

Using architectural decision records to streamline technical decision-making for a software development project

아키텍처 의사 결정 레코드를 사용하여 소프트웨어 개발 프로젝트에 대한 기술 의사 결정 간소화

Aws Prescriptive Guidance

AWS 규범적 지침



Aws Prescriptive Guidance: Using architectural decision records to streamline technical decision-making for a software development. project

AWS Prescriptive Guidance: 아키텍처 의사 결정 레코드를 사용하여 소프트웨어 개발을 위한 기술적 의사 결정을 간소화합니다. 프로젝트

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

저작권 © 2025 Amazon Web Services, Inc. 및/또는 그 계열사. 판권 소유.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

아마존의 상표 및 트레이드 드레스는 아마존이 아닌 상품 또는 서비스와 관련하하거나, 고객에게 혼란을 야기할 수 있는 방식으로 사용하거나, 아마존을 폄하하거나 신용을 떨어뜨리는 방식으로 사용될 수 없습니다. 아마존이 소유하지 않은 기타 모든 상표는 해당 소유자의 자산이며, 해당 소유자는 아마존과 제휴하거나, 연결되거나, 아마존이 후원할 수도 있습니다.

Table of Contents

Table of 내용

Introduction ...

소개...

Targeted business outcomes

목표 비즈니스 결과

ADR process . 3

ADR 프로세스 . 3

Scope of the ADR process. 3

ADR 프로세스의 범위입니다. 3

ADR contents. 3

ADR 콘텐츠. 3

ADR adoption process . 4

ADR 채택 프로세스 . 4

ADR review process ... 7

ADR 검토 프로세스 ... 7

Best practices 9

권장사항 9

FAQ .. 10

자주 묻는 질문(FAQ).. 10

What are the benefits of creating an ADR process? 10

ADR 프로세스를 생성하면 어떤 이점이 있습니까? 10

When should the project team create an ADR? 10

프로젝트 팀은 언제 ADR을 생성해야 합니까? 10

How often should the project team review an ADR? . 10

프로젝트 팀은 ADR을 얼마나 자주 검토해야 합니까? . 10

Who should create an ADR? 10

ADR은 누가 만들어야 하나요? 10

What information should an ADR contain? 10

ADR에는 어떤 정보가 포함되어야 합니까? 10

Where can I find ADR templates? .. 11

ADR 템플릿은 어디에서 찾을 수 있습니까? .. 11

Next steps and resources. ... 12

다음 단계 및 리소스. ... 12

Appendix: Example ADR13

부록: ADR 예 13

Document history ... 15

문서 기록 ... 15

Glossary 16

용어집.... 16

16

16

A. 17

□ 17

B 20

비| 20

c 22

□ 22

D. 25

D. 25

E 29

E 29

F 31

에프 31

G 33

지| 33

H. . 34

H. . 34

35

35

38

38

M. 39

중. 39

0 43

O 43

P 46

46쪽

Q 48

질문 48

R.. 49

R.. 49

S. 52

52절

T 56

E| 56

U . 57

U . 57

V. 58

58절

W... 58

W... 58

Z 59

Z 59

Using architectural decision records to streamline technical decision-making for a software development project

아키텍처 의사 결정 레코드를 사용하여 간소화합니다. 소프트웨어 개발 프로젝트를 위한 기술적 의사 결정

Darius Kunce and Dominik Goby, Amazon Web Services (Aws)

Darius Kunce와 Dominik Goby, Amazon Web Services(AWS)

March 2022 (document history)

2022년 3월(문서 기록)

This guide introduces the architectural decision records (ADR) process for software engineeringe projects. ADRs support team alignment, document strategic directions for a project or product, and reduce recurring and time-consuming decision-making efforts..

이 가이드에서는 소프트웨어 엔지니어링 프로젝트를 위한 아키텍처 의사 결정 기록(ADR) 프로세스를 소개합니다. ADR은 팀 조정을 지원하고, 프로젝트 또는 제품에 대한 전략적 방향을 문서화하며, 반복적이고 시간이 많이 걸리는 의사 결정 노력을 줄입니다..

During project and product development, software engineering teams need to make architectural.

프로젝트 및 제품 개발 중에 소프트웨어 엔지니어링 팀은 아키텍처를 만들어야 합니다.

decisions to reach their goals. These decisions can be technical, such as deciding to use the.

목표를 달성하기 위한 결정. 이러한 결정은 사용하기로 결정하는 것과 같은 기술적일 수 있습니다.

command query responsibility segregation (CQRS) pattern, or process-related, such as deciding.

CQRS(명령 쿼리 책임 분리) 패턴 또는 결정과 같은 프로세스 관련.

to use the GitFlow workflow to manage source code. Making these decisions is a time-consuming.

GitFlow 워크플로를 사용하여 소스 코드를 관리할 수 있습니다. 이러한 결정을 내리는 데는 시간이 많이 걸립니다.

and difficult process. Teams must justify, document, and communicate these decisions to relevant stakeholders.

그리고 어려운 과정. 팀은 이러한 결정을 정당화하고 문서화하며 관련 이해관계자에게 전달해야 합니다.

Three major anti-patterns often emerge when making architectural decisions:

아키텍처 결정을 내릴 때 세 가지 주요 안티 패턴이 자주 나타납니다.

.No decision is made at all, out of fear of making the wrong choice..

. 잘못된 선택을 할까 봐 전혀 결정을 내리지 않습니다..

.A decision is made without any justification, and people don't understand why it was made. This results in the same topic being discussed multiple times..

. 정당한 이유 없이 결정이 내려지고, 사람들은 왜 그런 결정이 내려졌는지 이해하지 못합니다. 이로 인해 동일한 주제가 여러 번 논의됩니다..

.The decision isn't captured in an architectural decision repository, so team members forget or don't know that the decision was made..

. 결정은 아키텍처 의사 결정 저장소에 캡처되지 않으므로 팀 구성원은 결정이 내려졌다는 사실을 잊어버리거나 알지 못합니다..

These anti-patterns are particularly important to tackle during the development process of a product or project.

이러한 안티 패턴은 제품 또는 프로젝트의 개발 프로세스 중에 해결하는 것이 특히 중요합니다.

Capturing the decision, the context, and considerations that led to the decision in the form of an ADR enables current and future stakeholders to collect information about the decisions made and the thought process behind each decision. This reduces software development time and provides better documentation for future teams.

ADR의 형태로 결정, 맥락 및 고려 사항을 캡처하면 현재 및 미래의 이해 관계자가 내린 결정과 각 결정 뒤에 있는 사고 과정에 대한 정보를 수집할 수 있습니다. 이를 통해 소프트웨어 개발 시간이 단축되고 향후 팀을 위한 더 나은 문서가 제공됩니다.

Targeted business outcomes

목표 비즈니스 성과

ADRs target three business outcomes:

ADR은 다음 세 가지 비즈니스 결과를 목표로 합니다.

.They align current and future team members..

. 그들은 현재와 미래의 팀원을 조정합니다..

. They set a strategic direction for the project or product..

. 그들은 프로젝트나 제품에 대한 전략적 방향을 설정합니다..

. They avoid decision anti-patterns by defining a process to properly document and communicate architectural decisions.

. 아키텍처 결정을 적절하게 문서화하고 전달하는 프로세스를 정의하여 의사 결정 안티 패턴을 방지합니다.

ADRs capture the context of the decision to inform future stakeholders. A collection of ADRs provide a hand-over experience and reference documentation. Team or project members use the ADR collection for follow-up projects and product feature planning. Being able to reference ADRs reduces the time required during development, reviews, and architectural decisions. ADRs also allow other teams to learn from, and gain insights into, considerations made by other project and product development teams.

ADR은 향후 이해 관계자에게 알리기 위해 결정의 컨텍스트를 캡처합니다. ADR 컬렉션은 핸드오버 경험 및 참조 문서를 제공합니다. 팀 또는 프로젝트 구성원은 후속 프로젝트 및 제품 기능 계획을 위해 ADR 컬렉션을 사용합니다. ADR을 참조할 수 있으면 개발, 검토 및 아키텍처 결정 중에 필요한 시간이 줄어듭니다. 또한 ADR을 통해 다른 팀은 다른 프로젝트 및 제품 개발 팀의 고려 사항으로부터 배우고 통찰력을 얻을 수 있습니다.

ADR process

ADR 프로세스

An architectural decision record (ADR) is a document that describes a choice the team makes about a significant aspect of the software architecture they're planning to build. Each ADR describes the architectural decision, its context, and its consequences. ADRs have states and therefore follow a lifecycle. For an example of an ADR, see the appendix.

ADR(아키텍처 의사 결정 레코드)은 팀이 빌드하려는 소프트웨어 아키텍처의 중요한 측면에 대해 선택하는 것을 설명하는 문서입니다. 각 ADR은 아키텍처 결정, 컨텍스트 및 결과를 설명합니다. ADR에는 상태가 있으므로 수명 주기를 따릅니다. ADR의 예는 부록을 참조하세요.

The ADR process outputs a collection of architectural decision records. This collection creates the decision log. The decision log provides the project context as well as detailed implementation and design information. Project members skim the headlines of each ADR to get an overview of the project context. They read the ADRs to dive deep into project implementations and design choices..

ADR 프로세스는 아키텍처 의사 결정 레코드 컬렉션을 출력합니다. 이 컬렉션은 의사 결정 로그를 만듭니다. 의사 결정 로그는 프로젝트 컨텍스트와 자세한 구현 및 설계 정보를 제공합니다. 프로젝트 구성원은 각 ADR의 헤드라인을 훑어보고 개요를 얻습니다. 프로젝트 컨텍스트. 그들은 ADR을 읽고 프로젝트 구현 및 설계 선택에 대해 자세히 알아봅니다..

When the team accepts an ADR, it becomes immutable. If new insights require a different decision, the team proposes a new ADR. When the team accepts the new ADR, it supersedes the previous ADR.

팀이 ADR을 수락하면 변경할 수 없게 됩니다. 새로운 인사이트에 다른 결정이 필요한 경우. 팀은 새 ADR을 제안합니다. 팀이 새 ADR을 수락하면 이전 ADR을 대체합니다.

Scope of the ADR process

ADR 프로세스의 범위

Project members should create an ADR for every architecturally significant decision that affects the software project or product, including the following (Richards and Ford 2020):

프로젝트 구성원은 다음을 포함하여 소프트웨어 프로젝트 또는 제품에 영향을 미치는 모든 아키텍처적으로 중요한 결정에 대해 ADR을 생성해야 합니다(Richards and Ford 2020).

.Structure (for example, patterns such as microservices).

. 구조(예: 마이크로 서비스와 같은 패턴).

. Non-functional requirements (security, high availability, and fault tolerance) .Dependencies (coupling of components).

. 비기능 요구 사항(보안, 고가용성 및 내결함성) . 종속성(구성 요소 결합).

.Interfaces (APIs and published contracts)

. 인터페이스(API 및 게시된 계약)

.Construction techniques (libraries, frameworks, tools, and processes).

. 구성 기술(라이브러리, 프레임워크, 도구 및 프로세스).

-unctional and non-functional requirements are the most common inputs to the ADR process.

-unctional 및 비기능 요구 사항은 ADR 프로세스에 대한 가장 일반적인 입력입니다.

ADR contents

ADR 콘텐츠

When the team identifies a need for an ADR, a team member starts to write the ADR based on a projectwide template. (See the ADR GitHub organization for example templates.) The template simplifies ADR creation and ensures that the ADR captures all the relevant information..

팀에서 ADR의 필요성을 식별하면 팀 구성원은 프로젝트 전체 템플릿을 기반으로 ADR 작성을 시작합니다. (예제 템플릿은 ADR GitHub 조직을 참조하세요.) 그만큼. 템플릿은 ADR 생성을 단순화하고 ADROI 모든 관련 정보를 캡처하도록 합니다..

At a minimum, each ADR should define the context of the decision, the decision itself, and the consequences of the decision for the project and its deliverables. (For examples of these sections,. see the appendix.) One of the most powerful aspects of the ADR structure is that it focuses on the reason for the decision rather than how the team implemented it. Understanding why the team. made the decision makes it easier for other team members to adopt the decision, and prevents other architects who weren't involved in the decision-making process to overrule that decision in. the future.

최소한 각 ADR은 결정의 맥락, 결정 자체, 프로젝트 및 결과물에 대한 결정의 결과를 정의해야 합니다. (이 섹션의 예는 부록을 참조하십시오.) ADR 구조의 가장 강력한 측면 중 하나는 팀이 결정을 구현한 방법보다는 결정의 이유에 초점을 맞춘다는 것입니다. 팀이 왜 내렸는지 이해합니다. 결정을 내리면 다른 팀 구성원이 결정을 더 쉽게 채택할 수 있고 의사 결정 프로세스에 관여하지 않은 다른 설계자가 해당 결정을 무효화하는 것을 방지할 수 있습니다. 미래.

ADR adoption process

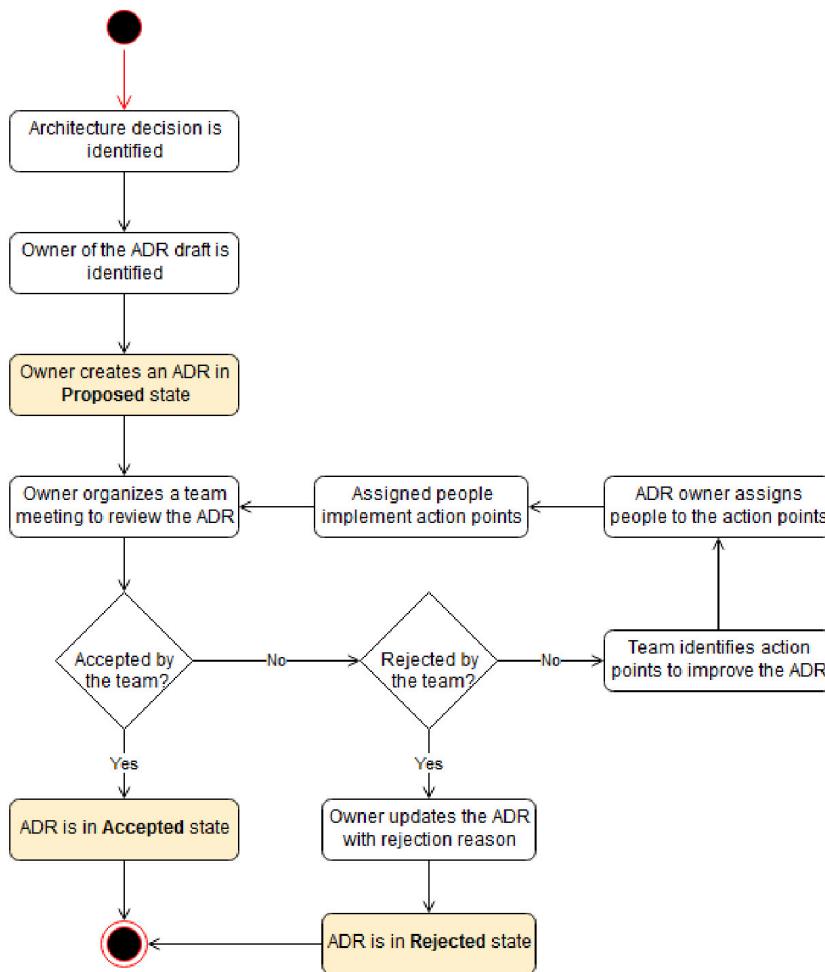
ADR 채택 프로세스

Every team member can create an ADR, but the team should establish a definition of ownership for an ADR. Each author who is the owner of an ADR should actively maintain and communicate the ADR content. To clarify this ownership, this guide refers to ADR authors as ADR owners in the following sections. Other team members can always contribute to an ADR. If the content of an ADF changes before the team accepts the ADR, the owner should approve these changes..

모든 팀 구성원은 ADR을 만들 수 있지만 팀은 ADR에 대한 소유권 정의를 설정해야 합니다. ADR의 소유자인 각 작성자는 ADR 콘텐츠를 적극적으로 유지 관리하고 전달해야 합니다. 이 소유권을 명확히 하기 위해 이 가이드에서는 다음 섹션에서 ADR 작성자를 ADR 소유자로 지칭 합니다. 다른 팀 구성원은 항상 ADR에 기여할 수 있습니다. 팀이 ADR을 수락하기 전에 ADF의 콘텐츠가 변경되는 경우 소유자는 이러한 변경 내용을 승인해야 합니다.

The following diagram illustrates the ADR creation, ownership, and adoption process.

다음 다이어그램에서는 ADR 생성, 소유권 및 채택 프로세스를 보여 줍니다.



After the team identifies an architectural decision and its owner, the ADR owner provides the ADR in the Proposed state at the beginning of the process. ADRs in the Proposed state are ready for review.

팀이 아키텍처 결정 및 해당 소유자를 식별한 후 ADR 소유자는 프로세스 시작 시 제안 상태의 ADR을 제공합니다. 제안 상태의 ADR은 검토 할 준비가 되었습니다.

The ADR owner then initiates the review process for the ADR. The goal of the ADR review process is to decide whether the team accepts the ADR, determines that it needs rework, or rejects the ADR. The project team, including the owner, reviews the ADR. The review meeting should start with a

그런 다음 ADR 소유자는 ADR에 대한 검토 프로세스를 시작합니다. ADR 검토 프로세스의 목표는 팀이 ADR을 수락할지, 재작업이 필요하다고 결정할지, ADR을 거부할지 여부를 결정하는 것입니다. 소유자를 포함한 프로젝트 팀은 ADR을 검토합니다. 검토 회의는

dedicated time slot to read the ADR. On average, 10 to 15 minutes should be enough. During this time, each team member reads the document and adds comments and questions to flag unclear topics. After the review phase, the ADR owner reads out and discusses each comment with the team.

ADR을 읽을 수 있는 전용 시간대. 평균적으로 10분에서 15분이면 충분합니다. 이 동안 시간, 각 팀원은 문서를 읽고 의견과 질문을 추가하여 불분명한 주제에 플래그를 지정합니다. 검토 단계가 끝나면 ADR 소유자는 각 의견을 읽고 팀과 논의합니다.

If the team finds action points to improve the ADR, the state of the ADR stays Proposed. The ADR owner formulates the actions, and, in collaboration with the team, adds an assignee to each action. Each team member can contribute and resolve the action points. It is the responsibility of the ADR owner to reschedule the review process.

