Building data spaces for sustainability use cases

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Building data spaces for sustainability use cases

Table of Contents

Building data spaces for sustainability use cases

Malte Gasseling and Ramy Hcini (Think-it)

January 2024 (document history)

The primary objective of this strategy is to provide you with a clear starting point for how to design, operate, and maintain data spaces. The document explains the benefits and potential of data spaces, particularly in the context of environmental, social, and corporate governance (ESG) data exchange initiatives. It showcases the building blocks and provides information about how to join a data space. It also provides examples of options for building data spaces on the Amazon Web Services (AWS) Cloud. This strategy document is substantiated by a technical pattern that combines concrete modules and materials with step-by-step technical guidance to make the strategy a reality.

Exchanging data through federated technology for environmental impact and beyond

Data spaces are federated networks for trusted data exchange, with control over one's data as a core principle. They enable organizations to share, exchange, and collaborate on data at scale by offering a cost-effective and technology-agnostic solution.

Data spaces have the potential to significantly drive efforts for a sustainable future by supporting empirical problem solving with an end-to-end approach that involves all relevant stakeholders. This can stimulate new ideas and the discovery of new opportunities through collaborative, data-driven innovation and help build the data value chain.

By breaking down data barriers and enabling diverse sources of data to be exchanged, your organization can tap into the combined knowledge of its peers, leading to new solutions and breakthroughs. Consequently, data spaces contribute to sustainability initiatives by enabling the sharing of ESG data at a large scale, promoting collaborative initiatives and industry standards. This is particularly pertinent in the context of evolving supply-chain due diligence and compliance requirements, including regulations such as the Non-Financial Reporting Directive (NFRD), the Corporate Sustainability Reporting Directive (CSRD), and similar initiatives.

In addition, data spaces help you to make informed decisions that support sustainable development and reduce environmental impact. By creating trusted and accessible exchange

networks for ESG data, data spaces can help your organization better track its progress towards sustainability targets, identify areas for improvement with a participatory perspective, and demonstrate compliance with regulatory requirements more efficiently.

In the context of this guide for decision makers and business executives, data spaces are one of the technologies to support the implementation of the recent political agreement reached by the European Parliament and the Council of the European Union on the European Data Act. The European Data Act seeks to unlock industrial data, improve data accessibility, and foster a competitive European cloud market, ultimately promoting data-driven solutions and collaboration, in line with the broader data strategy for Europe. This aligns with the principles of data spaces in facilitating data exchange and collaboration for sustainable development, because both initiatives aim to empower organizations through data-driven solutions.

To learn more about the benefits of cloud technology for data spaces and the role of AWS, see the blog post Enabling data sharing through data spaces and AWS.

Creating positive environmental impact through data spaces

Organizations that participate in data spaces, by design, own and control their involvement and collaboration within such networks. This might act as a barrier of entry, but it's also considered as a potential opportunity for your organization to learn how to better control its data and increase the value captured from data assets.

Observed benefits for organizations building new or joining existing data spaces include the following:

- **Improved data quality and integrity** – Using standardized data formats, validating data sources, and implementing data validation rules
- **Increased efficiency** – Automating data exchange processes, reducing manual errors, and streamlining workflows
- **Enhanced collaboration** – Facilitating cross-organizational collaboration, accelerating innovation, and creating new business opportunities

Data spaces as support for ESG reporting

Organizations and cities use data spaces to empower informed decisions that support sustainable development and reduce environmental impact. Sustainability goals are omnipresent in almost all industries. The following examples highlight how data space initiatives can drive ESG goals and targets:

- **Smart cities** – Data spaces can help optimize energy consumption, traffic management, waste management, and urban infrastructure, leading to reduced environmental footprints and improved quality of life for citizens. Initiatives such as City Dataspace and Smart Parking promote sustainability by reducing traffic congestion and promoting efficient use of resources. For more information, see the International Data Spaces: Data Spaces Radar page.
- **Healthcare and public health** – Data exchanged through data spaces can help to improve disease surveillance, pandemic preparedness, and resource allocation. These improvements lead to more efficient and sustainable healthcare systems.
- **Renewable energy optimization** – Data-driven technologies can optimize the generation, distribution, and consumption of renewable energy sources, such as solar and wind, to increase

their efficiency and integration into the energy grid. Initiatives such as Data spAces for smaRt Energy (DARE) and Post-Platforms for Renewable Energy aim to reduce energy consumption, minimize waste, and promote sustainable economic growth. For more information about the Post-Platforms for Renewable Energy initiative, see the International Data Spaces: Data Spaces Radar page.

