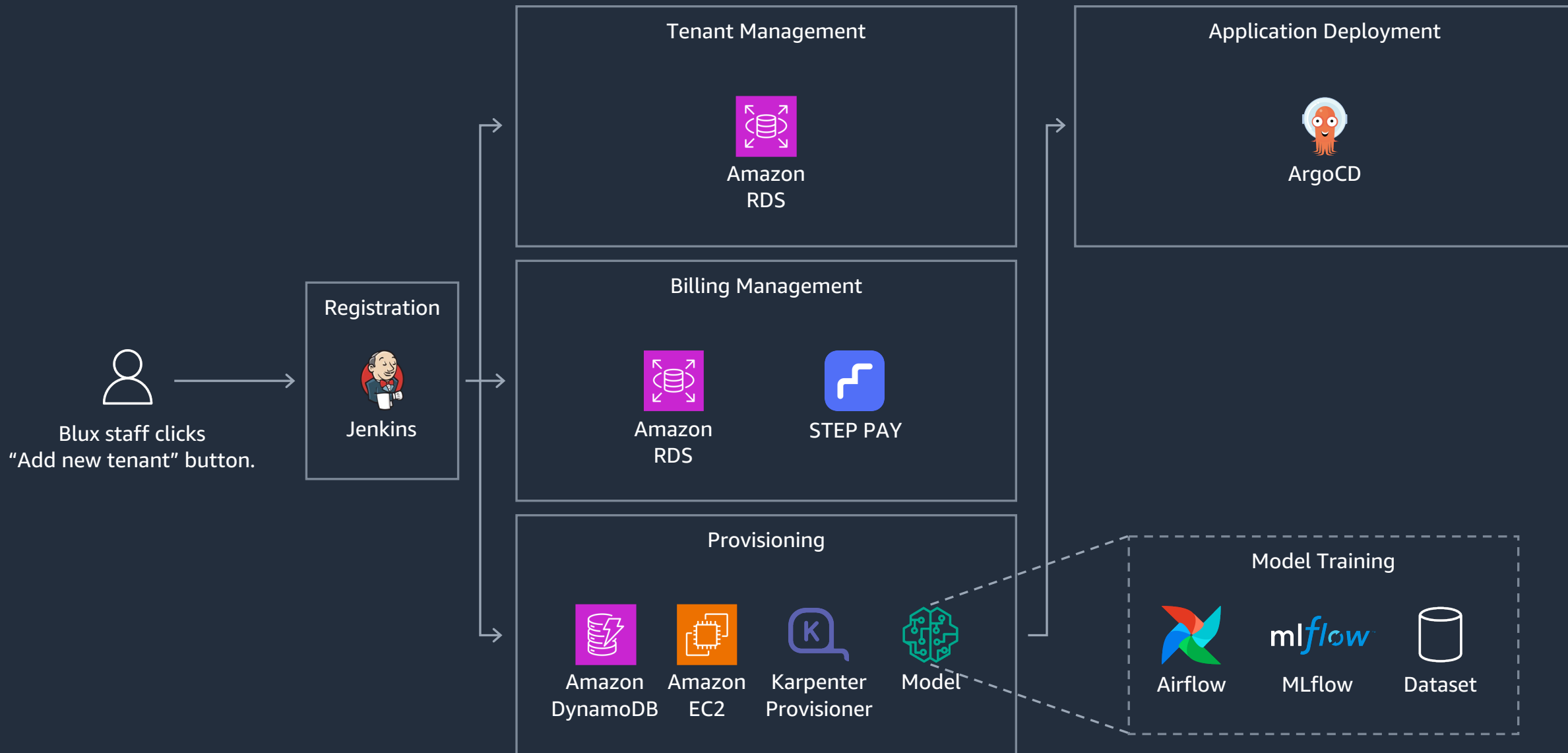




How were the Tech Considerations Addressed?

기술 고려사항이 어떻게 해결되었나요?