팀에서 ADR을 개선하기 위한 작업 지점을 찾으면 ADR의 상태는 제안된 상태로 유지됩니다. ADR 소유자는 작업을 공식화하고 팀과 협력하여 각 작업에 담당자를 추가합니다. 각 팀 구성원은 작업 지점에 기여하고 해결할 수 있습니다. 검토 프로세스를 다시 예약하는 것은 ADR 소유자의 책임입니다.

The team can also decide to reject the ADR. In this case, the ADR owner adds a reason for the rejection to prevent future discussions on the same topic. The owner changes the ADR state to Rejected.

팀은 ADR을 거부할지 결정할 수도 있습니다. 이 경우 ADR 소유자는 동일한 주제에 대한 향후 토론을 방지하기 위해 거부 이유를 추가합니다. 소유자는 ADR 상태를 거부됨으로 변경합니다.

If the team approves the ADR, the owner adds a timestamp, version, and list of stakeholders. The owner then updates the state to Accepted..

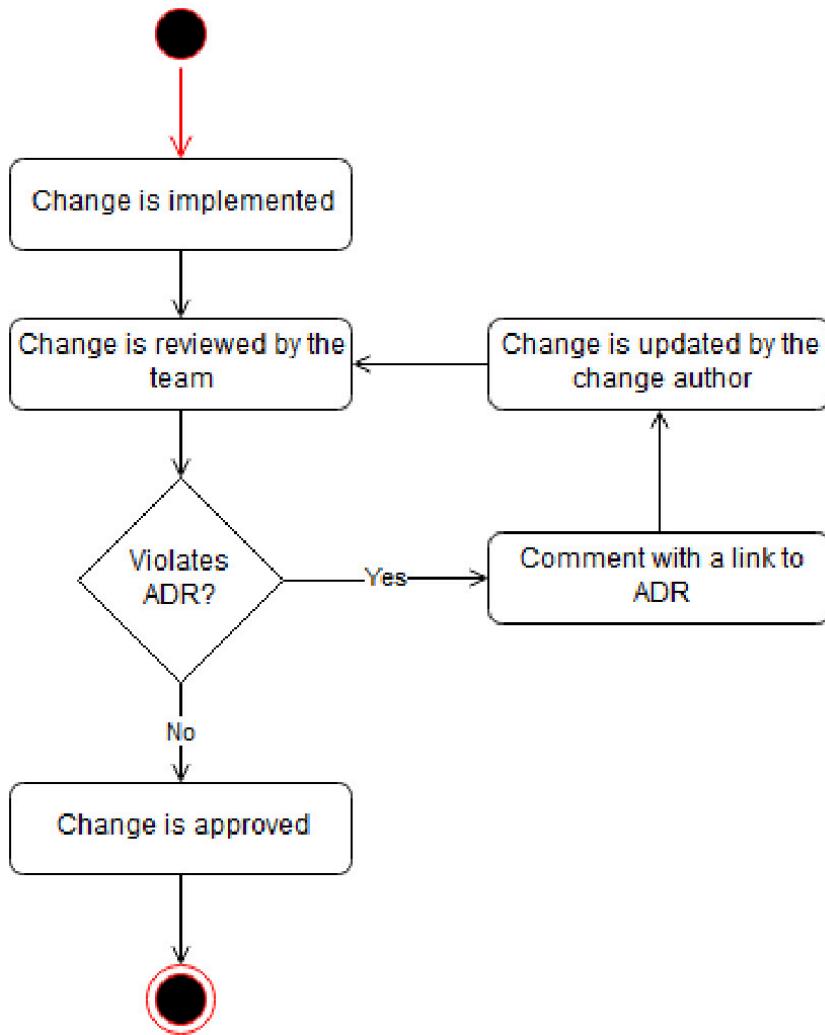
팀에서 ADR을 승인하면 소유자는 타임스탬프, 버전 및 관련자 목록을 추가합니다. 그런 다음 소유자는 상태를 수락됨으로 업데이트합니다.

ADRs and the decision log they create represent decisions made by the team and provide a history of all decisions. The team uses the ADRs as a reference during code and architectural reviews where possible. In addition to performing code reviews, design tasks, and implementation tasks, team members should consult ADRs for strategic decisions for the product..

ADR과 ADRO이 생성하는 의사 결정 로그는 팀에서 내린 결정을 나타내며 기록을 제공합니다. 모든 결정의 팀은 가능한 경우 코드 및 아키텍처 검토 중에 ADR을 참조로 사용합니다. 코드 검토, 설계 작업 및 구현 작업을 수행하는 것 외에도 팀 구성원은 제품에 대한 전략적 결정을 위해 ADR을 참조해야 합니다.

The following diagram shows the process of applying an ADR to validate if a change in a software component conforms to the agreed decisions..

다음 다이어그램에서는 소프트웨어 구성 요소의 변경 내용이 합의된 결정을 준수하는지 확인하기 위해 ADR을 적용하는 프로세스를 보여 줍니다.



As a good practice, each software change should go through peer reviews and require at least one approval. During the code review, a code reviewer might find changes that violate one or more ADRs. In this case, the reviewer asks the author of the code change to update the code, and shares a link to the ADR. When the author updates the code, it is approved by peer reviewers and merged into the main code base.

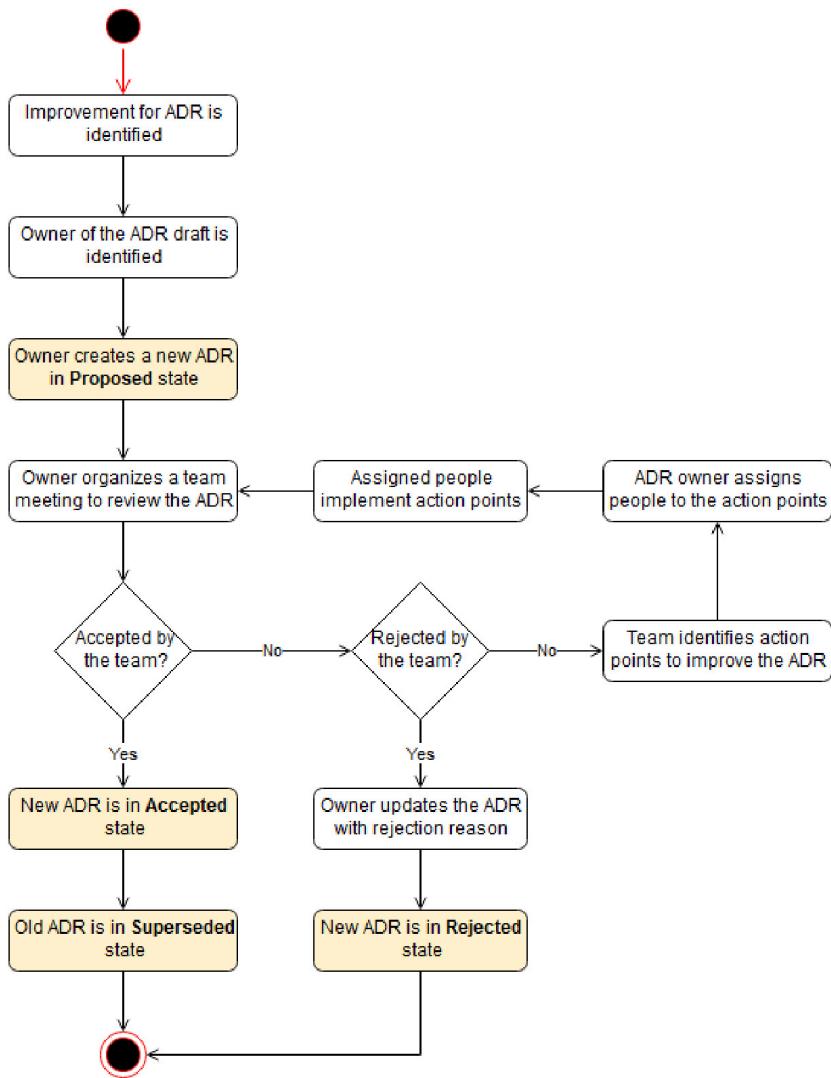
각 소프트웨어 변경 내용은 피어 리뷰를 거쳐야 하며 하나 이상의 승인이 필요합니다. 코드 검토 중에 코드 검토자는 하나 이상의 ADR을 위반하는 변경 내용을 찾을 수 있습니다. 이 경우 검토자는 코드 변경 작성자에게 코드 업데이트를 요청하고 ADR에 대한 링크를 공유합니다. 작성자가 코드를 업데이트하면 피어 검토자가 코드를 승인하고 기본 코드 베이스에 병합됩니다.

ADR review process

ADR 검토 프로세스

The team should treat ADRs as immutable documents after the team accepts or rejects them.. Changes to an existing ADR requires creating a new ADR, establishing a review process for the new ADR, and approving the ADR. If the team approves the new ADR, the owner should change the state of the old ADR to Superseded. The following diagram illustrates the update process..

팀은 ADR을 수락하거나 거부한 후 ADR을 변경할 수 없는 문서로 처리해야 합니다.. 기존 ADR을 변경하려면 새 ADR을 만들고, 새 ADR에 대한 검토 프로세스를 설정하고, ADR을 승인해야 합니다. 팀에서 새 ADR을 승인하는 경우 소유자는 변경해야 합니다. 이전 ADR의 상태를 대체됨으로 변경합니다. 다음 다이어그램에서는 업데이트 프로세스를 보여 줍니다.



Best practices

권장사항

Promote ownership. Each project team member should be empowered to create and own an ADR. This practice distributes architectural research work among team members and offloads that work from the solutions architect or team lead. It also fosters a sense of ownership in the decision-making process. This helps the team adopt those decisions faster instead of treating them as decisions that were imposed from higher levels of the organization..

소유권을 증진합니다. 각 프로젝트 팀 구성원은 ADR을 만들고 소유할 수 있는 권한을 부여받아야 합니다. 이 관행은 팀 구성원에게 아키텍처 연구 작업을 분배하고 솔루션 설계자 또는 팀 리더의 작업을 오프로드합니다. 또한 주인의식을 키워줍니다. 의사 결정 과정. 이를 통해 팀은 이러한 결정을 조직의 상위 수준에서 부과된 결정으로 취급하는 대신 더 빨리 채택할 수 있습니다..

Preserve ADR history. ADRs should have a change history, and each change should have an owner. When the ADR owner updates the ADR, they should change the status of the old ADR to Superseded, note their changes in the change history of the new ADR, and keep the old ADR in the decision log.

ADR 기록을 보존합니다. ADR에는 변경 기록이 있어야 하며 각 변경 내용에는 소유자가 있어야 합니다. ADR 소유자가 ADR을 업데이트할 때 이전 ADR의 상태를 다음으로 변경해야 합니다. 대체됨, 새 ADR의 변경 기록에서 변경 내용을 기록하고 이전 ADR을 예 유지합니다. 의사 결정 로그.

Schedule regular review meetings. If you are on a new (greenfield) project, the ADR process can be quite intense in the beginning. We recommend that you establish a cadence of regular ADR discussion and review meetings before or after the daily standup. With this approach, the defined ADRs will stabilize in two or three sprints, and you can build a solid foundation with fewer meetings.

정기적인 검토 회의를 예약합니다. 새로운(그린필드) 프로젝트를 진행하는 경우 ADR 프로세스는 처음에는 상당히 강렬할 수 있습니다. 일일 스탠드업 전후에 정기적인 ADR 토론 및 검토 회의의 주기를 설정하는 것이 좋습니다. 이 접근 방식을 사용하면 정의된 ADR이 두세 번의 스프린트에서 안정화되고 더 적은 회의로 견고한 기반을 구축할 수 있습니다.

Store ADRs in a central location. Each project member should have access to the collection of ADRs. We recommend that you store the ADRs in a central location and reference them on the main page of your project documentation. There are two popular options for storing ADRs:

ADR을 중앙 위치에 저장합니다. 각 프로젝트 구성원은 ADR 컬렉션에 액세스할 수 있어야 합니다. ADR을 중앙 위치에 저장하고 프로젝트 설명서의 기본 페이지에서 참조하는 것이 좋습니다. ADR을 저장하는 데 널리 사용되는 두 가지 옵션이 있습니다.

- .A Git repository, which makes it easier to version ADRs.
- .A wiki page, which makes the ADRs accessible to all team members.
- .ADR의 버전을 더 쉽게 지정할 수 있는 Git 리포지토리입니다.
- .모든 팀 구성원이 ADR에 액세스할 수 있도록 하는 위키 페이지

Address non-compliant code. The ADR process doesn't solve the issue of non-compliant legacy code. If you have legacy code that doesn't support the established ADRs, you can either update the outdated code base or artifacts gradually, while introducing new changes, or your team can decide to refactor the code explicitly by creating technical debt tasks..

비규격 코드를 해결합니다. ADR 프로세스는 비규격 레거시 문제를 해결하지 않습니다. 코드. 설정된 ADR을 지원하지 않는 레거시 코드가 있는 경우 오래된 코드 베이스 또는 아티팩트를 점진적으로 업데이트하면서 새로운 변경 내용을 도입하거나 팀에서 기술 부채 작업을 만들어 코드를 명시적으로 리팩터링하기로 결정할 수 있습니다.

FAQ

자주 묻는 질문(FAQ)

What are the benefits of creating an ADR process?

ADR 프로세스를 생성하면 어떤 이점이 있습니까?

The project team should create an ADR process to streamline architectural decision-making, prevent repeated discussions about the same architectural topics, and communicate architectural decisions effectively.

프로젝트 팀은 아키텍처 의사 결정을 간소화하고, 동일한 아키텍처 주제에 대한 반복적인 논의를 방지하고, 아키텍처 결정을 효과적으로 전달하기 위해 ADR 프로세스를 만들어야 합니다.

When should the project team create an ADR?

프로젝트 팀은 언제 ADR을 생성해야 합니까?

The project team should create an ADR for every aspect of the software that affects structure (patterns such as microservices), non-functional requirements (security, high availability, and fault tolerance), dependencies (coupling of components), interfaces (APIs and published contracts), and construction techniques (libraries, frameworks, tools, and processes).

프로젝트 팀은 구조(マイクロサービス와 같은 패턴), 비기능적 요구 사항(보안, 고가용성 및 내결함성), 종속성(구성 요소 결합), 인터페이스 (API 및 게시된 계약) 및 구성 기술(라이브러리, 프레임워크, 도구 및 프로세스)에 영향을 미치는 소프트웨어의 모든 측면에 대해 ADR을 만

들어야 합니다.

How often should the project team review an ADR?

프로젝트 팀은 ADR을 얼마나 자주 검토해야 합니까?

The project team should review the ADR at least once before accepting it.

프로젝트 팀은 ADR을 수락하기 전에 적어도 한 번 검토해야 합니다.

Who should create an ADR?

ADR은 누가 만들어야 하나요?

Every team member can create an ADR. We recommend that you promote a notion of ownership for ADRs. An author who owns the ADR should actively maintain and communicate the ADR content. Other team members can always contribute to an ADR. The ADR owner should approve changes to an ADR.

모든 팀 구성원은 ADR을 만들 수 있습니다. ADR에 대한 소유권 개념을 홍보하는 것이 좋습니다. ADR을 소유한 작성자는 ADR 콘텐츠를 적극적으로 유지 관리하고 전달해야 합니다. 다른 팀 구성원은 항상 ADR에 기여할 수 있습니다. ADR 소유자는 ADR에 대한 변경 내용을 승인해야 합니다.

What information should an ADR contain?

ADR에는 어떤 정보가 포함되어야 합니까?

At a minimum, each ADR has to define the context of the decision, the decision itself, and the consequences of the decision for the project and its deliverables. The context should mention possible solutions the team considered. It should also contain any relevant information relating to the project, customer, or technology stack. The decision must clearly state, in imperative language, the solution the team has decided to adopt. Avoid using words such as "should," and phrase each decision to say "We use..." or "The team has to use..." The consequences section should mention all known trade-offs of making the decision. Each ADR must have a status and a changelog that contains the change date and the person who is responsible for the change..

최소한 각 ADR은 결정의 맥락, 결정 자체 및 정의해야 합니다. 프로젝트 및 결과물에 대한 결정의 결과. 컨텍스트는 언급해야 합니다. 팀이 고려한 가능한 솔루션. 또한 관련된 모든 관련 정보도 포함되어야 합니다. 프로젝트, 고객 또는 기술 스택. 결정에는 명령형 언어로 명확하게 명시되어야 합니다. 팀이 채택하기로 결정한 솔루션입니다. "해야 한다"와 같은 단어를 사용하지 말고 각 결정에 "우리는 ..." 또는 "팀은 사용해야 합니다..." 결과 섹션에는 결정을 내릴 때 알려진 모든 장단점을 언급해야 합니다. 각 ADR에는 변경 날짜와 변경 책임자가 포함된 상태 및 변경 로그가 있어야 합니다..

Where can I find ADR templates?

ADR 템플릿은 어디에서 찾을 수 있나요?

There are multiple versions and variants of ADR templates available. For a public collection of commonly used ADR templates, see the ADR GitHub repository.

ADR 템플릿에는 여러 버전과 변형을 사용할 수 있습니다. 일반적으로 사용되는 ADR 템플릿의 공개 컬렉션은 ADR GitHub 리포지토리를 참조하세요.

Next steps and resources

다음 단계 및 리소스

We recommend that you start small and see the benefit that ADRs bring to your team. If you are working on an ongoing project, identify the next architectural change and apply the proposed ADR process to create your first ADR.

소규모로 시작하여 ADRO이 팀에 제공하는 이점을 확인하는 것이 좋습니다. 진행 중인 프로젝트에서 작업 중인 경우 다음 아키텍처 변경 사항을 식별하고 제안된 ADR 프로세스를 적용하여 첫 번째 ADR을 만듭니다.

Another starting point is to document your overall software development process by using ADRs. Often, the development process is based on tacit knowledge that the team didn't capture in any documentation. Documenting this process enables a smoother experience for new members of the team.

또 다른 출발점은 ADR을 사용하여 전체 소프트웨어 개발 프로세스를 문서화하는 것입니다. 개발 프로세스는 팀이 문서에서 포착하지 않은 암묵적 지식을 기반으로 하는 경우가 많습니다. 이 프로세스를 문서화하면 팀의 새로운 구성원이 보다 원활하게 경험을 할 수 있습니다.

If you are on a greenfield project, apply the ADR process and start capturing all the decisions from the beginning in a few sentences. You can then iterate on those ADRs and supplement them with new information. After you establish your ADRs, you can start using them as a reference in your code review process.