Examples of data spaces built on top of AWS services

AWS has played a pivotal role in shaping the surrounding landscapes of data spaces and collaborative ecosystems across various industries. By providing robust and scalable cloud-native services, AWS has empowered organizations to create and manage data spaces that facilitate data sharing, collaboration, and innovation.

This section introduces two examples of ongoing data spaces built on AWS infrastructure, showcasing how the technology can be harnessed to foster data-driven initiatives, streamline information exchange, and drive advancements in diverse sectors. These real-world examples illustrate the versatility and potential of AWS in catalyzing the development of data spaces and collaborative networks.

SFC Exchange Network for the logistics industry

The Smart Freight Centre (SFC) Exchange Network is a collaborative network focused on creating a data space in the logistics sector with the primary goal of promoting transparency and decarbonization in transport chains by facilitating activity and logistics emission data exchange and reporting. The project involves various stakeholders, including logistics service providers, shippers, carriers, and tool providers, who collaborate under a shared governance framework that emphasizes data sovereignty and security.

To achieve the SFC Exchange Network targets, a roadmap of several top use cases based on the input and needs of its participants has been drafted. The initial use case is the "Corporate Target Monitoring & Reporting." This use case focuses on assessing the percentage of participating companies that accurately report their carbon emissions, thus ensuring transparency and accountability in carbon reduction efforts.

Catena-X for the automotive industry

Catena-X is one of the most advanced data spaces to date, driven by the automotive industry to address challenges and opportunities in traceability, sustainability, circular economy, and efficient supply chains. The data space has shown an immense commitment to sustainability, specifically in measuring and reducing carbon emissions within the automotive industry supply chain, and in its efforts to standardize and improve carbon data management.

Catena-X has committed to reducing carbon emissions throughout the product life cycle. To achieve this target, the association has identified the need for standardized measurements

along the value chain, accurate documentation of real carbon data, and comparability within the automotive industry. One of the initiatives focuses on developing a Product Carbon Footprint Rulebook, which provides a uniform methodology for recording and comparing carbon data.

The association has collaborated with stakeholders from technology, industry, and associations, including the World Business Council for Sustainable Development (WBCSD), to develop these standards and procedures. One key objective for the success of Catena-X is to include the entire supply chain, especially small and medium-sized enterprises (SMEs) in the data exchange and therefore the success of their initiative.

Building data spaces

As explained on the AWS Blog, a data space at its core "helps overcome the problem of inter-organizational data integration across heterogeneous technology stacks, environments, and geographies." The technology enables organizations to retain control over their data while facilitating innovation, collaboration and sharing insights with others.

Data spaces provide a distributed alternative to traditional centralized data management systems such as data lakes and data lake houses, which often rely on a single point of trust. This makes data spaces more resilient and robust than traditional systems. It also encourages collaboration and shared responsibility, which builds trust among stakeholders because they are following open standards and compatible rules for data exchange. The balance between control and cooperation keeps sensitive data safe and encourages innovation.

Core roles in a data space

Building a data space involves the following three core roles:

- **Data space authority** – As defined by the International Data Spaces Association, the data space authority manages one or several data spaces that include participant registration and might entail mandating business or technical requirements. For example, a Data Space Authority might require participants to obtain some form of business certification. A Data Space authority may also impose technical requirements such as support for the technical enforcement of specific usage policies.
- **Data providers** – The provider manages the data assets to be shared. The provider helps to ensure data asset quality and determines usage policies.
- **Data consumers** – The consumers typically interact with the provider to obtain the data they need. The consumers might use the data for analysis, decision-making, research, or other applications.