1. Amazon EKS 기반 SaaS 아키텍처 구축
2. 온보딩 프로세스 간소화 및 자동화



How were the Tech Considerations Addressed?

기술 고려사항이 어떻게 해결되었나요?

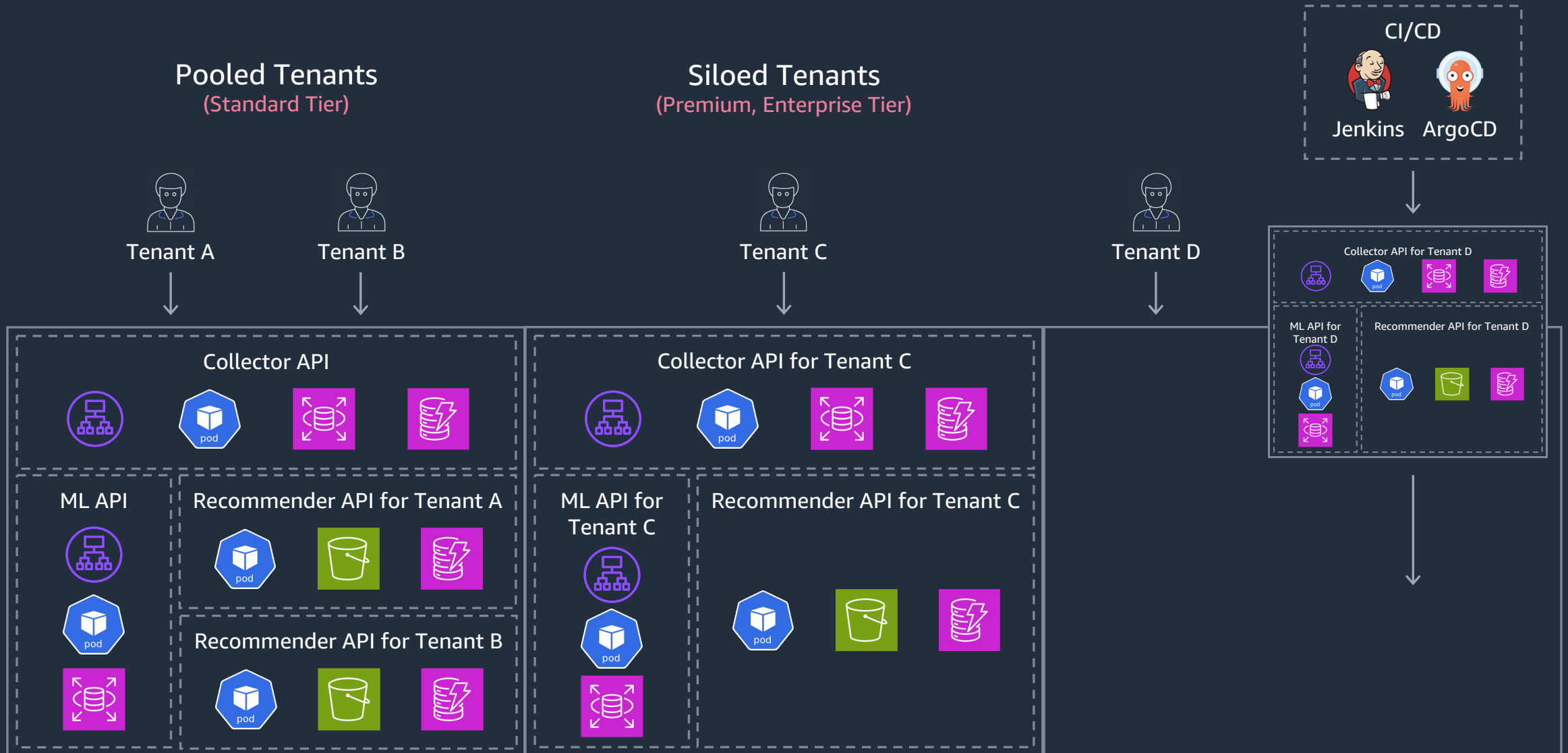
1. Amazon EKS 기반 SaaS 아키텍처 구축
2. 온보딩 프로세스 간소화 및 자동화
3. 다중 티어 시스템 도입 및 배포 모델 수립

How were the Tech Considerations Addressed?