그린필드 프로젝트에 있는 경우 ADR 프로세스를 적용하고 몇 문장으로 처음부터 모든 결정을 캡처하기 시작합니다. 그런 다음 해당 ADR을 반복하고 새로운 정보로 보완할 수 있습니다. ADR을 설정한 후에는 코드 검토 프로세스에서 참조로 사용할 수 있습니다.

Resources

리소스

.Architecture Decision Records. <https://adr.github.io/>.

. 아키텍처 결정 기록. <https://adr.github.io/>.

.Richards, Mark and Neal Ford. 2021. Fundamentals of Software Architecture. Sebastopol: O'Reilly Media.

. 리차즈, 마크, 닐 포드. 2021. 소프트웨어 아키텍처의 기초. 세바스토폴: 오라일리 미디어.

Appendix: Example ADR

부록: ADR 예제

Title

타이틀

This decision defines the software development lifecycle approach for ABC application development.

이 결정은 ABC 애플리케이션 개발을 위한 소프트웨어 개발 수명 주기 접근 방식을 정의합니다.

Status

상태

Accepted

허용

Date

날짜

2022-03-11

2022-03-11

Context

문맥

ABC application is a packaged solution, which will be deployed to the customer's environment by using a deployment package. We need to have a development process that will enable us to have a controllable feature, hotfix, and release pipeline..

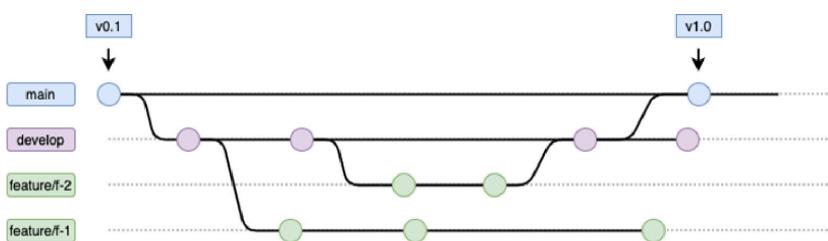
ABC 애플리케이션은 고객의 환경에 배포되는 패키지 솔루션입니다. 배포 패키지를 사용합니다. 제어 가능한 기능, 핫픽스 및 릴리스 파이프라인을 가질 수 있는 개발 프로세스가 필요합니다..

Decision

결정

We use an adapted version of the GitFlow workflow to develop ABC application.

우리는 GitFlow 워크플로의 적응된 버전을 사용하여 ABC 애플리케이션을 개발합니다.



For simplicity, we will not be using the hotfix/* and release/* branches, because ABC application will be packaged instead of being deployed to a specific environment. For this reason, there is no need for additional complexity that might prevent us from reacting quickly to fix bugs in production releases, or testing releases in a separate environment.

단순화를 위해 ABC 애플리케이션이 특정 환경에 배포되는 대신 패키지화되기 때문에 핫픽스/* 및 릴리스/* 분기를 사용하지 않습니다. 이러한 이유로 프로덕션 릴리스의 버그를 수정하거나 별도의 환경에서 릴리스를 테스트하기 위해 신속하게 대응하는 데 방해가 될 수 있는 추가 복잡성이 필요하지 않습니다.

The following is the agreed branching strategy:

합의된 분기 전략은 다음과 같습니다.

.Each repository must have a protected main branch that will be used to tag releases.

. 각 리포지토리에는 릴리스에 태그를 지정하는 데 사용할 보호된 기본 분기가 있어야 합니다.

.Each repository must have a protected develop branch for all ongoing development work..

. 각 리포지토리에는 진행 중인 모든 개발 작업에 대해 보호된 개발 분기가 있어야 합니다.

Consequences

결과

Positive:

플러스:

.Adapted GitFlow process will enable us to control release versioning of the ABC application.

. 적응된 GitFlow 프로세스를 통해 ABC 애플리케이션의 릴리스 버전 관리를 제어할 수 있습니다.

Negative:

マイナス:

.GitFlow is more complicated than trunk-based development or GitHub flow and has more overhead.

. GitFlow는 트렁크 기반 개발이나 GitHub 흐름보다 더 복잡하며 오버헤드가 더 많습니다.

Compliance

컴플라이언스

The main and develop branches in each repository must be marked as Protected. . Changes to the main and develop branches must be propagated by using merge requests. .At least one approval is required for every merge request.

각 리포지토리의 기본 및 개발 분기는 보호됨으로 표시되어야 합니다. . 기본 및 개발 분기에 대한 변경 내용은 병합 요청을 사용하여 전파해야 합니다.. 모든 병합 요청에 대해 하나 이상의 승인이 필요합니다.

Notes

노트

Author: Jane Doe.

저자: 제인 도우.

Version: 0.1

버전: 0.1

Changelog:e .0.1: Initial proposed version

변경 로그:e .0.1: 초기 제안 버전

Document history

문서 기록

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

다음 표에서는 이 가이드의 주요 변경 사항에 대해 설명합니다. 향후 업데이트에 대한 알림을 받으려면 RSS 피드를 구독할 수 있습니다.

Change Description Date Initial publication March 16, 2022

변경 설명 날짜 최초 발행일 2022년 3월 16일

AWS Prescriptive Guidance glossary

AWS 규범적 지침 용어집

The following are commonly used terms in strategies, guides, and patterns provided by AWS. Prescriptive Guidance. To suggest entries, please use the Provide feedback link at the end of the glossary.

다음은 AWS에서 제공하는 전략, 가이드 및 패턴에서 일반적으로 사용되는 용어입니다. 규범적 지침. 항목을 제안하려면 용어집 끝에 있는 피드백 제공 링크를 사용하십시오.

Numbers

숫자

7 Rs

7 루피

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

애플리케이션을 클라우드로 이동하기 위한 7가지 일반적인 마이그레이션 전략. 이러한 전략은 Gartner가 2011년에 식별한 5R을 기반으로 하며 다음으로 구성됩니다.

- . Refactor/re-architect - Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- . 리팩터링/재설계 - 클라우드 네이티브 기능을 최대한 활용하여 애플리케이션을 이동하고 아키텍처를 수정하여 민첩성, 성능 및 확장성을 개선합니다. 여기에는 일반적으로 운영 체제 및 데이터베이스 포팅이 포함됩니다. 예: 온프레미스 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 버전으로 마이그레이션합니다.
- . Replatform (lift and reshape) - Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDs) for Oracle in the Aws Cloud.
- . 플랫폼 변경(리프트 및 재구성) - 애플리케이션을 클라우드로 이동하고 일부 수준을 도입합니다. 클라우드 기능을 활용하기 위한 최적화. 예: 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 Oracle용 Amazon Relational Database Service(Amazon RD)로 마이그레이션

합니다.

. Repurchase (drop and shop) - Switch to a different product, typically by moving from a traditional license to a SaaS model.

Example: Migrate your customer relationship management (CRM) system to Salesforce.com.

. 재구매(드롭 앤 샵) - 일반적으로 기존 라이선스에서 SaaS 모델로 이동하여 다른 제품으로 전환합니다. 예: 고객 관계 관리(CRM) 시스템을 Salesforce.com 로 마이그레이션합니다.

: Rehost (lift and shift) - Move an application to the cloud without making any changes to take advantage of cloud capabilities.

Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the Aws Cloud..

: 리호스트(리프트 앤 시프트) - 클라우드 기능을 활용하기 위해 변경하지 않고 애플리케이션을 클라우드로 이동합니다. 예: 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 EC2 인스턴스에 있는 Oracle로 마이그레이션합니다.

. Relocate (hypervisor-level lift and shift) - Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to Aws..

. 재배치(하이퍼바이저 수준 리프트 앤 시프트) - 새 하드웨어를 구입하거나, 애플리케이션을 다시 작성하거나, 기존 작업을 수정하지 않고 인프라를 클라우드로 이동합니다. 서버를 온프레미스 플랫폼에서 동일한 플랫폼의 클라우드 서비스로 마이그레이션합니다. 예: Microsoft Hyper-V 애플리케이션을 AWS로 마이그레이션합니다.

. Retain (revisit) - Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.

. 유지(다시 방문) - 원본 환경에 애플리케이션을 유지합니다. 여기에는 주요 리팩터링이 필요하고 해당 작업을 나중으로 연기하려는 애플리케이션과 마이그레이션에 대한 비즈니스 정당성이 없기 때문에 유지하려는 레거시 애플리케이션이 포함될 수 있습니다.

. Retire - Decommission or remove applications that are no longer needed in your source environment.

. 사용 중지 - 원본 환경에서 더 이상 필요하지 않은 애플리케이션을 서비스 해제하거나 제거합니다.

A

A

ABAC

아박

See attribute-based access control. abstracted services

속성 기반 액세스 제어를 참조하십시오. 추상화된 서비스

See managed services. ACID

관리 서비스를 참조하세요. 산

See atomicity, consistency, isolation, durability. active-active migration

원자성, 일관성, 격리, 내구성을 참조하세요. 활성-활성 마이그레이션

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method

supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

원본 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되고(양방향 복제 도구 또는 이중 쓰기 작업 사용) 두 데이터베이스 모두 마이그레이션 중에 연결된 애플리케이션의 트랜잭션을 처리하는 데이터베이스 마이그레이션 방법입니다. 이 방법은 일회성 컷오프를 요구하는 대신 작고 제어된 일괄 처리로 마이그레이션을 지원합니다. 더 유연하지만 활성-수동 마이그레이션보다 더 많은 작업이 필요합니다.

active-passive migration

액티브-파시브 마이그레이션

A database migration method in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

원본 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되지만 데이터가 동기화되는 동안 원본 데이터베이스만 응용 프로그램 연결의 트랜잭션을 처리하는 데이터베이스 마이그레이션 방법입니다. 대상 데이터베이스에 복제됩니다. 대상 데이터베이스는 동안 트랜잭션을 허용하지 않습니다. 이주.

aggregate function

집계 함수

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include SUM and MAX.

행 그룹에 대해 작동하고 그룹에 대한 단일 반환 값을 계산하는 SQL 함수입니다. 집계 함수의 예로는 SUM 및 MAX가 있습니다.

AI See artificial intelligence.

알 인공 지능을 참조하십시오.

AIOps See artificial intelligence operations.

AIOps 인공 지능 작업을 참조하세요.

anonymization

익명화

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data..

데이터 세트에서 개인 정보를 영구적으로 삭제하는 프로세스입니다. 익명화는 개인 정보를 보호하는 데 도움이 될 수 있습니다. 익명화된 데이터는 더 이상 개인 데이터로 간주되지 않습니다..

anti-pattern

안티 패턴

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

솔루션이 비생산적이거나 비효율적이거나 대안보다 덜 효과적인 반복되는 문제에 대해 자주 사용되는 솔루션입니다.

application control

애플리케이션 제어

A security approach that allows the use of only approved applications in order to help protect a system from malware.

멀웨어로부터 시스템을 보호하기 위해 승인된 애플리케이션만 사용할 수 있도록 허용하는 보안 접근 방식입니다.

application portfolio

애플리케이션 포트폴리오

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized..

다음을 포함하여 조직에서 사용하는 각 애플리케이션에 대한 자세한 정보 모음입니다. 애플리케이션을 구축하고 유지 관리하는 비용과 비즈니스 가치. 이 정보는 포트폴리오 검색 및 분석 프로세스의 핵심이며 마이그레이션, 현대화 및 최적화할 애플리케이션을 식별하고 우선 순위를 지정하는 데 도움이 됩니다.

artificial intelligence (AI)

인공 지능(AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence? Artificial intelligence operations (AIOps)

컴퓨팅 기술을 사용하여 학습, 해결과 같이 일반적으로 인간과 관련된 인지 기능을 수행하는 데 전념하는 컴퓨터 과학 분야입니다. 문제 및 패턴 인식. 자세한 내용은 인공 지능이란? Artificial Intelligence Operations(AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the Aws migration strategy, see the operations integration guide.

기계 학습 기술을 사용하여 운영 문제를 해결하고, 운영 사고 및 인적 개입을 줄이고, 서비스 품질을 높이는 프로세스입니다. AWS 마이그레이션 전략에서 AIOps를 사용하는 방법에 대한 자세한 내용은 운영 통합 안내서를 참조하세요.

asymmetric encryption

비대칭 암호화

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

한 쌍의 키, 암호화를 위한 공개 키 및 암호 해독을 위한 개인 키를 사용하는 암호화 알고리즘입니다. 암호 해독에 사용되지 않으므로 공개 키를 공유할 수 있지만 개인 키에 대한 액세스는 엄격하게 제한되어야 합니다.

atomicity, consistency, isolation, durability (ACID)

원자성, 일관성, 절연성, 내구성(ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

오류, 정전 또는 기타 문제의 경우에도 데이터베이스의 데이터 유효성 및 운영 안정성을 보장하는 소프트웨어 속성 집합입니다.

attribute-based access control (ABAC)

속성 기반 액세스 제어(ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for Aws in the Aws Identity and Access Management (IAM) documentation.

부서, 직무 역할 및 팀 이름과 같은 사용자 속성을 기반으로 세분화된 권한을 생성하는 관행입니다. 자세한 내용은 AWS 자격 증명 및의 Aws에 대한 ABAC를 참조하세요. 액세스 관리(IAM) 설명서.

authoritative data source

신뢰할 수 있는 데이터 원본

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

가장 신뢰할 수 있는 정보 원본으로 간주되는 데이터의 기본 버전을 저장하는 위치입니다. 익명화, 수정 또는 가명화와 같은 데이터를 처리하거나 수정할 목적으로 신뢰할 수 있는 데이터 원본에서 다른 위치로 데이터를 복사할 수 있습니다.

Availability Zone

가용 영역

A distinct location within an Aws Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

다른 가용성의 장애로부터 격리된 AWS 리전 내의 고유한 위치입니다. 영역을 지정하고 동일한 리전의 다른 가용 영역에 대한 저렴하고 자연 시간이 짧은 네트워크 연결을 제공합니다.

AWS Cloud Adoption Framework (Aws CAF)

AW 클라우드 채택 프레임워크(AWS CAF)

A framework of guidelines and best practices from Aws to help organizations develop an efficient and effective plan to move successfully to the cloud. Aws CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, Aws CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the Aws CAF website an the AwS CAF whitepaper.

조직이 클라우드로 성공적으로 전환하기 위한 효율적이고 효과적인 계획을 개발하는 데 도움이 되는 AWS의 지침 및 모범 사례 프레임워크입니다. AWS CAF는 비즈니스, 사람, 거버넌스, 플랫폼, 보안이라는 6가지 초점 영역으로 지침을 구성합니다. 및 운영. 비즈니스, 사람 및 거버넌스 관점은 비즈니스 기술과 프로세스에 중점을 둡니다. 플랫폼, 보안 및 운영 관점은 기술 및 프로세스에 중점을 둡니다. 예를 들어, 사람 관점은 인적 자원(HR), 인력 기능 및 인력 관리를 담당하는 이해 관계자를 대상으로 합니다. 이러한 관점을 위해 AWS CAF는 조직이 성공적인 클라우드 채택을 준비할 수 있도록 인력 개발, 교육 및 커뮤니케이션에 대한 지침을 제공합니다. 자세한 내용은 Aws CAF 웹 사이트 및 AwS CAF 백서를 참조하세요.

Aws Workload Qualification Framework (Aws wQF)

AWS 워크로드 자격 프레임워크(AWS wQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AwS wQF is included with Aws Schema Conversion Tool (Aws ScT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports..

데이터베이스 마이그레이션 워크로드를 평가하고, 마이그레이션 전략을 권장하고, 작업 견적을 제공하는 도구입니다. AwS wQF는 Aws Schema Conversion Tool(Aws ScT)에 포함되어 있습니다. 데이터베이스 스키마 및 코드 객체, 애플리케이션 코드, 종속성 및 성능 특성을 분석하고 평가 보고서를 제공합니다.

B

B

bad bot

나쁜 봇

A bot that is intended to disrupt or cause harm to individuals or organizations. BCP

개인 또는 조직을 방해하거나 해를 끼치려는 봇입니다. BCP

See business continuity planning. behavior graph

비즈니스 연속성 계획을 참조하십시오. 동작 그래프

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

시간 경과에 따른 리소스 동작 및 상호 작용에 대한 통합된 대화형 보기입니다. Amazon Detective와 함께 동작 그래프를 사용하여 실패한 로그온 시도, 의심스러운 API를 검사할 수 있습니다. 호출 및 유사한 작업을 수행할 수 있습니다. 자세한 내용은 Detective 설명서의 동작 그래프의 데이터를 참조하세요.

big-endian system

빅엔디안 시스템

A system that stores the most significant byte first. See also endianness. binary classification

가장 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하십시오. 이진 분류

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as "Is this email spam or not spam?" or "Is this product a book or a car?"

이진 결과(가능한 두 가지 클래스 중 하나)를 예측하는 프로세스입니다. 예를 들어 ML 모델은 "이 이메일은 스팸입니까, 스팸이 아닌가?" 또는 "이 제품은 책입니까 아니면 자동차입니까?"

bloom filter

블룸 필터

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

요소가 집합의 멤버인지 여부를 테스트하는 데 사용되는 확률적이고 메모리 효율적인 데이터 구조입니다.

blue/green deployment

블루/그린 배포

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact. bot

두 개의 별개이지만 동일한 환경을 만드는 배포 전략입니다. 를 실행합니다. 한 환경의 현재 애플리케이션 버전(파란색)과 다른 환경의 새 애플리케이션 버전(녹색)을 사용합니다. 이 전략은 최소한의 영향으로 신속하게 롤백하는 데 도움이 됩니다. 봇

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as bad bots, are intended to disrupt or cause harm to individuals or organizations..

인터넷을 통해 자동화된 작업을 실행하고 인간을 시뮬레이션하는 소프트웨어 응용 프로그램입니다. 활동 또는 상호 작용. 인터넷의 정보를 색인화하는 웹 크롤러와 같은 일부 봇은 유용하거나 유익합니다. 나쁜 봇으로 알려진 일부 다른 봇은 방해를 받거나 방해하기 위한 것입니다. 개인이나 조직에 해를 끼치는 행위..

botnet

봇넷

Networks of bots that are infected by malware and are under the control of a single party., known as a bot herder or bot operator. Botnets are the best-known mechanism to scale bots and their impact.