The provider makes data available in a structured and accessible manner, while the consumer accesses and utilizes the data according to the agreed upon contract. As data spaces grow and mature, additional roles and responsibilities can be introduced. For example, the following roles are common:

- **Application providers** – Entities responsible for developing and offering software applications that using the data within a data space.
- **Orienting partners** – Entities that facilitate the integration of new data sources, data producers, or data consumers into a data space. They play a crucial role in expanding and enriching the data space ecosystem.
- **Trusted technical partners** – Entities that act as an intermediaries or facilitators in technical matters related to data sharing and collaboration within a data space. They cover a wide range of responsibilities, including the following:
  - Data governance
  - Data quality
  - Security
  - Facilitating data integration and compatibility
  - Technical support and troubleshooting
  - Monitoring data space health
  - Compliance with regulations

How data spaces are typically structured and managed

Both the relationships among participants and their data readiness define the ground rules of governance and trust in a data space. To establish trust between the participants, data space authorities can adopt one of three typical patterns:

- **Centralized data space authority** – The data space authority creates participation rules and manages the registry of the data space participants. Core data space services are managed and accessed through this central entity, which facilitates data sharing and helps ensure consistent governance. This approach offers simplicity and uniformity, but it might raise concerns about data control and potential single points of failure or trust.
- **Federated data space authority** – In the federated (or distributed) model, the data space authority retains some degree of centralized control but improves on the technical and security challenges. Multiple entities share responsibility for providing the core services, instead of just one entity. Federation promotes autonomy, scalability, and flexibility while helping ensure control over data and addressing privacy concerns.
- **Decentralized data space authority** – A fully decentralized authority eliminates the need for a central point of trust, and governance is distributed among participating organizations.

Decentralization promotes autonomy, privacy, and resilience, but it might introduce challenges related to coordination, consensus, and governance.

Key steps in building a data space

The data space authority leads and drives building the data space by owning or delegating several key steps that cover business, legal, operational, functional, and technical considerations.

The Data Space Support Centre (DSSC) provides a starter kit that includes a set of foundational questions to answer within each dimension. Starter kit questions are included in the following considerations:

1. **Define the scope and purpose of the data space** – Determine what types of data will be included in the data space, who will use it, and what business needs it will fulfill. The data types and use cases might evolve over time as the data space adoption increases.
2. **Identify initial participants, source systems, and datasets** – Determine the initial requirements and expectations from involved stakeholders. Identify the first set of data sources that will be exchanged in the data space, and determine which datasets are most relevant for the intended use cases.
3. **Establish governance principles and processes** – Define roles and responsibilities for data management and usage. Establish data standards, data exchange policies, and security protocols. Provide incentives for an environment of collaboration.
4. **Test and validate the data space use cases** – Test the data space to ensure that it meets the requirements of the intended use case, and validate that key performance indicator (KPI) targets are achieved.
5. **Deploy and operate the data space technical infrastructure** – Deploy the data space in a production environment, and monitor its services' performance and usage to identify areas for improvement. For more information, see the technical pattern.
6. **Continuously improve the data space** – Refine the ecosystem over time based on feedback from users and stakeholders by updating the policies and improving both the developers' and the participants' ecosystems.
7. **Scale up** – Expand the data space with more participants, more and higher-quality data, integrated data analytics, and other services. For a successful scale-up, it's important to ensure close cooperation between IT and business.

A financially sound business model is vital to ensure the success and growth of data spaces. Revenue optimization and business model design are, however, not part of the scope of this document. This strategy focuses on providing a blueprint for cost-efficient architectures based on and powered by AWS services.

Core technical components of a data space

When you build a data space, the following components are essential:

- **Trust framework** – A set of guidelines, standards, and principles that define the trust and security measures within a data space. The trust framework outlines the rules, policies, and best practices for ensuring the secure exchange of data among participants.
- **Dataspace Protocol** – A set of rules and specifications that dictate how data is transmitted, exchanged, and accessed within a data space. The Dataspace Protocol outlines the technical standards and methods for data sharing, maintaining control over data, interoperability, and efficient communication between participants.
- **Identity hub** – The central management of the identity of the participants and authentication methods.
- **Discovery service** – A way to search for data and share it with others.
- **Data space connectors** – An implementation of connectors that provides and manages the data space policies, also referred to as the data exchange rules.

Trust frameworks

A trust framework defines the trust and security approaches and measures within a data space. Trust frameworks are the foundational layer on which data spaces can be built. Two commonly used frameworks have contributed to the implementation and adoption of data spaces.

International Data Space Association and the IDS Trust Framework

The International Data Space Association (IDSA) is a nonprofit organization based in Germany that was founded in 2016. Its aim is to provide a secure, privacy-preserving, and trustworthy scheme for data exchange, known as the International Data Space (IDS).