Tier	Standard	Premium	Enterprise
실시간 추천 제공 여부	✓	✓	✓
전용 Endpoint 제공 여부	✗ (Endpoint 공유)	✓	✓
전용 노드 제공 여부	✗	✓	✓ (더 많은 자원 보장)
테넌트 격리 방식	IRSA, Namespace	IRSA, 노드 그룹	IRSA, 노드 그룹

Pooled Tenants (Standard Tier)

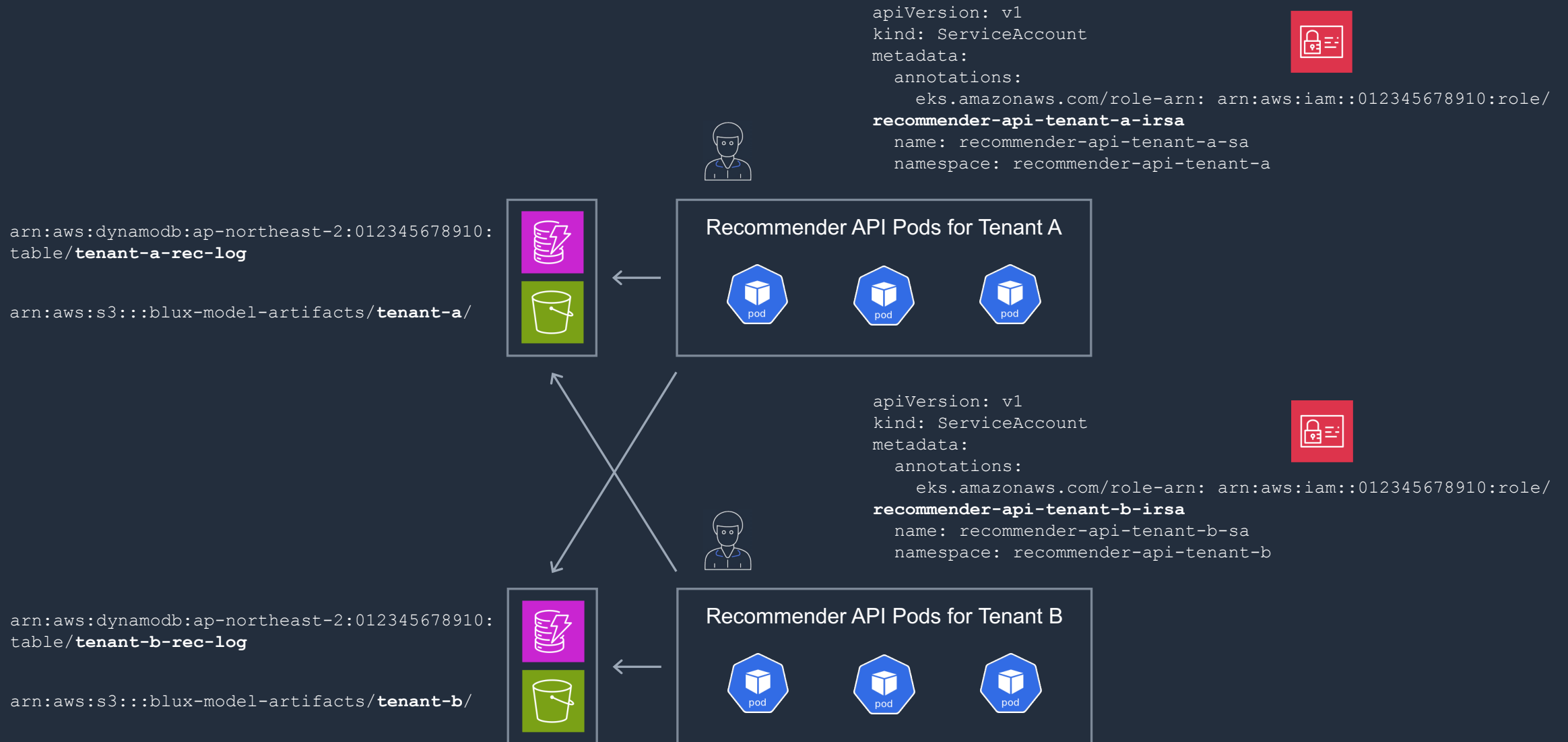
Siloed Tenants (Premium, Enterprise Tier)