맬웨어에 감염되어 단일 당사자의 통제하에 있는 봇 네트워크입니다. 봇 헤더 또는 봇 운영자로 알려져 있습니다. 봇넷은 봇과 그 영향을 확장하는 가장 잘 알려진 메커니즘입니다.

branch

가지

A contained area of a code repository. The first branch created in a repository is the main branch. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a feature branch. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

코드 리포지토리의 포함된 영역입니다. 리포지토리에서 만든 첫 번째 분기는 기본 분기입니다. 기존 분기에서 새 분기를 만든 다음 새 분기에서 기능을 개발하거나 버그를 수정할 수 있습니다. 기능을 빌드하기 위해 만드는 분기는 일반적으로 다음과 같습니다. 기능 분기라고 합니다. 기능을 릴리스할 준비가 되면 기능을 병합합니다. 다시 main 분기로 분기합니다. 자세한 내용은 분기 정보(GitHub 설명서)를 참조하세요.

break-glass access

유리 차단 접근

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스할 수 있는 권한이 없는 AWS 계정에 액세스할 수 있는 빠른 방법입니다. 자세한 내용은 AWS Well-Architected 지침의 Break-glass procedures 구현 표시기를 참조하세요.

brownfield strategy

브라운필드 전략

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

환경의 기존 인프라. a. 시스템 아키텍처에 대한 브라운필드 전략을 채택할 때 현재 시스템 및 인프라의 제약 조건을 중심으로 아키텍처를 설계합니다. 기존 인프라를 확장하는 경우 브라운필드 및 그린필드 전략을 혼합할 수 있습니다.

buffer cache

버퍼 캐시

The memory area where the most frequently accessed data is stored. business capability

가장 자주 액세스하는 데이터가 저장되는 메모리 영역입니다. 비즈니스 기능

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

비즈니스가 가치를 창출하기 위해 수행하는 작업(예: 영업, 고객 서비스 또는 마케팅). 마이크로서비스 아키텍처 및 개발 의사 결정은 비즈니스 기능에 의해 주도될 수 있습니다. 자세한 내용은 AWS에서 컨테이너화된 마이크로서비스 실행 백서의 비즈니스 기능을 중심으로 구성 섹션을 참조하세요.

business continuity planning (BCP)

비즈니스 연속성 계획(BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

대규모 마이그레이션과 같은 중단 이벤트가 운영에 미치는 잠재적 영향을 해결하고 비즈니스가 신속하게 운영을 재개할 수 있도록 하는 계획입니다.

C

C

CAF

카페

See Aws Cloud Adoption Framework. canary deployment

AWS 클라우드 채택 프레임워크를 참조하세요. 카나리아 배포

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety..

최종 사용자에게 버전을 느리고 점진적으로 릴리스합니다. 확신이 있을 때 새 버전을 배포하고 현재 버전 전체를 바꿉니다.

CCoE

CCoE

See Cloud Center of Excellence. CDC

클라우드 우수 센터를 참조하십시오. CDC

See change data capture. change data capture (CDC)

변경 데이터 캡처를 참조하세요. 변경 데이터 캡처(CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

데이터베이스 테이블과 같은 데이터 원본에 대한 변경 사항을 추적하고 변경 내용에 대한 메타데이터를 기록하는 프로세스입니다. 동기화를 유지하기 위해 대상 시스템의 변경 사항을 감사하거나 복제하는 등 다양한 용도로 CDC를 사용할 수 있습니다.

chaos engineering

카오스 엔지니어링

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (Aws FIS) to perform experiments that stress your AwS workloads and evaluate their response.

시스템의 복원력을 테스트하기 위해 의도적으로 장애 또는 중단 이벤트를 도입합니다. AWS Fault Injection Service(AWS FIS)를 사용하여 AwS 워크로드에 스트레스를 주는 실험을 수행하고 응답을 평가할 수 있습니다.

CI/CD

CI/CD

See continuous integration and continuous delivery. classification

지속적 통합 및 지속적 전달을 참조하세요. 분류

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image..

예측을 생성하는 데 도움이 되는 분류 프로세스입니다. 분류 문제에 대한 ML 모델은 이산 값을 예측합니다. 이산 값은 항상 서로 구별됩니다. 예를 들어 모델은 이미지에 자동차가 있는지 여부를 평가해야 할 수 있습니다.

client-side encryption

클라이언트 측 암호화

Encryption of data locally, before the target Aws service receives it.

대상 AWS 서비스가 데이터를 수신하기 전에 로컬에서 데이터를 암호화합니다.

Cloud Center of Excellence (CCoE)

클라우드 우수 센터(CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the Aws Cloud Enterprise Strategy Blog.

클라우드 모범 사례 개발, 리소스 동원, 마이그레이션 일정 설정, 대규모 혁신을 통해 조직 주도 등 조직 전체의 클라우드 채택 노력을 추진하는 다분야 팀입니다. 자세한 내용은 AWS Cloud Enterprise Strategy 블로그의 CCoE 게시물을 참조하세요.

cloud computing

클라우드 컴퓨팅

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

일반적으로 원격 데이터 저장 및 IoT 장치 관리에 사용되는 클라우드 기술입니다. 클라우드 컴퓨팅은 일반적으로 엣지 컴퓨팅 기술과 연결됩니다.

cloud operating model

클라우드 운영 모델

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model. oud stages of adoption

IT 조직에서 하나 이상의 클라우드 환경을 구축, 성숙 및 최적화하는 데 사용되는 운영 모델입니다. 자세한 내용은 클라우드 운영 모델 구축을 참조하세요. OUD 챕터 단계

The four phases that organizations typically go through when they migrate to the Aws Cloud:

조직이 AWS 클라우드로 마이그레이션할 때 일반적으로 거치는 네 가지 단계:

. Project - Running a few cloud-related projects for proof of concept and learning purposes

. 프로젝트 - 개념 증명 및 학습 목적으로 몇 가지 클라우드 관련 프로젝트 실행

: Foundation - Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model).

: 기초 - 클라우드 채택을 확장하기 위한 기본 투자(예: 랜딩 존 만들기, CCoE 정의, 운영 모델 설정).

. Migration - Migrating individual applications.

. 마이그레이션 - 개별 애플리케이션을 마이그레이션합니다.

: Re-invention - Optimizing products and services, and innovating in the cloud.

: 재창조 - 제품과 서비스를 최적화하고 클라우드에서 혁신합니다.

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the Aws Cloud Enterprise Strategy blog. For information about how they relate to the Aws migration strategy, see the migration readiness guide..

이러한 단계는 Stephen Orban의 Aws Cloud Enterprise Strategy 블로그의 게시물 The Journey Toward Cloud-First 및 Stages of Adoption에서 정의했습니다. AWS 마이그레이션 전략과 어떤 관련이 있는지에 대한 자세한 내용은 마이그레이션 준비 안내서를 참조하세요.

CMDB

CMDB

See configuration management database, code repository

구성 관리 데이터베이스를 참조하십시오. 코드 저장소

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a branch. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/cD pipeline can use multiple repositories.

소스 코드 및 기타 자산(예: 문서, 샘플 및 스크립트)이 저장되고 버전 제어 프로세스를 통해 업데이트되는 위치입니다. 일반적인 클라우드 리포지토리에는 GitHub 또는 Bitbucket Cloud가 포함됩니다. 코드의 각 버전을 분기라고 합니다. 마이크로서비스 구조에서 각 리포지토리는 단일 기능 전용입니다. 단일 CI/cD 파이프라인은 여러 리포지토리를 사용할 수 있습니다.

cold cache

콜드 캐시

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

비어 있거나, 잘 채워지지 않았거나, 부실하거나 관련 없는 데이터를 포함하는 버퍼 캐시입니다. 이는 데이터베이스 인스턴스가 주 메모리 또는 디스크에서 읽어야 하기 때문에 성능에 영향을 미치며, 이는 버퍼 캐시에서 읽는 것보다 느립니다.

cold data

콜드 데이터

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

거의 액세스되지 않고 일반적으로 기록적인 데이터입니다. 이러한 종류의 데이터를 쿼리할 때 일반적으로 느린 쿼리가 허용됩니다. 이 데이터를 성능이 낮고 비용이 저렴한 스토리지 계층 또는 클래스로 이동하면 비용을 줄일 수 있습니다.

computer vision (CV)

컴퓨터 비전(CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

기계 학습을 사용하여 디지털 이미지 및 비디오와 같은 시각적 형식에서 정보를 분석하고 추출하는 AI 분야입니다. 예를 들어 Amazon SageMaker AI은 이미지를 제공합니다. CV에 대한 처리 알고리즘.

configuration drift**구성 드리프트**

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional..

워크로드의 경우 예상 상태에서 구성이 변경됩니다. 이로 인해 워크로드가 비준수될 수 있으며 일반적으로 점진적이고 의도하지 않습니다..

configuration management database (CMDB)**구성 관리 데이터베이스(CMDB)**

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration..

하드웨어 및 소프트웨어 구성 요소와 해당 구성은 포함하여 데이터베이스 및 IT 환경에 대한 정보를 저장하고 관리하는 저장소입니다. 일반적으로 사용합니다. 마이그레이션의 포트폴리오 검색 및 분석 단계에서 CMDB의 데이터..

conformance pack**적합성 팩**

A collection of Aws Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AwS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the Aws Config documentation..

규정 준수 및 보안 검사를 사용자 지정하기 위해 조합할 수 있는 Aws Config 규칙 및 문제 해결 작업의 모음입니다. 적합성 팩을 단일 엔터티로 배포할 수 있습니다. AwS 계정 및 리전 또는 조직 전체에 YAML 템플릿을 사용합니다. 자세한 내용은 Aws Config 설명서의 적합성 팩을 참조하세요.

continuous integration and continuous delivery (CI/CD)**지속적 통합 및 지속적 전달(CI/CD)**

The process of automating the source, build, test, staging, and production stages of the.

소스, 빌드, 테스트, 스테이징 및 프로덕션 단계를 자동화하는 프로세스입니다.

software release process. CI/CD is commonly described as a pipeline. CI/CD can help you.

소프트웨어 릴리스 프로세스. CI/CD는 일반적으로 파이프라인으로 설명됩니다. CI/CD가 도움이 될 수 있습니다.

automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for continuous deployment.

프로세스를 자동화하고, 생산성을 개선하고, 코드 품질을 개선하고, 더 빠르게 제공합니다. 자세한 내용은 지속적인 업데이트의 이점을 참조하세요. CD는 지속적인 배포를 의미할 수도 있습니다.

For more information, see Continuous Delivery vs. Continuous Deployment..

자세한 내용은 지속적 전달과 지속적 배포를 참조하십시오.

CV**코로나 바이러스**

See computer vision.

컴퓨터 비전을 참조하십시오.

D

D

data at rest

미사용 데이터

Data that is stationary in your network, such as data that is in storage. data classification

스토리지에 있는 데이터와 같이 네트워크에서 고정된 데이터입니다. 데이터 분류

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the Aws Well-Architected Framework. For more information, see [Data classification](#).

중요도와 민감도에 따라 네트워크의 데이터를 식별하고 분류하는 프로세스입니다. 데이터에 대한 적절한 보호 및 보존 제어를 결정하는데 도움이 되므로 사이버 보안 위험 관리 전략의 중요한 구성 요소입니다. 데이터 분류는 Aws Well-Architected Framework의 보안 핵심 요소입니다. 자세한 내용은 데이터 분류 섹션을 참조하세요.

data drift

데이터 드리프트

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions..

프로덕션 데이터와 ML 모델을 훈련하는 데 사용된 데이터 간의 의미 있는 변화 또는 시간 경과에 따른 입력 데이터의 의미 있는 변화. 데이터 드리프트는 전체를 줄일 수 있습니다. ML 모델 예측의 품질, 정확성 및 공정성..

data in transit

전송 중인 데이터

Data that is actively moving through your network, such as between network resources. data mesh

네트워크 리소스 간과 같이 네트워크를 통해 활발하게 이동하는 데이터입니다. 데이터 메시

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

중앙 집중식 관리 및 거버넌스를 통해 분산된 분산형 데이터 소유권을 제공하는 아키텍처 프레임워크입니다.

data minimization

데이터 최소화

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the Aws Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

꼭 필요한 데이터만 수집하고 처리하는 원칙. 연습. AWS 클라우드의 데이터 최소화는 개인 정보 보호 위험, 비용 및 분석 탄소 발자국을 줄일 수 있습니다.

data perimeter

데이터 경계

A set of preventive guardrails in your Aws environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see [Building a data perimeter on Aws..](#)

신뢰할 수 있는 자격 증명만 예상 네트워크에서 신뢰할 수 있는 리소스에 액세스하도록 하는 데 도움이 되는 AWS 환경의 예방 가드레일 세트입니다. 자세한 내용은 Aws에서 데이터 경계 구축 섹션을 참조하세요.

data preprocessing

데이터 전처리

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

원시 데이터를 ML 모델에서 쉽게 구문 분석할 수 있는 형식으로 변환합니다. 데이터 전처리는 특정 열이나 행을 제거하고 누락되거나 일관되지 않거나 중복된 값을 해결하는 것을 의미할 수 있습니다.

data provenance

데이터 출처

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

데이터 생성, 전송 및 저장 방법과 같이 수명 주기 전반에 걸쳐 데이터의 출처와 기록을 추적하는 프로세스입니다.

data subject

데이터 주체

An individual whose data is being collected and processed. data warehouse

데이터가 수집 및 처리되는 개인입니다. 데이터 웨어하우스

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

분석과 같은 비즈니스 인텔리전스를 지원하는 데이터 관리 시스템입니다. 데이터 웨어하우스에는 일반적으로 많은 양의 기록 데이터가 포함되어 있으며 일반적으로 쿼리 및 분석에 사용됩니다.

database definition language (DDL)

데이터베이스 정의 언어(DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

데이터베이스에서 테이블 및 개체의 구조를 만들거나 수정하기 위한 문 또는 명령입니다.

database manipulation language (DML)

데이터베이스 조작 언어(DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

데이터베이스의 정보를 수정(삽입, 업데이트 및 삭제)하기 위한 문 또는 명령입니다.

DDL

DDL

See database definition language. deep ensemble

데이터베이스 정의 언어를 참조하십시오. 딥 앙상블

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions..

예측을 위해 여러 딥 러닝 모델을 결합합니다. 딥 양상블을 사용하여 보다 정확한 예측을 얻거나 예측의 불확실성을 추정할 수 있습니다.

deep learning

딥 러닝

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

여러 계층의 인공 신경망을 사용하여 입력 데이터와 관심 대상 변수 간의 매핑을 식별하는 ML 하위 필드입니다.

defense-in-depth

심층 방어

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption..

기밀성, 무결성을 보호하기 위해 일련의 보안 메커니즘 및 제어 기능을 컴퓨터 네트워크 전체에 신중하게 계층화하는 정보 보안 접근 방식입니다. 네트워크 및 내부 데이터의 가용성. AWS에서 이 전략을 채택하면 도움을 주기 위해 AWS Organizations 구조의 여러 계층에 여러 컨트롤러를 추가합니다. 리소스 보안. 예를 들어, 심층 방어 접근 방식은 다단계 인증, 네트워크 세분화 및 암호화를 결합할 수 있습니다.

delegated administrator

위임된 관리자

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the delegated administrator for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

AWS Organizations에서 호환되는 서비스는 AWS 멤버 계정을 등록하여 조직의 계정을 관리하고 해당 서비스에 대한 권한을 관리할 수 있습니다. 이 계정을 해당 서비스에 대한 위임된 관리자라고 합니다. 자세한 내용과 호환되는 서비스 목록은 AWS Organizations 설명서의 AWS Organizations와 함께 작동하는 서비스를 참조하세요.

deployment

전개

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments..

대상에서 애플리케이션, 새 기능 또는 코드 수정사항을 사용할 수 있도록 하는 프로세스입니다. 환경. 배포에는 코드 베이스에서 변경 사항을 구현한 다음 애플리케이션 환경에서 해당 코드 베이스를 빌드하고 실행하는 작업이 포함됩니다.

development environment

개발 환경

See environment detective control

환경 탐정 제어 참조

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in Implementing security controls on AWS.

이벤트가 발생한 후 탐지, 기록 및 경고하도록 설계된 보안 제어입니다. 이러한 제어는 두 번째 방어선으로, 예방적 제어를 우회한 보안 이벤트에 대해 경고합니다. 자세한 내용은 AWS에서 보안 제어 구현의 탐지 제어 섹션을 참조하세요.

development value stream mapping (DVSM)

개발 가치 흐름 매핑(DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVsM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create, and move value through the software development process..

소프트웨어 개발 수명 주기에서 속도와 품질에 부정적인 영향을 미치는 제약 조건을 식별하고 우선 순위를 지정하는 데 사용되는 프로세스입니다. DVsM은 원래 가치 흐름 프로세스를 확장합니다. 린 제조 관행을 위해 설계되었습니다. 만드는 데 필요한 단계와 팀에 중점을 둡니다. 소프트웨어 개발 프로세스를 통해 가치를 이동합니다..

digital twin

디지털 트윈

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

건물, 공장, 산업 장비 또는 생산 라인과 같은 실제 시스템의 가상 표현입니다. 디지털 트윈은 예측 유지 관리, 원격 모니터링 및 생산 최적화를 지원합니다.

dimension table

차원 테이블

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

스타 스키마에서 팩트 테이블의 정량적 데이터에 대한 데이터 속성을 포함하는 더 작은 테이블입니다. 차원 테이블 속성은 일반적으로 동작하는 텍스트 필드 또는 이산 숫자입니다. 텍스트와 같습니다. 이러한 속성은 일반적으로 쿼리 제한, 필터링 및 결과 집합에 사용됩니다. 라벨.

disaster

재난

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack..

워크로드 또는 시스템이 기본 비즈니스 목표를 달성하지 못하게 하는 이벤트입니다. 배포된 위치. 이러한 이벤트는 자연 재해, 기술적 오류 또는 의도하지 않은 잘못된 구성 또는 멀웨어 공격과 같은 인간 행동의 결과일 수 있습니다.

disaster recovery (DR)

재해 복구(DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on Aws: Recovery in the Cloud in the Aws Well-Architected Framework..

재해로 인한 가동 중지 시간 및 데이터 손실을 최소화하는 데 사용하는 전략 및 프로세스입니다. 자세한 내용은 AWS의 워크로드 재해 복구: 클라우드에서의 복구를 참조하세요. Aws Well-Architected Framework..

DML

DML

See database manipulation language. domain-driven design

데이터베이스 조작 언어를 참조하십시오. 도메인 기반 설계

An approach to developing a complex software system by connecting its components to, evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

구성 요소를 연결하여 복잡한 소프트웨어 시스템을 개발하는 접근 방식입니다. 각 구성 요소가 제공하는 진화하는 도메인 또는 핵심 비즈니스 목표. 이 개념은 다음과 같습니다. Eric Evans가 그의 저서 Domain-Driven Design: Tackling Complexity in the Heart of Software(Boston: Addison-Wesley Professional, 2003)에서 소개했습니다. 스트랭글러 무화과 패턴과 함께 도메인 기반 설계를 사용하는 방법에 대한 자세한 내용은 컨테이너 및 Amazon API Gateway를 사용하여 레거시 Microsoft ASP.NET(ASMX) 웹 서비스 증분 현대화 단원을 참조하십시오.

DR

박사

See disaster recovery.

재해 복구를 참조하십시오.

drift detection

드리프트 감지

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

기준선 구성에서 편차를 추적합니다. 예를 들어 AWS CloudFormation을 사용하여 시스템 리소스의 드리프트를 감지하거나 AWS Control Tower를 사용하여 거버넌스 요구 사항 준수에 영향을 줄 수 있는 랜딩 존의 변경 사항을 감지할 수 있습니다.

DVSM

DVSM

See development value stream mapping.

개발 가치 흐름 매핑을 참조하세요.

E

E

EDA

EDA

See [exploratory data analysis EDI](#)

탐색적 데이터 분석 EDI 보기

See [electronic data interchange. edge computing](#)

전자 데이터 교환을 참조하십시오. 에지 컴퓨팅

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

IoT 네트워크 에지에서 스마트 장치의 컴퓨팅 성능을 높이는 기술입니다. 클라우드 컴퓨팅과 비교할 때 엣지 컴퓨팅은 통신 대기 시간을 줄이고 응답 시간을 개선할 수 있습니다.

[electronic data interchange \(EDI\)](#)

전자 데이터 교환(EDI)

The automated exchange of business documents between organizations. For more information, see [What is Electronic Data Interchange](#).

조직 간 비즈니스 문서의 자동 교환. 자세한 내용은 전자 데이터 교환이란?

[encryption](#)

암호화

A computing process that transforms plaintext data, which is human-readable, into ciphertext. [encryption key](#)

사람이 읽을 수 있는 일반 텍스트 데이터를 암호문으로 변환하는 컴퓨팅 프로세스입니다. 암호화 키

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

암호화 알고리즘에 의해 생성되는 임의의 비트의 암호화 문자열입니다. 키의 길이는 다양할 수 있으며 각 키는 예측할 수 없고 고유하도록 설계되었습니다.

[endianness](#)

엔디안(endianness)

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first..

바이트가 컴퓨터 메모리에 저장되는 순서입니다. 빅 엔디안 시스템은 가장 중요한 바이트를 먼저 저장합니다. 리틀 엔디안 시스템은 가장 중요한 바이트를 먼저 저장합니다..

endpoint

끝점

See [service endpoint](#).

서비스 엔드포인트를 참조하세요.

[endpoint service](#)

엔드포인트 서비스

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

Virtual Private Cloud(VPC)에서 호스팅하여 다른 사용자와 공유할 수 있는 서비스입니다. AWS PrivateLink를 사용하여 엔드포인트 서비스를 생성하고 다른 AWS 계정 또는 AWS Identity and Access Management(IAM) 보안 주체에 권한을 부여할 수 있습니다. 이러한 계정 또는 보안 주체. 인터페이스 VPC 엔드포인트를 생성하여 엔드포인트 서비스에 비공개로 연결할 수 있습니다. 자세한 내용은 Amazon Virtual Private Cloud(Amazon VPC) 설명서의 엔드포인트 서비스 생성을 참조하세요.

enterprise resource planning (ERP)

전사적 자원 관리(ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

기업의 주요 비즈니스 프로세스(예: 회계, MES 및 프로젝트 관리)를 자동화하고 관리하는 시스템입니다.

envelope encryption

봉투 암호화

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

암호화 키를 다른 암호화 키로 암호화하는 프로세스입니다. 자세한 내용은 AWS Key Management Service(AWS KMS) 설명서의 봉투 암호화를 참조하세요.

environment

환경

An instance of a running application. The following are common types of environments in cloud computing:

실행 중인 애플리케이션의 인스턴스입니다. 다음은 클라우드 컴퓨팅의 일반적인 환경 유형입니다.

. development environment - An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a test environment.

. 개발 환경 - 애플리케이션 유지 관리를 담당하는 핵심 팀만 사용할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. 개발 환경은 상위 환경으로 승격하기 전에 변경 사항을 테스트하는 데 사용됩니다. 이러한 유형의 환경을 테스트 환경이라고도 합니다.

. lower environments - All development environments for an application, such as those used for initial builds and tests..

. 하위 환경 - 초기 빌드 및 테스트에 사용되는 환경과 같이 애플리케이션의 모든 개발 환경입니다.

. production environment - An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.

. 프로덕션 환경 - 일반 사용자가 액세스할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. CI/CD 파이프라인에서 프로덕션 환경은 마지막 배치 환경입니다.

. upper environments - All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing..

. 상위 환경 - 핵심 개발 팀 이외의 사용자가 액세스할 수 있는 모든 환경입니다. 여기에는 프로덕션 환경, 사전 프로덕션 환경 및 사용자 승인 테스트를 위한 환경이 포함될 수 있습니다.

epic

서사시

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AwS. CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the Aws. migration strategy, see the program implementation guide..

애자일 방법론에서는 작업을 구성하고 우선순위를 지정하는 데 도움이 되는 기능 범주입니다. 에픽은 요구 사항 및 구현 작업에 대한 높은 수준의 설명을 제공합니다. 예를 들어 AwS. CAF 보안 에픽에는 ID 및 액세스 관리, 탐지 제어, 인프라 보안, 데이터 보호 및 사고 대응이 포함됩니다. Aws. 마이그레이션 전략의 에픽에 대한 자세한 내용은 프로그램 구현 안내서를 참조하세요.

ERP

ERP

See enterprise resource planning, exploratory data analysis (EDA)

전사적 자원 관리를 참조하세요. EDA(예비 데이터 분석)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

데이터 세트를 분석하여 주요 특성을 이해하는 프로세스입니다. 데이터를 수집하거나 집계한 다음 초기 조사를 수행하여 패턴을 찾고, 이상 징후를 감지하고, 가정을 확인합니다. EDA는 요약 통계를 계산하고 데이터 시각화를 생성하여 수행됩니다.

F

F

fact table

팩트 테이블

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

별모양 스키마의 중앙 테이블입니다. 비즈니스 운영에 대한 정량적 데이터를 저장합니다. 일반적으로 팩트 테이블에는 측정값이 포함된 열과 차원 테이블에 대한 외래 키가 포함된 열의 두 가지 유형의 열이 포함됩니다.

fail fast

빠른 실패

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

빈번하고 점진적인 테스트를 사용하여 개발 수명 주기를 줄이는 철학입니다. 이는 애자일 접근 방식의 중요한 부분입니다.

fault isolation boundary

결합 격리 경계

In the Aws Cloud, a boundary such as an Availability Zone, Aws Region, control plane, or data. plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see Aws Fault Isolation Boundaries.

AWS 클라우드에서 가용 영역, AWS 리전, 컨트롤 플레인 또는 데이터와 같은 경계입니다. 장애의 영향을 제한하고 워크로드의 복원력을 개선하는 데 도움이 되는 평면입니다. 자세한 내용은 AWS 장애 격리 경계 섹션을 참조하세요.

feature branch

기능 분기

See branch.

분기 참조.

features

기능

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line..

예측을 수행하는 데 사용하는 입력 데이터입니다. 예를 들어, 제조 컨텍스트에서 기능은 제조 라인에서 주기적으로 캡처되는 이미지일 수 있습니다.

feature importance

기능 중요도

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations. (SHAP) and integrated gradients. For more information, see Machine learning model. interpretability with Aws.

모델의 예측에 대한 기능이 얼마나 중요한지. 이는 일반적으로 Shapley Additive Explanations와 같은 다양한 기술을 통해 계산할 수 있는 숫자 점수로 표현됩니다. (SHAP) 및 통합 그라디언트. 자세한 내용은 기계 학습 모델 섹션을 참조하세요. Aws를 사용한 해석 가능성.

feature transformation

기능 변환

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the "2021-05-27 00:15:37". date into "2021" "2021" "May", "Thu", and "15" "15" you can help the learning algorithm learn nuanced patterns associated with different data components..

추가 소스로 데이터를 보강하거나, 값을 확장하거나, 단일 데이터 필드에서 여러 정보 세트를 추출하는 것을 포함하여 ML 프로세스에 맞게 데이터를 최적화합니다. 이를 통해 ML 모델은 데이터의 이점을 활용할 수 있습니다. 예를 들어 "2021-05-27 00:15:37". 날짜를 "2021" "2021" "May", "Thu"로 분류하면 "15" "15" 학습 알고리즘이 미묘한 차이를 학습하는 데 도움이 될 수 있습니다. 다양한 데이터 구성 요소와 관련된 패턴..

few-shot prompting

소수 샷 프롬프트

Providing an LLM with a small number of examples that demonstrate the task and desired.

LLM에 작업과 원하는 작업을 보여주는 소수의 예제를 제공합니다.

output before asking it to perform a similar task. This technique is an application of in-context.

유사한 작업을 수행하도록 요청하기 전에 출력합니다. 이 기술은 컨텍스트 내의 적용입니다.

learning, where models learn from examples (shots) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain.

모델이 프롬프트에 포함된 예제(샷)에서 학습하는 학습입니다. 소샷 프롬프트는 특정 형식, 추론 또는 도메인이 필요한 작업에 효과적일 수 있습니다.

knowledge. See also zero-shot prompting..

지식. 제로샷 프롬프트도 참조하십시오..

FGAC

FGAC

See fine-grained access control. fine-grained access control (FGAC)

세분화된 액세스 제어를 참조하십시오. 세분화된 액세스 제어(FGAC)

The use of multiple conditions to allow or deny an access request. flash-cut migration.

액세스 요청을 허용하거나 거부하기 위해 여러 조건을 사용합니다. 플래시 컷 마이그레이션.

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum..

단계적 접근 방식을 사용하는 대신 변경 데이터 캡처를 통해 지속적인 데이터 복제를 사용하여 가능한 한 최단 시간 내에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다..

FM

FM

See foundation model foundation model (FM)

파운데이션 모델 파운데이션 모델(FM)을 참조하십시오.

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models..

일반화된 데이터와 레이블이 지정되지 않은 데이터의 방대한 데이터 세트에 대해 훈련해 온 대규모 딥 러닝 신경망입니다. FM은 언어 이해, 텍스트 및 이미지 생성, 자연어 대화와 같은 다양한 일반 작업을 수행할 수 있습니다. 자세한 내용은 파운데이션 모델이란 무엇입니까를 참조하세요.

G

G

generative AI

생성 AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

대량의 데이터에 대해 훈련되고 간단한 텍스트 프롬프트를 사용하여 이미지, 비디오, 텍스트 및 오디오와 같은 새 콘텐츠 및 아티팩트를 생성 할 수 있는 AI 모델의 하위 집합입니다. 자세한 정보는 생성형 AI이란 무엇입니까?

leo blocking

사자자리 차단

See geographic restrictions. geographic restrictions (geo blocking)

지리적 제한을 참조하세요. 지리적 제한(지역 차단)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Amazon CloudFront에서 특정 국가의 사용자가 콘텐츠 배포에 액세스하지 못하도록 하는 옵션입니다. 허용 목록 또는 차단 목록을 사용하여 승인 및 금지된 국가를 지정할 수 있습니다. 자세한 내용은 CloudFront 설명서의 콘텐츠의 지리적 배포 제한을 참조하세요.

Gitflow workflow

Gitflow 워크플로

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

하위 환경과 상위 환경이 소스 코드 리포지토리에서 서로 다른 분기를 사용하는 접근 방식입니다. Gitflow 워크플로는 레거시로 간주되며 트렁크 기반 워크플로는 현대적이고 선호되는 접근 방식입니다.

golden image

황금 이미지

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

해당 시스템 또는 소프트웨어의 새 인스턴스를 배포하기 위한 템플릿으로 사용되는 시스템 또는 소프트웨어의 스냅샷입니다. 예를 들어, 제조 분야에서 골든 이미지를 사용하여 여러 장치에 소프트웨어를 프로비저닝할 수 있으며 장치 제조 작업의 속도, 확장성 및 생산성을 개선하는데 도움이 됩니다.

greenfield strategy

그린필드 전략

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

새로운 환경에 기존 인프라가 없습니다. 그린필드를 채택할 때. 시스템 아키텍처에 대한 전략을 사용하면 브라운필드라고도 하는 기존 인프라와의 호환성 제한 없이 모든 새로운 기술을 선택할 수 있습니다. 기존 인프라를 확장하는 경우 브라운필드와 그린필드 전략을 혼합할 수 있습니다.

guardrail

가드 레일

A high-level rule that helps govern resources, policies, and compliance across organizational.

조직 전체에서 리소스, 정책 및 규정 준수를 관리하는 데 도움이 되는 개략적인 규칙입니다.

units (OUs). Preventive guardrails enforce policies to ensure alignment to compliance standards.

단위(OU). 예방적 가드레일은 규정 준수 표준에 부합하도록 정책을 시행합니다.

They are implemented by using service control policies and IAM permissions boundaries..

서비스 제어 정책 및 IAM 권한 경계를 사용하여 구현됩니다.

Detective guardrails detect policy violations and compliance issues, and generate alerts.

탐지 가드레일은 정책 위반 및 규정 준수 문제를 감지하고 경고를 생성합니다.

for remediation. They are implemented by using Aws Config, Aws Security Hub, Amazon.

문제 해결을 위해. Aws Config, Aws Security Hub, Amazon을 사용하여 구현됩니다.

GuardDuty, Aws Trusted Advisor, Amazon Inspector, and custom Aws Lambda checks..

GuardDuty, Aws Trusted Advisor, Amazon Inspector 및 사용자 지정 AWS Lambda 검사..

H

H

HA

하

See high availability. heterogeneous database migration

고가용성을 참조하십시오. 이기종 데이터베이스 마이그레이션

Migrating your source database to a target database that uses a different database engine. (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-. architecting effort, and converting the schema can be a complex task. AWS provides Aws SCT that helps with schema conversions..

소스 데이터베이스를 다른 데이터베이스 엔진을 사용하는 대상 데이터베이스로 마이그레이션합니다. (예: Oracle에서 Amazon Aurora로). 이기종 마이그레이션은 일반적으로 재. 설계 작업 및 스키마 변환은 복잡한 작업이 될 수 있습니다. AWS는 스키마 변환에 도움이 되는 Aws SCT를 제공합니다.

high availability (HA)

고가용성(HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

발생 시 개입 없이 지속적으로 작동하는 워크로드의 기능입니다. 도전 또는 재해. HA 시스템은 자동으로 장애 조치하고, 고품질 성능을 일관되게 제공하며, 최소한의 성능으로 다양한 부하와 장애를 처리하도록 설계되었습니다. 영향.

historian modernization

역사가 현대화

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A historian is a type of database that is used to collect and store data from various sources in a factory.

제조 산업의 요구 사항을 더 잘 충족하기 위해 운영 기술(OT) 시스템을 현대화하고 업그레이드하는 데 사용되는 접근 방식입니다. 히스토리언은 공장의 다양한 소스에서 데이터를 수집하고 저장하는 데 사용되는 데이터베이스 유형입니다.

holdout data

홀드아웃 데이터

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data..

기계 학습 모델을 훈련하는 데 사용되는 데이터 세트에서 보류되는 레이블이 지정된 과거 데이터의 일부입니다. 훌드아웃 데이터를 사용하여 모델 성능을 평가할 수 있습니다. 모델 예측을 훌드아웃 데이터와 비교합니다.

homogeneous database migration

동종 데이터베이스 마이그레이션

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration. is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

소스 데이터베이스를 동일한 데이터베이스 엔진을 공유하는 대상 데이터베이스로 마이그레이션합니다(예: Microsoft SQL Server에서 Amazon RDS for SQL Server로). 동종 마이그레이션. 일반적으로 리호스팅 또는 플랫폼 변경 작업의 일부입니다. 네이티브 데이터베이스 유ти리티를 사용하여 스키마를 마이그레이션할 수 있습니다.

hot data

핫 데이터

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

실시간 데이터 또는 최근 변환 데이터와 같이 자주 액세스되는 데이터입니다. 이 데이터에는 일반적으로 빠른 쿼리 응답을 제공하기 위해 고성능 스토리지 계층 또는 클래스가 필요합니다.

hotfix

핫픽스

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

프로덕션 환경의 중요한 문제에 대한 긴급 수정입니다. 긴급성으로 인해 핫픽스는 일반적으로 일반적인 DevOps 릴리스 워크플로 외부에서 만들어집니다.

hypercare period

하이퍼케어 기간

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1-4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

컷오프 직후, 마이그레이션 팀이 문제를 해결하기 위해 클라우드에서 마이그레이션된 애플리케이션을 관리하고 모니터링하는 기간입니다. 일반적으로 이 기간은 1-4일입니다. 하이퍼케어 기간이 끝나면 마이그레이션 팀은 일반적으로 애플리케이션에 대한 책임을 클라우드 운영 팀으로 이전합니다.

IaC

아이아C

See infrastructure as code identity-based policy

코드형 인프라 ID 기반 정책 보기

A policy attached to one or more IAM principals that defines their permissions within the Aws Cloud environment.

하나 이상의 IAM 보안 주체에 연결된 정책으로, AWS 클라우드 환경 내에서 해당 권한을 정의합니다.

idle application

유튜브 프로그램

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

90일 동안 평균 CPU 및 메모리 사용량이 5%에서 20% 사이인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하거나 온-프레미스에 유지하는 것이 일반적입니다.

IIoT

IIoT

See industrial Internet of Things. immutable infrastructure

산업용 사물인터넷을 참조하십시오. 불변의 인프라

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the Aws Well-Architected Framework.

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드를 위한 새 인프라를 배포하는 모델입니다. 불변 인프라는 본질적으로 변경 가능한 인프라보다 더 일관되고 안정적이며 예측 가능합니다. 자세한 내용은 Aws Well-Architected Framework의 불변 인프라를 사용하여 배포 모범 사례를 참조하세요.

inbound (ingress) VPC

인바운드(수신) VPC

In an AwS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The Aws Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet..

AwS 다중 계정 아키텍처에서 네트워크를 수락, 검사 및 라우팅하는 VPC입니다. 애플리케이션 외부의 연결. AWS 보안 참조 아키텍처에서는 인바운드, 아웃바운드 및 검사 VPC를 사용하여 네트워크 계정을 설정하여 네트워크를 보호할 것을 권장합니다. 애플리케이션과 광범위한 인터넷 간의 양방향 인터페이스..

incremental migration

증분 마이그레이션

A cutover strategy in which you migrate your application in small parts instead of performing.