How were the Tech Considerations Addressed?

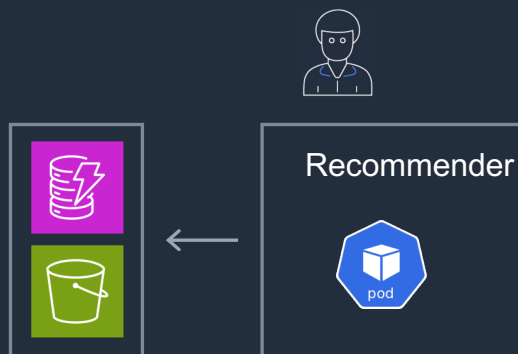
기술 고려사항이 어떻게 해결되었나요?

1. Amazon EKS 기반 SaaS 아키텍처 구축
2. 온보딩 프로세스 간소화 및 자동화
3. 다중 티어 시스템 도입 및 배포 모델 수립
4. IAM roles for service accounts (IRSA)를 활용한 테넌트 자원 격리



arn:aws:dynamodb:ap-northeast-2:012345678910:table/**tenant-a-rec-log**

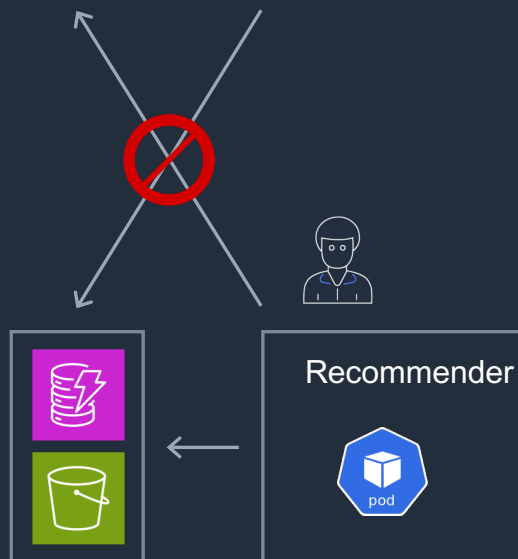
arn:aws:s3::blux-model-artifacts/**tenant-a/**



```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "AllowForTenantA",  
6       "Effect": "Allow",  
7       "Action": [  
8         "s3:GetObject",  
9         "dynamodb:PutItem"  
10      ],  
11      "Resource": [  
12        "arn:aws:s3::blux-model-artifacts/tenant-a/*",  
13        "arn:aws:dynamodb:ap-northeast-2:012345678910:table/tenant-a-rec-log"  
14      ]  
15    }  
16  ]  
17 }
```

arn:aws:dynamodb:ap-northeast-2:012345678910:table/**tenant-b-rec-log**

arn:aws:s3::blux-model-artifacts/**tenant-b/**



```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "AllowForTenantB",  
6       "Effect": "Allow",  
7       "Action": [  
8         "s3:GetObject",  
9         "dynamodb:PutItem"  
10      ],  
11      "Resource": [  
12        "arn:aws:s3::blux-model-artifacts/tenant-b/*",  
13        "arn:aws:dynamodb:ap-northeast-2:012345678910:table/tenant-b-rec-log"  
14      ]  
15    }  
16  ]  
17 }
```

How were the Tech Considerations Addressed?