애플리케이션을 수행하는 대신 작은 부분으로 마이그레이션하는 단독형 전략입니다.

a single, full cutover. For example, you might move only a few microservices or users to the.

단일 전체 컷오프. 예를 들어 몇 개의 마이크로 서비스 또는 사용자만으로 이동할 수 있습니다.

new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This.

새 시스템을 처음에 사용할 수 있습니다. 모든 것이 제대로 작동하는지 확인한 후 레거시 시스템을 서비스 해제할 수 있을 때까지 추가 마이크로서비스 또는 사용자를 점진적으로 이동할 수 있습니다. 이.

strategy reduces the risks associated with large migrations..

전략은 대규모 마이그레이션과 관련된 위험을 줄입니다..

Industry 4.0

인더스트리 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation., analytics, and AI/ML.

2016년 Klaus Schwab이 연결성, 실시간 데이터, 자동화 등의 발전을 통한 제조 프로세스의 현대화를 지칭하기 위해 도입한 용어입니다. 분석 및 AI/ML.

infrastructure

인프라

All of the resources and assets contained within an application's environment.

애플리케이션 환경 내에 포함된 모든 리소스 및 자산입니다.

infrastructure as code (IaC)

코드형 인프라(IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

구성 파일 집합을 통해 애플리케이션의 인프라를 프로비저닝하고 관리하는 프로세스입니다. IaC는 인프라 관리를 중앙 집중화하고, 리소스를 표준화하고, 신속하게 확장하여 새로운 환경을 반복 가능하고 안정적이며 일관되게 조정할 수 있도록 설계되었습니다.

industrial Internet of Things (IIoT)

산업용 사물 인터넷(IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as. manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy..

산업 부문에서 인터넷에 연결된 센서 및 장치의 사용. 제조, 에너지, 자동차, 의료, 생명 과학 및 농업. 자세한 내용은 산업용 사물 인터넷 (IIoT) 디지털 혁신 전략 구축을 참조하세요.

inspection VPC

검사 VPC

In an Aws multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different Aws Regions), the internet, and on-premises networks. The Aws Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Aws 다중 계정 아키텍처에서 VPC(동일하거나 다른 AWS 리전), 인터넷 및 온프레미스 네트워크 간의 네트워크 트래픽 검사를 관리하는 중앙 집중식 VPC입니다. AWS 보안 참조 아키텍처에서는 애플리케이션과 광범위한 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

Internet of Things (IoT)

사물 인터넷(IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local. communication network. For more information, see What is IoT?.

인터넷 또는 로컬을 통해 다른 디바이스 및 시스템과 통신하는 내장된 센서 또는 프로세서가 있는 연결된 물리적 개체의 네트워크입니다. 통신 네트워크. 자세한 내용은 IoT란?을 참조하세요.

interpretability

해석 가능성

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with Aws.

인간이 모델의 예측이 입력에 따라 어떻게 의존하는지 이해할 수 있는 정도를 설명하는 기계 학습 모델의 특성입니다. 자세한 내용은 AWS를 사용한 기계 학습 모델 해석 가능성 섹션을 참조하세요.

IoT

IoT

See Internet of Things. IT information library (ITIL)

사물 인터넷을 참조하십시오. IT 정보 라이브러리(ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT 서비스를 제공하고 이러한 서비스를 비즈니스 요구 사항에 맞추기 위한 일련의 모범 사례입니다. ITIL은 ITSM의 기반을 제공합니다.

IT service management (ITSM)

IT 서비스 관리(ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

조직의 IT 서비스 설계, 구현, 관리 및 지원과 관련된 활동입니다. 클라우드 작업을 ITSM 도구와 통합하는 방법에 대한 자세한 내용은 운영 통합 가이드를 참조하세요.

ITIL

이틸

See IT information library. ITSM

IT 정보 라이브러리를 참조하십시오. ITSM

See IT service management.

IT 서비스 관리를 참조하십시오.

L

L

label-based access control (LBAC)

레이블 기반 액세스 제어(LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

사용자와 데이터 자체에 각각 보안 레이블 값이 명시적으로 할당되는 MAC(필수 액세스 제어)의 구현입니다. 사용자 보안 레이블과 데이터 보안 레이블 간의 교차점은 사용자가 볼 수 있는 행과 열을 결정합니다.

landing zone

착륙 구역

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

랜딩 존은 확장 가능하고 안전한 잘 설계된 다중 계정 AWS 환경입니다. 이는 조직이 신속하게 시작하고 배포할 수 있는 출발점입니다. 보안 및 인프라 환경에 대한 확신을 가진 워크로드 및 애플리케이션. 랜딩 존에 대한 자세한 내용은 안전하고 확장 가능한 다중 계정 AWS 환경 설정 섹션을 참조하세요.

large language model (LLM)

대규모 언어 모델(LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

방대한 양의 데이터에 대해 사전 훈련된 딥 러닝 AI 모델입니다. LLM은 질문에 답하고, 문서를 요약하고, 텍스트를 다른 언어로 번역하고, 문장을 완성하는 등 여러 작업을 수행할 수 있습니다. 자세한 내용은 LLM이란 무엇입니까?

large migration

대규모 마이그레이션

A migration of 300 or more servers.

300개 이상의 서버 마이그레이션.

LBAC

LBAC

See label-based access control.

레이블 기반 액세스 제어를 참조하세요.

least privilege

최소 권한

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.. lift and shift

작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 IAM 설명서의 최소 권한 권한 적용을 참조하세요. 리프트 앤 시프트

See 7 Rs. little-endian system

7 Rs. 리틀 엔디안 시스템 참조

A system that stores the least significant byte first. See also endianness. LLM

가장 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하십시오. LLM

See large language model. lower environments

대규모 언어 모델을 참조하세요. 하위 환경

See environment

환경 보기

m**m**

machine learning (ML)

기계 학습(ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning..

패턴 인식 및 학습을 위해 알고리즘과 기술을 사용하는 인공 지능 유형입니다. ML은 사물 인터넷(IoT) 데이터와 같은 기록된 데이터를 분석하고 학습하여 패턴을 기반으로 통계 모델을 생성합니다. 자세한 내용은 기계 학습을 참조하세요.

See branch. malware

분기를 참조하십시오. 악성코드

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers..

컴퓨터 보안 또는 개인 정보를 손상시키도록 설계된 소프트웨어. 맬웨어는 컴퓨터 시스템을 방해하거나 민감한 정보를 유출하거나 무단 액세스를 얻을 수 있습니다. 예. 맬웨어에는 바이러스, 웜, 랜섬웨어, 트로이 목마, 스파이웨어 및 키로거가 포함됩니다.

managed services

관리형 서비스

AwS services for which AwS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as abstracted services.

AwS가 인프라 계층, 운영 체제 및 플랫폼을 운영하는 AwS 서비스를 사용하는 경우 엔드포인트에 액세스하여 데이터를 저장하고 검색할 수 있습니다. Amazon 단순 스토리지 서비스(Amazon S3) 및 Amazon DynamoDB는 관리형 서비스의 예입니다. 이를 추상화된 서비스라고도 합니다.

manufacturing execution system (MES)

제조 실행 시스템(MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

작업 현장에서 원자재를 완제품으로 변환하는 생산 프로세스를 추적, 모니터링, 문서화 및 제어하기 위한 소프트웨어 시스템입니다.

MAP

지도

See Migration Acceleration Program. mechanism

マイグレーション 가속화 프로그램을 참조하십시오. 기구

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the Aws Well-Architected Framework.

도구를 생성하고, 도구 채택을 유도한 다음, 결과를 검사하여 조정하는 전체 프로세스입니다. 메커니즘은 작동하면서 자체적으로 강화하고 개선하는 주기입니다. 자세한 내용은 Aws Well-Architected Framework에서 메커니즘 구축 섹션을 참조하세요.

member account

회원 계정

All Aws accounts other than the management account that are part of an organization in Aws Organizations. An account can be a member of only one organization at a time..

Aws Organizations의 조직에 속한 관리 계정 이외의 모든 AWS 계정. 계정은 한 번에 하나의 조직에만 구성원이 될 수 있습니다..

MES

MES

See manufacturing execution system.

제조 실행 시스템을 참조하십시오.

Message Queuing Telemetry Transport (MQTT)

MQTT(메시지 큐 원격 분석 전송)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/ subscribe pattern, for resource-constrained IoT devices.

리소스가 제한된 IoT 디바이스를 위한 게시/구독 패턴을 기반으로 하는 경량 M2M(Machine-to-Machine) 통신 프로토콜입니다.

microservice

マイク로서비스

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using Aws serverless services.

잘 정의된 API를 통해 통신하는 소규모 독립 서비스로, 일반적으로 소규모 독립형 팀이 소유합니다. 예를 들어, 보험 시스템에는 영업 또는 마케팅과 같은 비즈니스 기능 또는 구매, 청구 또는 분석과 같은 하위 도메인에 매핑되는 마이크로서비스가 포함될 수 있습니다. 마이크로서비스의 이점에는 민첩성, 유연한 확장, 손쉬운 배포, 재사용 가능한 코드 및 복원력이 포함됩니다. 자세한 내용은 AWS 서비스 서비스를 사용하여 마이크로서비스 통합을 참조하세요.

microservices architecture

マイク로서비스 アーキテクチャ

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed,

각 애플리케이션 프로세스를 마이크로서비스로 실행하는 독립적인 구성 요소를 사용하여 애플리케이션을 빌드하는 접근 방식입니다. 이러한 마이크로서비스는 경량 API를 사용하여 잘 정의된 인터페이스를 통해 통신합니다. 이 아키텍처의 각 마이크로서비스는 업데이트, 배포,

and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on Aws.

애플리케이션의 특정 기능에 대한 수요를 충족하도록 확장됩니다. 자세한 내용은 AWS에서 마이크로서비스 구현 섹션을 참조하세요.

Migration Acceleration Program (MAP)

マイグレーション 加速化 プログラム(MAP)

An Aws program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

조직이 클라우드로 전환하기 위한 강력한 운영 기반을 구축하고 마이그레이션의 초기 비용을 상쇄하는 데 도움이 되는 컨설팅 지원, 교육 및 서비스를 제공하는 AWS 프로그램입니다. MAP에는 체계적인 방식으로 레거시 마이그레이션을 실행하기 위한 마이그레이션 방법론과 일반적인 마이그레이션 시나리오를 자동화하고 가속화하는 도구 세트가 포함되어 있습니다.

migration at scale

대규모 마이그레이션

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a migration factory of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy..

대부분의 애플리케이션 포트폴리오를 클라우드로 웨이브로 이동하는 프로세스입니다. 각 웨이브에서 더 많은 애플리케이션이 더 빠른 속도로 이동했습니다. 이 단계에서는 이전 단계에서 배운 모범 사례와 교훈을 사용하여 팀, 도구 및 프로세스로 구성된 마이그레이션 팩토리를 구현하여 자동화 및 민첩한 제공을 통해 워크로드 마이그레이션을 간소화합니다. 이것은 AWS 마이그레이션 전략의 세 번째 단계입니다..

migration factory

마이그레이션 팩토리

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

자동화된 민첩한 접근 방식을 통해 워크로드 마이그레이션을 간소화하는 다기능 팀입니다. 마이그레이션 팩토리 팀에는 일반적으로 스프린트에서 작업하는 운영, 비즈니스 분석가 및 소유자, 마이그레이션 엔지니어, 개발자 및 DevOps 전문가가 포함됩니다. 엔터프라이즈 애플리케이션 포트폴리오의 20%에서 50% 사이는 팩토리 접근 방식으로 최적화할 수 있는 반복적인 패턴으로 구성됩니다. 자세한 내용은 이 콘텐츠 세트의 마이그레이션 팩토리 및 클라우드 마이그레이션 팩토리 가이드에 대한 설명을 참조하세요.

migration metadata

마이그레이션 메타데이터

The information about the application and server that is needed to complete the migration.. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account..

마이그레이션을 완료하는 데 필요한 애플리케이션 및 서버에 대한 정보입니다. 각 마이그레이션 패턴에는 서로 다른 마이그레이션 메타데이터 세트가 필요합니다. 마이그레이션 메타데이터의 예로는 대상 서브넷, 보안 그룹 및 AWS 계정이 있습니다.

migration pattern

마이그레이션 패턴

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

마이그레이션 전략, 마이그레이션 대상, 사용된 마이그레이션 애플리케이션 또는 서비스를 자세히 설명하는 반복 가능한 마이그레이션 작업입니다. 예: AWS Application Migration Service를 사용하여 Amazon EC2로 마이그레이션을 다시 호스팅합니다.

Migration Portfolio Assessment (MPA)

마이그레이션 포트폴리오 평가(MPA)

An online tool that provides information for validating the business case for migrating to the Aws Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AwS consultants and APN Partner consultants.

AWS 클라우드로 마이그레이션하기 위한 비즈니스 사례를 검증하기 위한 정보를 제공하는 온라인 도구입니다. MPA는 상세한 포트폴리오 평가(서버 적정 규모 조정, 요금, TCO 비교, 마이그레이션 비용 분석)와 마이그레이션 계획(애플리케이션 데이터 분석 및 데이터 수집, 애플리케이션 그룹화, 마이그레이션 우선 순위 지정 및 웨이브 계획)을 제공합니다. 그만큼. MPA 도구(로그인 필요)는 모든 AwS 컨설턴트 및 APN 파트너 컨설턴트가 무료로 사용할 수 있습니다.

Migration Readiness Assessment (MRA)

마이그레이션 준비 상태 평가(MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the Aws CAF. For more information, see the migration readiness guide. MRA is the first phase of the Aws migration strategy.

AWS CAF를 사용하여 조직의 클라우드 준비 상태에 대한 통찰력을 얻고, 강점과 약점을 식별하고, 식별된 격차를 해소하기 위한 실행 계획을 수립하는 프로세스입니다. 자세한 내용은 마이그레이션 준비 안내서를 참조하세요. MRA는 AWS 마이그레이션 전략의 첫 번째 단계입니다.

migration strategy

마이그레이션 전략

The approach used to migrate a workload to the Aws Cloud. For more information, see the 7 Rs.

워크로드를 AWS 클라우드로 마이그레이션하는 데 사용되는 접근 방식입니다. 자세한 내용은 7R을 참조하세요.

entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

이 용어집의 항목을 참조하고 조직을 모빌리 대규모 마이그레이션을 가속화하는 것을 참조하세요.

ML

ML

See machine learning. modernization

기계 학습을 참조하세요. 현대화

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

오래된(레거시 또는 모놀리식) 애플리케이션과 해당 인프라를 클라우드에서 민첩하고 탄력적이며 가용성이 높은 시스템으로 전환하여 비용을 절감하고 효율성을 높이고 활용합니다. 혁신의 활용. 자세한 내용은 AWS 클라우드에서 애플리케이션을 현대화하기 위한 전략 섹션을 참조하세요.

modernization readiness assessment

현대화 준비 상태 평가

An evaluation that helps determine the modernization readiness of an organization's.

조직의 현대화 준비 상태를 결정하는 데 도움이 되는 평가입니다.

applications; identifies benefits, risks, and dependencies; and determines how well the.

응용 프로그램; 이점, 위험 및 종속성을 식별합니다. 얼마나 잘 결정합니다.

organization can support the future state of those applications. The outcome of the assessment.

조직은 이러한 애플리케이션의 향후 상태를 지원할 수 있습니다. 평가 결과입니다.

is a blueprint of the target architecture, a roadmap that details development phases and.

대상 아키텍처의 청사진, 개발 단계를 자세히 설명하는 로드맵입니다.

milestones for the modernization process, and an action plan for addressing identified gaps. For.

현대화 프로세스의 이정표 및 식별된 격차를 해결하기 위한 실행 계획. 때문에.

more information, see Evaluating modernization readiness for applications in the Aws Cloud.

자세한 내용은 AWS 클라우드의 애플리케이션에 대한 현대화 준비 상태 평가를 참조하세요.

monolithic applications (monoliths)

모놀리식 애플리케이션(모놀리식)

Applications that run as a single service with tightly coupled processes. Monolithic applications. have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can.

긴밀하게 결합된 프로세스가 있는 단일 서비스로 실행되는 애플리케이션입니다. 모놀리식 애플리케이션. 몇 가지 단점이 있습니다. 하나의 애플리케이션 기능에 수요가 급증하면 . 전체 아키텍처를 확장해야 합니다. 모놀리식 애플리케이션의 기능을 추가하거나 개선하는 것도 코드 기반이 증가할 때 더 복잡해집니다. 이러한 문제를 해결하려면 다음을 수행할 수 있습니다.

use a microservices architecture. For more information, see Decomposing monoliths into microservices.

マイ크로 서비스 아키텍처를 사용합니다. 자세한 내용은 모놀리스를 마이크로 서비스로 분해 섹션을 참조하세요.

MPA

MPA

See Migration Portfolio Assessment MQTT

マイ그레이션 포트폴리오 평가 MQTT 참조

See Message Queuing Telemetry Transport. multiclass classification

메시지 큐 원격 분석 전송을 참조하십시오. 다중 클래스 분류

A process that helps generate predictions for multiple classes (predicting one of more than. two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?".

여러 클래스에 대한 예측을 생성하는 데 도움이 되는 프로세스입니다(두 개 이상의 결과 중 하나를 예측). 예를 들어 ML 모델은 "이 제품은 책인가, 자동차 또는 휴대폰인가?" 또는 "이 고객에게 가장 흥미로운 제품 범주는 무엇입니까?"라고 질문할 수 있습니다.

mutable infrastructure

변경 가능한 인프라

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the Aws Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

프로덕션 워크로드를 위해 기존 인프라를 업데이트하고 수정하는 모델입니다. 일관성, 안정성 및 예측 가능성을 개선하기 위해 Aws Well-Architected Framework는 불변 인프라를 모범 사례로 사용할 것을 권장합니다.

O

O

OAC

OAC

See origin access control. OAI

원본 액세스 제어를 참조하십시오. 오하이

See origin access identity. OCM

원본 액세스 ID를 참조하십시오. OCM

See organizational change management. offline migration

조직 변경 관리를 참조하세요. 오프라인 마이그레이션

A migration method in which the source workload is taken down during the migration process.

마이그레이션 프로세스 중에 원본 워크로드가 중단되는 마이그레이션 방법입니다.

This method involves extended downtime and is typically used for small, non-critical workloads.