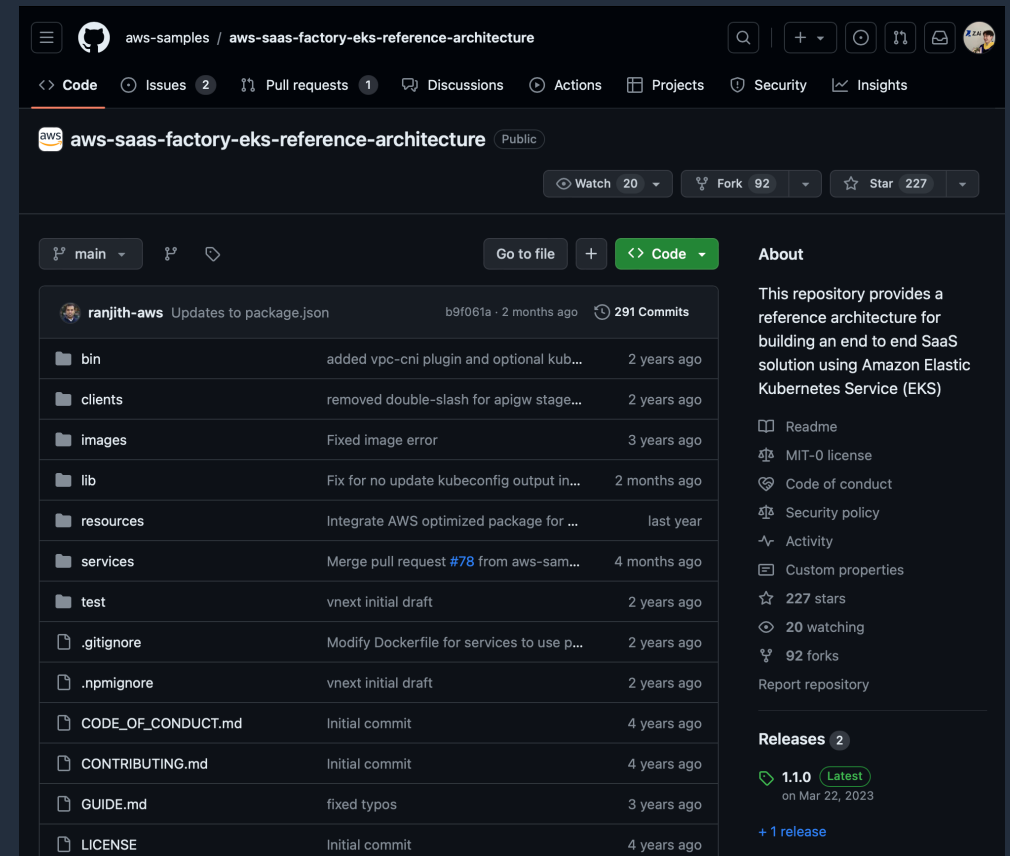
기술 고려사항이 어떻게 해결되었나요?

1. Amazon EKS 기반 SaaS 아키텍처 구축
2. 온보딩 프로세스 간소화 및 자동화
3. 다중 티어 시스템 도입 및 배포 모델 수립
4. IAM roles for service accounts (IRSA)를 활용한 테넌트 자원 격리

The Results of Blux's SaaS Solution Enhancement

BLUX의 SAAS 솔루션 고도화 결과

- AWS SaaS 아키텍처의 Best Practice 반영
 - AWS의 SaaS 아키텍처 구축 패턴을 적용함으로써 보다 높은 **안정성, 확장성** 그리고 **보안성**을 갖춘 SaaS 솔루션을 구축할 수 있었습니다.
 - <https://github.com/aws-samples/aws-saas-factory-eks-reference-architecture>