이 방법은 가동 중지 시간이 길어지며 일반적으로 중요하지 않은 작고 워크로드에 사용됩니다.

See operations integration.

운영 통합을 참조하십시오.

OLA

올라

See operational-level agreement. online migration

운영 수준 계약을 참조하십시오. 온라인 마이그레이션

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

원본 워크로드가 대상 시스템에 복사되지 않고 복사되는 마이그레이션 방법입니다. 오프라인으로 전환됩니다. 워크로드에 연결된 애플리케이션은 그 동안 계속 작동할 수 있습니다. 마이그레이션. 이 방법은 가동 중지 시간이 0에서 최소한이며 일반적으로 중요한 프로덕션 워크로드에 사용됩니다.

OPC-UA

OPC-UA

See Open Process Communications – Unified Architecture.

Open Process Communications – Unified Architecture를 참조하십시오.

Open Process Communications – Unified Architecture (OPc-UA)

개방형 프로세스 통신 – 통합 아키텍처(OPc-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPc-UA provides an interoperability standard with data encryption, authentication, and authorization. schemes.

산업 자동화를 위한 M2M(Machine-to-Machine) 통신 프로토콜입니다. OPc-UA는 데이터 암호화, 인증 및 권한 부여를 통한 상호 운용성 표준을 제공합니다. 구성표.

operational-level agreement (OLA)

운영 수준 계약(OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

SLA(서비스 수준 계약)를 지원하기 위해 기능 IT 그룹이 서로에게 제공하기로 약속하는 내용을 명확히 하는 계약입니다.

operational readiness review (ORR)

운영 준비 상태 검토(ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the Aws Well-Architected Framework..

인시던트 및 가능한 실패의 범위를 이해, 평가, 예방 또는 줄이는 데 도움이 되는 질문 및 관련 모범 사례의 체크리스트입니다. 자세한 내용은 참조하세요. AWS Well-Architected Framework의 운영 준비 검토(ORR)..

operational technology (OT)

운영 기술(OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

물리적 환경과 함께 작동하여 산업 운영, 장비 및 인프라를 제어하는 하드웨어 및 소프트웨어 시스템입니다. 제조 분야에서 OT와 정보 기술 (IT) 시스템의 통합은 인더스트리 4.0 혁신의 핵심 초점입니다.

operations integration (OI)

운영 통합(OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide. organization trail

준비 계획, 자동화 및 통합을 포함하는 클라우드에서 운영을 현대화하는 프로세스입니다. 자세한 내용은 운영 통합 가이드를 참조하세요. 조직 추적

A trail that's created by Aws CloudTrail that logs all events for all AwS accounts in an organization in Aws Organizations. This trail is created in each AwS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

Aws CloudTrail에서 생성한 추적으로, Aws Organizations의 조직에 있는 모든 AwS 계정에 대한 모든 이벤트를 기록합니다. 이 추적은 조직의 일부인 각 AwS 계정에서 생성되며 각 계정의 활동을 추적합니다. 자세한 내용은 CloudTrail 설명서의 조직에 대한 추적 생성을 참조하세요.

organizational change management (OCM)

조직 변경 관리(OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OcM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the Aws migration strategy, this framework is called people acceleration, because of the speed of change required in cloud adoption projects. For more information, see the OcM guide.

사람, 문화 및 리더십 관점에서 주요 파괴적인 비즈니스 혁신을 관리하기 위한 프레임워크입니다. OcM은 조직이 변경 채택을 가속화하고, 전환 문제를 해결하고, 문화 및 조직 변화를 주도하여 새로운 시스템과 전략을 준비하고 전환할 수 있도록 지원합니다. AWS 마이그레이션 전략에서 이 프레임워크는 클라우드 채택 프로젝트에 필요한 변화 속도 때문에 인력 가속화라고 합니다. 자세한 내용은 OcM 가이드를 참조하세요.

origin access control (OAC)

OAC(원본 액세스 제어)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage.

CloudFront에서 Amazon Simple Storage를 보호하기 위해 액세스를 제한하는 향상된 옵션입니다.

Service (Amazon S3) content. OAC supports all S3 buckets in all AwS Regions, server-side.

서비스(Amazon S3) 콘텐츠. OAC는 서버 측의 모든 AwS 리전의 모든 S3 버킷을 지원합니다.

encryption with AWS KMS (SSE-KMS), and dynamic PUT and DELETE requests to the S3 bucket.

AWS KMS(SSE-KMS)를 사용한 암호화와 S3 버킷에 대한 동적 PUT 및 DELETE 요청.

origin access identity (OAI)

오리진 액세스 ID(OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

CloudFront에서 Amazon S3 콘텐츠를 보호하기 위해 액세스를 제한하는 옵션입니다. OAI를 사용하는 경우 CloudFront는 Amazon S3가 인증할 수 있는 보안 주체를 생성합니다. 인증된 보안 주체는 특정 CloudFront 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. 보다 세분화되고 향상된 액세스 제어를 제공하는 OAC도 참조하십시오.

ORR

오르

See operational readiness review. OT

운영 준비 상태 검토를 참조하세요. 오토

See operational technology. outbound (egress) VPC

운영 기술을 참조하세요. 아웃바운드(송신) VPC

In an AwS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The Aws Security Reference Architecture recommends. setting up your Network account with inbound, outbound, and inspection VPCs to protect the. two-way interface between your application and the broader internet..

AwS 다중 계정 아키텍처에서 애플리케이션 내에서 시작되는 네트워크 연결을 처리하는 VPC입니다. AWS 보안 아키텍처를 권장합니다. 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정하여 보호합니다. 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스..

P

P

permissions boundary

권한 경계

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

사용자 또는 역할이 가질 수 있는 최대 권한을 설정하기 위해 IAM 보안 주체에 연결된 IAM 관리 정책입니다. 자세한 내용은 IAM의 권한 경계 섹션을 참조하세요. 문서조사.

personally identifiable information (PII)

개인 식별 정보(PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

직접 보거나 다른 관련 데이터와 함께 볼 때 개인의 신원을 합리적으로 추론하는 데 사용할 수 있는 정보입니다. PII의 예로는 이름, 주소 및 연락처 정보가 있습니다.

PII

PII

See personally identifiable information. playbook

개인 식별 정보를 참조하십시오. 플레이북

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

제공과 같은 마이그레이션과 관련된 작업을 캡처하는 사전 정의된 단계 집합입니다. 클라우드의 핵심 운영 기능. 플레이북은 스크립트, 자동화된 Runbook 또는 현대화된 작업을 운영하는 데 필요한 프로세스 또는 단계의 요약 형태를 취할 수 있습니다. 환경.

PLC

PLC

See programmable logic controller. PLM

프로그래머블 로직 컨트롤러를 참조하십시오. PLM

See product lifecycle management. policy

제품 수명 주기 관리를 참조하십시오. 정책

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in Aws Organizations (see service control policy).

권한을 정의하거나(자격 증명 기반 정책 참조), 액세스 조건을 지정하거나(리소스 기반 정책 참조) Aws Organizations에서 조직의 모든 계정에 대한 최대 권한을 정의할 수 있는 객체입니다(서비스 제어 정책 참조).

polyglot persistence

다국어 지속성

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor

performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store

데이터 액세스 패턴 및 기타 요구 사항에 따라 마이크로서비스의 데이터 스토리지 기술을 독립적으로 선택합니다. 마이크로서비스에 동일한 데이터 스토리지 기술이 있는 경우 가능합니다. 구현 문제가 발생하거나 성능이 저하됩니다. 마이크로서비스는 데이터 저장소를 사용하는 경우 더 쉽게 구현되고 더 나은 성능과 확장성을 달성합니다.

best adapted to their requirements. For more information, see Enabling data persistence in microservices.

요구 사항에 가장 잘 맞습니다. 자세한 내용은 마이크로서비스에서 데이터 지속성 사용을 참조하세요.

portfolio assessment

포트폴리오 평가

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness. predicate

マイグ레이션을 계획하기 위해 애플리케이션 포트폴리오를 검색, 분석 및 우선 순위를 지정하는 프로세스입니다. 자세한 내용은 마이그레이션 준비 상태 평가를 참조하세요. 술어

A query condition that returns true or false, commonly located in a WHERE clause. predicate pushdown

일반적으로 WHERE 절에 있는 true 또는 false를 반환하는 쿼리 조건입니다. 조건자 푸시다운

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

전송 전에 쿼리의 데이터를 필터링하는 데이터베이스 쿼리 최적화 기술입니다. 이렇게 하면 관계형 데이터베이스에서 검색하고 처리해야 하는 데이터의 양이 줄어들고 쿼리 성능이 향상됩니다.

preventative control

예방적 통제

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in Implementing security controls on Aws..

이벤트가 발생하지 않도록 설계된 보안 제어입니다. 이러한 제어는 네트워크에 대한 무단 액세스 또는 원치 않는 변경을 방지하는 데 도움이 되는 첫 번째 방어선입니다. 때문에, 자세한 내용은 Aws에서 보안 제어 구현의 예방적 제어를 참조하세요.

principal

원금

An entity in Aws that can perform actions and access resources. This entity is typically a root user for an Aws account, an IAM role, or a user. For more information, see Principal in Roles terms and concepts in the IAM documentation..

작업을 수행하고 리소스에 액세스할 수 있는 AWS의 엔터티입니다. 이 엔터티는 일반적으로 AWS 계정의 루트 사용자, IAM 역할 또는 사용자입니다. 자세한 내용은 IAM 설명서의 역할 용어 및 개념의 보안 주체를 참조하세요.

privacy by design

개인 정보 보호 설계

A system engineering approach that takes privacy into account through the whole development process.

전체 개발 프로세스에서 개인 정보 보호를 고려하는 시스템 엔지니어링 접근 방식입니다.

private hosted zones

프라이빗 호스팅 영역

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation..

Amazon Route 53이 하나 이상의 VPC 내의 도메인 및 해당 하위 도메인에 대한 DNS 쿼리에 응답하는 방법에 대한 정보를 보유하는 컨테이너입니다. 자세한 내용은 Route 53 설명서의 프라이빗 호스팅 영역 작업을 참조하세요.

proactive control

사전 예방적 제어

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the

규정을 준수하지 않는 리소스의 배포를 방지하도록 설계된 보안 제어입니다. 이러한 컨트롤은 프로비저닝되기 전에 리소스를 스캔합니다. 리소스가 를 준수하지 않는 경우, 컨트롤, 프로비저닝되지 않습니다. 자세한 내용은 컨트롤 참조 가이드의

Aws Control Tower documentation and see Proactive controls in Implementing security controls on AWS.

AWS Control Tower 설명서를 참조하고 AWS에서 보안 제어 구현의 사전 예방적 제어를 참조하세요.

product lifecycle management (PLM)

제품 라이프사이클 관리(PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

설계, 개발, 출시, 성장 및 성숙, 쇠퇴 및 제거에 이르기까지 전체 수명 주기에 걸쳐 제품에 대한 데이터 및 프로세스를 관리합니다.

production environment

생산 환경

See environment programmable logic controller (PLC)

환경 프로그래머블 로직 컨트롤러(PLC) 참조

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

제조업에서는 기계를 모니터링하고 제조 공정을 자동화하는 신뢰성이 높고 적응력이 뛰어난 컴퓨터입니다.

prompt chaining

프롬프트 체인

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model's responses and allows for more granular, personalized results.

한 LLM 프롬프트의 출력을 다음 프롬프트의 입력으로 사용하여 더 나은 응답을 생성합니다. 이 기술은 복잡한 작업을 하위 작업으로 나누거나 예비 응답을 반복적으로 개선하거나 확장하는 데 사용됩니다. 이는 모델 응답의 정확성과 관련성을 개선하고 보다 세분화되고 개인화된 결과를 가능하게 합니다.

pseudonymization

가명화

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

데이터 세트의 개인 식별자를 자리 표시자 값으로 바꾸는 프로세스입니다. 가명화는 개인 정보를 보호하는 데 도움이 될 수 있습니다. 가명화된 데이터는 여전히 개인 데이터로 간주됩니다.

publish/subscribe (pub/sub)

게시/구독(pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MEs, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service..

マイクロ 서비스 간의 비동기 통신을 개선할 수 있는 패턴입니다. 확장성 및 응답성. 예를 들어 마이크로 서비스 기반 ME에서 마이크로 서비스는 다른 마이크로 서비스가 구독할 수 있는 채널에 이벤트 메시지를 게시할 수 있습니다. 시스템은 할 수 있습니다. 게시 서비스를 변경하지 않고 새 마이크로 서비스를 추가합니다..

Q

Q

query plan

쿼리 계획

A series of steps, like instructions, that are used to access the data in a SQL relational database System.

SQL 관계형 데이터베이스 시스템의 데이터에 액세스하는 데 사용되는 명령과 같은 일련의 단계입니다.

query plan regression

쿼리 계획 회귀

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

데이터베이스 서비스 최적화 프로그램이 데이터베이스 환경에 대한 지정된 변경 전보다 덜 최적의 계획을 선택하는 경우. 이는 통계, 제약 조건, 환경 설정, 쿼리 매개 변수 바인딩 및 데이터베이스 엔진 업데이트에 대한 변경으로 인해 발생할 수 있습니다.

R

R

RACI matrix

RACI 매트릭스

See responsible, accountable, consulted, informed (RACI). RAG

책임감, 책임감, 상담, 정보(RACI)를 참조하세요. 네마

See Retrieval Augmented Generation.

검색 증강 생성을 참조하십시오.

ransomware

랜섬웨어

A malicious software that is designed to block access to a computer system or data until a payment is made.

결제가 이루어질 때까지 컴퓨터 시스템 또는 데이터에 대한 액세스를 차단하도록 설계된 악성 소프트웨어입니다.

RASCI matrix

RASCI 매트릭스

See responsible, accountable, consulted, informed (RAcl) RCAC

책임, 책임, 상담, 정보(RAcl) RCAC 참조

See row and column access control, read replica

행 및 열 액세스 제어를 참조하세요. 읽기 전용 복제본

A copy of a database that's used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

읽기 전용 목적으로 사용되는 데이터베이스의 복사본입니다. 쿼리를 읽기 전용 복제본으로 라우팅하여 기본 데이터베이스의 부하를 줄일 수 있습니다.

re-architect

재설계

See 7 Rs. recovery point objective (RPO)

7 Rs. 복구 지점 목표(RPO)를 참조하십시오.

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

마지막 데이터 복구 지점 이후 허용되는 최대 시간입니다. 이는 마지막 복구 지점과 서비스 중단 사이에 허용되는 데이터 손실로 간주되는 것을 결정합니다.

recovery time objective (RTO)

복구 시간 목표(RTO)

The maximum acceptable delay between the interruption of service and restoration of service. refactor

서비스 중단과 서비스 복원 사이에 허용되는 최대 지연입니다. 리팩터링

See 7 Rs. Region

7 Rs. 지역 참조

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use..

지리적 영역의 AWS 리소스 모음입니다. 각 AWS 리전은 격리되어 있습니다. 내결함성, 안정성 및 복원력을 제공하기 위해 다른 리전과 독립적입니다. 자세한 내용은 계정에서 사용할 수 있는 AWS 리전 지정 섹션을 참조하세요.

regression

회귀

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage)..

숫자 값을 예측하는 ML 기술입니다. 예를 들어, "이 집은 얼마에 팔릴까요?" 문제를 해결하기 위해 ML 모델은 선형 회귀 모델을 사용하여 집에 대한 알려진 사실(예: 평방 피트)을 기반으로 집의 판매 가격을 예측할 수 있습니다.

rehost**재호스팅**

See 7 Rs. release

7 Rs. 릴리스 참조

In a deployment process, the act of promoting changes to a production environment. relocate

배포 프로세스에서 프로덕션 환경에 대한 변경 내용을 승격하는 작업입니다. 재배치

See 7 Rs. replatform

7 Rs. 플랫폼 변경 참조

See 7 Rs. repurchase

7 Rs. repurchase 참조

See 7 Rs. resiliency

7 Rs. 복원력 참조

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the Aws Cloud. For more information, see Aws Cloud Resilience.

중단에 저항하거나 복구하는 애플리케이션의 능력입니다. 고가용성 및 재해 복구는 AWS 클라우드의 복원력을 계획할 때 일반적으로 고려 사항입니다. 자세한 내용은 AWS 클라우드 복원력을 참조하세요.

resource-based policy**리소스 기반 정책**

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

Amazon S3 버킷, 엔드포인트 또는 암호화 키와 같은 리소스에 연결된 정책입니다. 이 유형의 정책은 액세스할 수 있는 보안 주체, 지원되는 작업 및 충족해야 하는 기타 조건을 지정합니다.

responsible, accountable, consulted, informed (RAcl) matrix**책임감, 책임감, 상담, 정보(RAcl) 매트릭스**

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a RAscl matrix, and if you exclude it, it's called a RACl matrix.

マイ그레이션 활동 및 클라우드 운영과 관련된 모든 당사자의 역할과 책임을 정의하는 매트릭스입니다. 매트릭스 이름은 매트릭스에 정의된 책임 유형(책임(R), 책임(A), 상담(C) 및 정보(I))에서 파생됩니다. 지원(S) 유형은 선택 사항입니다. 지원을 포함하는 경우 매트릭스를 RAscl 매트릭스라고 하고, 제외하면 RACl 매트릭스라고 합니다.

responsive control**반응형 제어**

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in Implementing security. controls on AwS.

부작용 또는 보안 기준과의 편차를 수정하도록 설계된 보안 제어입니다. 자세한 내용은 보안 구현의 반응형 제어를 참조하세요. AwS에 대한 제어.

retain

유지

See 7 Rs.

7 Rs를 참조하십시오.

retire

은퇴

See 7 Rs.

7 Rs를 참조하십시오.

Retrieval Augmented Generation (RAG)

검색 증강 생성(RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

LLM이 응답을 생성하기 전에 학습 데이터 원본 외부에 있는 신뢰할 수 있는 데이터 원본을 참조하는 생성형 AI 기술입니다. 예를 들어 RAG 모델은 조직의 기술 자료 또는 사용자 지정 데이터에 대한 의미 체계 검색을 수행할 수 있습니다. 자세한 내용은 RAG란?

rotation

회전

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

공격자가 자격 증명에 액세스하기 어렵게 만들기 위해 비밀을 주기적으로 업데이트하는 프로세스입니다.

row and column access control (RCAC)

행 및 열 액세스 제어(RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

액세스 규칙을 정의한 기본 유연한 SQL 표현식의 사용. RCAC는 행 권한과 컬럼 마스크로 구성됩니다.

RPO

RPO

See recovery point objective. RTO

복구 지점 목표를 참조하세요. RTO

See recovery time objective. runbook

복구 시간 목표를 참조하세요. 런북

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

특정 작업을 수행하는 데 필요한 일련의 수동 또는 자동 절차입니다. 이는 일반적으로 반복적인 작업이나 오류율이 높은 절차를 간소화하기 위해 구축되었습니다.

SAML 2.0

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

많은 자격 증명 공급자(IdP)가 사용하는 개방형 표준입니다. 이 기능을 사용하면 페더레이션 싱글 사인온(SSO)이 활성화되므로 조직 내 모든 사용자에 대해 IAM에서 사용자를 생성하지 않고도 사용자가 AWS Management Console에 로그인하거나 AWS API 작업을 호출할 수 있습니다. SAML 2.0 기반 페더레이션에 대한 자세한 내용은 IAM 설명서의 SAML 2.0 기반 페더레이션 정보를 참조하세요.

SCADA

스카다

See supervisory control and data acquisition. SCP

감시 제어 및 데이터 수집을 참조하십시오. SCP

See service control policy. secret

서비스 제어 정책을 참조하십시오. 비밀

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata. The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

AWS Secrets Manager에서 암호화된 형식으로 저장하는 암호 또는 사용자 자격 증명과 같은 기밀 또는 제한된 정보입니다. 보안 암호 값과 해당 메타데이터로 구성됩니다. 보안 암호 값은 바이너리, 단일 문자열 또는 여러 문자열일 수 있습니다. 자세한 내용은 Secrets Manager 설명서의 Secrets Manager 보안 암호에 포함된 내용? 을 참조하세요.

security by design

설계에 의한 보안

A system engineering approach that takes security into account through the whole development process.

전체 개발 프로세스에서 보안을 고려하는 시스템 엔지니어링 접근 방식입니다.

security control

보안 제어

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

위협 행위자가 보안 취약성을 악용하는 능력을 방지, 탐지 또는 감소시키는 기술 또는 관리 가드레일입니다. 보안 제어에는 예방, 탐지, 대응 및 사전 예방의 네 가지 기본 유형이 있습니다.

security hardening

보안 강화

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating

unnecessary features in configuration files.

공격에 대한 저항력을 높이기 위해 공격 표면을 줄이는 프로세스입니다. 여기에는 더 이상 필요하지 않은 리소스 제거, 최소 권한 부여의 보안 모범 사례 구현 또는 구성 파일에서 불필요한 기능 비활성화와 같은 작업이 포함될 수 있습니다.

security information and event management (SIEM) system

보안 정보 및 이벤트 관리(SIEM) 시스템

Tools and services that combine security information management (SiM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

보안 정보 관리(SiM)와 보안 이벤트를 결합한 도구 및 서비스입니다. 관리(SEM) 시스템. SIEM 시스템은 서버에서 데이터를 수집, 모니터링 및 분석합니다. 네트워크, 장치 및 기타 소스를 사용하여 위협 및 보안 위반을 감지하고 경고를 생성합니다.

security response automation

보안 대응 자동화

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

보안 이벤트에 자동으로 응답하거나 수정하도록 설계된 사전 정의되고 프로그래밍된 작업입니다. 이러한 자동화는 AWS 보안 모범 사례를 구현하는 데 도움이 되는 탐지 또는 반응형 보안 제어 역할을 합니다. 자동화된 응답 작업의 예로는 VPC 보안 그룹 수정, Amazon EC2 인스턴스 패치 적용 또는 자격 증명 교체가 있습니다.

server-side encryption

서버 측 암호화

Encryption of data at its destination, by the AWS service that receives it. service control policy (SCP)

대상에서 데이터를 수신하는 AWS 서비스에 의한 데이터 암호화. 서비스 제어 정책(SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

AWS Organizations의 조직에 있는 모든 계정에 대한 권한을 중앙 집중식으로 제어하는 정책입니다. SCP는 관리자가 사용자 또는 역할에 위임할 수 있는 작업에 대한 가드레일을 정의하거나 제한을 설정합니다. SCP를 허용 목록 또는 거부 목록으로 사용하여 허용 또는 금지되는 서비스 또는 작업을 지정할 수 있습니다. 자세한 내용은 서비스 제어 정책을 참조하세요. AWS Organizations 설명서.

service endpoint

서비스 엔드포인트

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in AWS General Reference.

AWS 서비스에 대한 진입점의 URL입니다. 엔드포인트를 사용하여 프로그래밍 방식으로 대상 서비스에 연결할 수 있습니다. 자세한 내용은 AWS 일반 참조의 AWS 서비스 엔드포인트를 참조하세요.

service-level agreement (SLA)

서비스 수준 계약(SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

IT 팀이 서비스 가동 시간 및 성능과 같이 고객에게 제공하겠다고 약속하는 내용을 명확히 하는 계약입니다.

service-level indicator (SLI)

서비스 수준 표시기(SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

오류율, 가용성 또는 처리량과 같은 서비스의 성능 측면을 측정합니다.

service-level objective (SLO)

서비스 수준 목표(SLO)

A target metric that represents the health of a service, as measured by a service-level indicator. shared responsibility model

서비스 수준 지표로 측정된 서비스의 상태를 나타내는 대상 메트릭입니다. 공동 책임 모델

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security of the cloud, whereas you are responsible for security in the cloud. For more information, see Shared responsibility model.

클라우드 보안 및 규정 준수에 대해 AWS와 공유하는 책임을 설명하는 모델입니다. AWS는 클라우드의 보안을 담당하는 반면, 클라우드의 보안은 사용자가 담당합니다. 자세한 내용은 공동 책임 모델 섹션을 참조하세요.

SIEM

시엠

See security information and event management system. single point of failure (SPOF)

보안 정보 및 이벤트 관리 시스템을 참조하십시오. 단일 실패 지점(SPOF)

A failure in a single, critical component of an application that can disrupt the system. SLA

시스템을 중단할 수 있는 애플리케이션의 중요한 단일 구성 요소의 오류입니다. 슬로가

See service-level agreement.

서비스 수준 계약을 참조하십시오.

SLI

슬리

See service-level indicator. SLO

서비스 수준 표시기를 참조하십시오. SLO

See service-level objective. split-and-seed model

서비스 수준 목표를 참조하십시오. 분할 및 시드 모델

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the Aws Cloud.

현대화 프로젝트를 확장하고 가속화하기 위한 패턴입니다. 새로운 기능과 제품 릴리스가 정의되면 핵심 팀이 분할되어 새로운 제품 팀을 만듭니다. 이를 통해 조직의 기능과 서비스를 확장하고 개발자 생산성을 개선하며 신속한 혁신을 지원합니다. 자세한 내용은 AWS 클라우드에서 애플리케이션을 현대화하기 위한 단계적 접근 방식을 참조하세요.

SPOF

스포프

See single point of failure. star schema

단일 실패 지점을 참조하십시오. 스타 스키마

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes..

하나의 큰 팩트 테이블을 사용하여 트랜잭션 또는 저장하는 데이터베이스 조직 구조입니다. 측정된 데이터를 사용하고 하나 이상의 작은 차원 테이블을 사용하여 데이터 속성을 저장합니다. 이 구조는 데이터 웨어하우스 또는 비즈니스 인텔리전스 목적으로 사용하도록 설계되었습니다.

strangler fig pattern

교살자 무화과 패턴

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

점진적으로 다시 작성하고 교체하여 모놀리식 시스템을 현대화하는 접근 방식입니다. 레거시 시스템을 서비스 해제할 수 있을 때까지의 시스템 기능입니다. 이 패턴은 확립된 트리로 자라서 결국 숙주를 극복하고 대체하는 무화과 덩굴의 비유를 사용합니다. 이 패턴은 다시 작성할 때 위험을 관리하는 방법으로 Martin Fowler에 의해 도입되었습니다. 모놀리식 시스템. 이 패턴을 적용하는 방법에 대한 예제는 레거시 현대화를 참조하세요. 컨테이너 및 Amazon API를 사용하여 충분 방식으로 ASMX(Microsoft ASP.NET) 웹 서비스를 제공합니다. 게이트웨이.

subnet

서브넷

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone. supervisory control and data acquisition (SCADA)

VPC의 IP 주소 범위입니다. 서브넷은 단일 가용 영역에 있어야 합니다. 감독 제어 및 데이터 수집(SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

제조업에서 하드웨어와 소프트웨어를 사용하여 물리적 자산과 생산 운영을 모니터링하는 시스템입니다.

symmetric encryption

대칭 암호화

An encryption algorithm that uses the same key to encrypt and decrypt the data. synthetic testing

동일한 키를 사용하여 데이터를 암호화하고 복호화하는 암호화 알고리즘입니다. 합성 테스트

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

잠재적인 문제를 감지하거나 성능을 모니터링하기 위해 사용자 상호 작용을 시뮬레이션하는 방식으로 시스템을 테스트합니다. Amazon CloudWatch Synthetics를 사용하여 이러한 테스트를 생성할 수 있습니다.

system prompt

시스템 프롬프트

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior.

LLM의 동작을 지시하기 위해 LLM에 컨텍스트, 지침 또는 지침을 제공하는 기술입니다.

System prompts help set context and establish rules for interactions with users..

시스템 프롬프트는 컨텍스트를 설정하고 사용자와의 상호 작용에 대한 규칙을 설정하는 데 도움이 됩니다..

T

T

tags

태그

Key-value pairs that act as metadata for organizing your AwS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AwS resources.

AwS 리소스를 구성하기 위한 메타데이터 역할을 하는 키-값 페어입니다. 태그는 리소스를 관리, 식별, 구성, 검색 및 필터링하는 데 도움이 될 수 있습니다. 자세한 내용은 태그 지정을 참조하세요. AwS 리소스.

target variable

대상 변수

The value that you are trying to predict in supervised ML. This is also referred to as an outcome variable. For example, in a manufacturing setting the target variable could be a product defect.

지도 ML에서 예측하려는 값입니다. 이를 결과 변수라고도 합니다. 예를 들어, 제조 설정에서 대상 변수는 제품 결함일 수 있습니다.

task list

작업 목록

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress..

Runbook을 통해 진행률을 추적하는 데 사용되는 도구입니다. 작업 목록에는 Runbook에 대한 개요와 완료해야 할 일반 작업 목록이 포함되어 있습니다. 각 일반 작업에 대해 필요한 예상 시간, 소유자 및 진행률이 포함됩니다.

test environment

테스트 환경

See environment

환경 보기

training

훈련

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data.

ML 모델이 학습할 데이터를 제공합니다. 훈련 데이터에는 정답이 포함되어야 합니다. 학습 알고리즘은 입력 데이터를 매핑하는 훈련 데이터에서 패턴을 찾습니다.

attributes to the target (the answer that you want to predict). It outputs an ML model that.

대상(예측하려는 답변)에 대한 속성입니다. ML 모델을 출력합니다.

captures these patterns. You can then use the ML model to make predictions on new data for.

이러한 패턴을 캡처합니다. 그런 다음 ML 모델을 사용하여 새 데이터에 대한 예측을 수행할 수 있습니다.

which you don't know the target..

대상을 모르는 것..

transit gateway

트랜짓 게이트웨이

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the Aws Transit Gateway documentation.

VPC와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다. 자세한 내용은 AWS Transit Gateway 설명서의 전송 게이트웨이란 무엇입니까?

trunk-based workflow

트렁크 기반 워크플로

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially..

개발자가 기능 분기에서 로컬로 기능을 빌드하고 테스트한 다음 해당 변경 내용을 기본 분기에 병합하는 접근 방식입니다. 그런 다음 기본 분기는 개발, 사전 프로덕션 및 프로덕션 환경에 순차적으로 빌드됩니다..

trusted access

신뢰할 수 있는 액세스

Granting permissions to a service that you specify to perform tasks in your organization in Aws. Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more. information, see Using Aws Organizations with other AwS services in the Aws Organizations documentation.

Aws의 조직에서 작업을 수행하도록 지정한 서비스에 권한을 부여합니다. Organizations 및 사용자를 대신하여 해당 계정에서 신뢰할 수 있는 서비스는 각 계정에서 해당 역할이 필요할 때 관리 작업을 수행하기 위해 서비스 연결 역할을 생성합니다. 자세한 내용은 자세한 내용은 Aws Organizations 설명서의 다른 AwS 서비스와 함께 Aws Organizations 사용을 참조하세요.

tuning

조율

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating. these steps several times under different settings to optimize the model..

ML 모델의 정확도를 개선하기 위해 훈련 프로세스의 측면을 변경합니다. 예를 들어 레이블 세트를 생성하고 레이블을 추가한 다음 반복하여 ML 모델을 훈련할 수 있습니다. 모델을 최적화하기 위해 서로 다른 설정에서 이러한 단계를 여러 번 수행합니다.

two-pizza team

투피자 팀

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development..

두 개의 피자를 먹일 수 있는 소규모 DevOps 팀입니다. 두 개의 피자 팀 규모는 소프트웨어 개발에서 협업을 위한 최상의 기회를 보장합니다..

U**U**

uncertainty

불확실

A concept that refers to imprecise, incomplete, or unknown information that can undermine the.

을 훼손할 수 있는 부정확하거나 불완전하거나 알 수 없는 정보를 가리키는 개념입니다.

reliability of predictive ML models. There are two types of uncertainty: Epistemic uncertainty.

예측 ML 모델의 신뢰성. 불확실성에는 두 가지 유형이 있습니다. 인식론적 불확실성.

is caused by limited, incomplete data, whereas aleatoric uncertainty is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in.

는 제한적이고 불완전한 데이터로 인해 발생하는 반면, 우연적 불확실성은 데이터에 내재된 노이즈와 무작위성으로 인해 발생합니다. 자세한 내용은 불확실성 정량화를 참조하세요.

deep learning systems guide.

딥 러닝 시스템 가이드.

undifferentiated tasks

미차별화 작업

Also known as heavy lifting, work that is necessary to create and operate an application but that doesn't provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning..

무거운 물들기라고도 하며 애플리케이션을 만들고 운영하는 데 필요한 작업입니다. 이는 최종 사용자에게 직접적인 가치를 제공하거나 경쟁 우위를 제공하지 않습니다. 차별화되지 않은 작업의 예로는 조달, 유지 관리 및 용량 계획이 있습니다.

upper environments

상부 환경

See environment

환경 보기

V**V**

vacuuming

진공 청소기

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

스토리지를 회수하고 성능을 향상시키기 위해 증분 업데이트 후 정리를 포함하는 데이터베이스 유지 관리 작업입니다.

version control

버전 관리

Processes and tools that track changes, such as changes to source code in a repository.

리포지토리의 소스 코드 변경과 같은 변경사항을 추적하는 프로세스 및 도구입니다.

VPC peering**VPC 피어링**

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.. vulnerability

프라이빗 IP 주소를 사용하여 트래픽을 라우팅할 수 있는 두 VPC 간의 연결입니다. 자세한 내용은 Amazon VPC 설명서의 VPC 피어링이란 무엇입니까?. 취약성

A software or hardware flaw that compromises the security of the system.

시스템 보안을 손상시키는 소프트웨어 또는 하드웨어 결함입니다.

warm cache**웜 캐시**

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

자주 액세스되는 현재 관련 데이터를 포함하는 버퍼 캐시입니다. 데이터베이스 인스턴스는 기본 메모리 또는 디스크에서 읽는 것보다 빠른 버퍼 캐시에서 읽을 수 있습니다.

warm data**웜 데이터**

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

자주 액세스하지 않는 데이터입니다. 이러한 종류의 데이터를 쿼리할 때 일반적으로 적당히 느린 쿼리가 허용됩니다.

window function**창 함수**

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row..

현재 레코드와 어떤 식으로든 관련된 행 그룹에 대해 계산을 수행하는 SQL 함수입니다. 창 함수는 이동 평균을 계산하거나 현재 행의 상대적 위치를 기반으로 행 값에 액세스하는 것과 같은 작업을 처리하는 데 유용합니다.

workload**작업**

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

고객 대면 애플리케이션 또는 백 엔드 프로세스와 같은 비즈니스 가치를 제공하는 리소스 및 코드의 모음입니다.

workstream**작업 스트림**

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

특정 작업 집합을 담당하는 마이그레이션 프로젝트의 기능 그룹입니다. 각 작업 스트림은 독립적이지만 프로젝트의 다른 작업 스트림을 지원합니다. 예를 들어 포트폴리오 작업 스트림은 애플리케이션의 우선 순위 지정, 웨이브 계획 및 마이그레이션 메타데이터 수집을 담당합니다. 포트폴리오 작업 스트림은 이러한 자산을 마이그레이션 작업 스트림에 전달한 다음 서버와 애플리케이션을 마이그레이션합니다.

WORM

벌레

See write once, read many. WQF

한 번 쓰고 많이 읽습니다. WQF

See Aws Workload Qualification Framework write once, read many (WORM)

AWS Workload Qualification Framework 한 번 쓰기, 여러 번 읽기(WORM) 단원을 참조하십시오.

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

데이터를 한 번 쓰고 데이터가 삭제되거나 수정되지 않도록 하는 스토리지 모델입니다. 권한이 부여된 사용자는 필요한 만큼 데이터를 읽을 수 있지만 변경할 수는 없습니다. 이 데이터 스토리지 인프라는 변경할 수 없는 것으로 간주됩니다.

zero-day exploit

제로데이 악스플로잇

An attack, typically malware, that takes advantage of a zero-day vulnerability. zero-day vulnerability

제로데이 취약점을 이용하는 공격(일반적으로 맬웨어)입니다. 제로데이 취약점

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

프로덕션 시스템의 완화되지 않은 결함 또는 취약성입니다. 위협 행위자는 이러한 유형의 취약성을 사용하여 시스템을 공격할 수 있습니다. 개발자는 공격의 결과로 취약성을 인식하는 경우가 많습니다.

zero-shot prompting

제로샷 프롬프트

Providing an LLM with instructions for performing a task but no examples (shots) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

LLM에 작업 수행에 대한 지침을 제공하지만 이를 안내하는 데 도움이 될 수 있는 예제(샷)는 제공하지 않습니다. LLM은 작업을 처리하기 위해 사전 학습된 지식을 사용해야 합니다. 제로샷 프롬프트의 효과는 작업의 복잡성과 프롬프트의 품질에 따라 달라집니다. 소수 샷 프롬프트도 참조하세요.

zombie application

좀비 응용 프로그램

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.

평균 CPU 및 메모리 사용량이 5% 미만인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하는 것이 일반적입니다.