The Results of Blux's SaaS Solution Enhancement

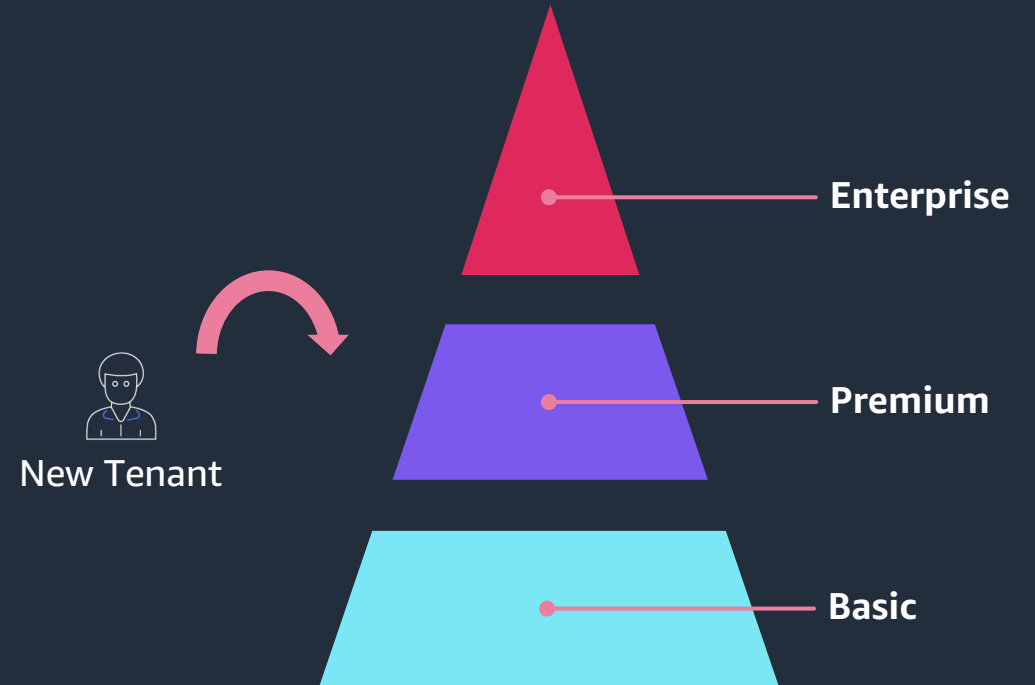
BLUX의 SAAS 솔루션 고도화 결과

- 온보딩 자동화의 효과
 - 수동 과정에서 종종 발생했던 **모델 경로 오기입, 공급 자원 누락, 의존성이 있는 작업 간의 선후 관계 착각** 등의 휴먼 에러를 줄일 수 있었습니다.
 - 단순 반복 작업이 전부 자동화되면서 기존에 **수 일**이 소요되던 온보딩, 초기 데이터 설정 그리고 모델 학습과 배포 과정이 **약 2~3시간 내외**로 줄어들었습니다.
 - 온보딩 안내, SDK 삽입, 추천 모델 개발, 자원 확보를 담당했던 인원들의 **투입 리소스**가 상당히 줄었습니다.

The Results of Blux's SaaS Solution Enhancement

BLUX의 SAAS 솔루션 고도화 결과

- 다중 티어 시스템 도입 및 다양한 격리 수준 적용
 - 티어마다 자원 공유 및 격리 수준을 다르게 제공하고, 이에 맞는 요금 측정 방법을 제공함으로써, **다양한 테넌트의 요구사항을 만족**시키는 SaaS 구독 솔루션을 제공할 수 있었습니다.



Agenda

Introducing Blux

Technological Considerations for Building Blux's SaaS Solution

How were the Technological Considerations Addressed?

The Results of Blux's SaaS Solution Enhancement



We Are Actively Hiring!



 <https://career.blux.ai>



 <https://www.linkedin.com/in/sunhongmin>



**여러분의 소중한 피드백을 기다립니다.
강연 종료 후, 강연 평가에 참여해주세요!**



감사합니다