The IDS Trust Framework provides a solution for data exchange between organizations and individuals, enabling secure and efficient data sharing, processing and use. The framework includes

a reference architecture, open source building blocks, and a certification process for creating and operating data spaces. IDSA works to promote the use of the IDS trust framework and to establish it as a global standard for data exchange and data sovereignty.

Gaia-X Trust Framework

The Gaia-X Trust Framework represents a significant advancement in data management by addressing challenges that traditional technologies struggled with. It excels in two critical aspects: data sovereignty and interoperability. The Gaia-X Trust Framework helps ensure that organizations retain control over their data even when sharing it, which establishes a robust framework for data security and privacy. This level of control is akin to a secure digital vault for sensitive information.

Furthermore, The Gaia-X Trust Framework excels in interoperability governance, integrating diverse computer systems and enabling them to communicate effectively. It facilitates an environment where various digital components work together harmoniously. This innovative approach enhances data sharing while reducing costs, making it accessible to a broader range of organizations. Unlike older technologies that could limit flexibility, the Gaia-X Trust Framework offers greater freedom of choice, fostering a modern and open ecosystem for data management.

The Dataspace Protocol

The Dataspace Protocol is a set of rules and standards that define how data is shared and consumed within a data space. Its development is driven and supported by the International Data Spaces Association (IDSA) to provide a common language and structure for data exchange across different domains and industries.

The Dataspace Protocol defines key concepts and components that act as a basis for standardization and interoperability of data exchange:

- **Data representation and cataloging** – Definition of the structure and format of the data being shared.
- **Data assets** – Individual pieces of data published to a data space. Assets can be versioned, and their metadata can include information such as timestamps, authors, and descriptions.
- **Data services** – Functionality provided by a data space to perform operations on assets, such as querying, filtering, or transforming data. Services can be invoked using REST APIs or message queues.
- **Exchange policies** – Rules governing how data can be accessed, modified, or deleted. Data usage and data control policies can be defined at multiple levels, including organizational, dataset, or

asset level. The policies are attached to each asset through a connector. Policy violations can initiate alerts and actions to enforce data governance.

Connector technologies for data spaces

Connectors are software tools that enable data to be shared and integrated between various systems, applications, and data sources. In the context of data spaces, connectors play a key role in communication and data exchange across different platforms, systems, and organizations that comply with the predefined standards and exchange policies of the Dataspace Protocol.

Eclipse Dataspace Components based connectors

The Eclipse Dataspace Components (EDC) framework is developed by the Eclipse Foundation as free and open source software. The goal of the EDC framework is to create an efficient and functional data transfer component that implements the protocols of the IDS standard and pursues compatibility with the requirements of the Gaia-X project.

As a central component, the connector enables the exchange of data through defined data sovereignty contracts that are automatically negotiated to govern access to data assets. With a focus on extensibility and adaptability, the architecture of the EDC was developed based on feedback from the IDS and Gaia-X initiatives.

The EDC framework is designed and built upon the following four pillars:

- **Identity** – Each participant remains in control of their identity.
- **Trust** – Each participant decides who to trust.
- **Sovereignty** – Each participant decides under what policies their data is shared.
- **Interoperability** – Each participant remains in control of their deployment.

FIWARE TRUE Connector

FIWARE TRUE Connector provides a specification that your organization can use to share data securely and efficiently within the International Data Spaces (IDS) ecosystem. It provides a standardized way of exchanging data securely and in a traceable manner. The tool consists of three main components:

- Execution Core Container
- FIWARE Data Application

- Usage-Control Data Application

These components work together to enable data exchange, communication with identity providers, and enforcement of usage control policies. By using FIWARE TRUE Connector, your organization can participate in the IDS ecosystem and benefit from secure, efficient, and interoperable data sharing.

Simpl

Simpl is a smart middleware platform that represents a significant step toward creating common European data spaces. It's designed to address the challenge of resource sharing while preserving control and security, fostering trust among stakeholders. Its role in promoting interoperability and resource sharing while ensuring control and security makes it a promising solution for public and private-sector entities. Collaboration is essential, and Simpl acts as a common glue, ensuring interoperability across diverse capacities without costly interfaces.

As the ecosystem continues to evolve, Simpl is positioned to adapt and become a vital connector for European data spaces. However, considerations about its decentralized identity system and the need for further integration remain important points to address. The potential for Simpl to be recommended or mandated by the European Commission highlights the ongoing importance of this project in the European data landscape.

Minimum viable data space as a starting point

A minimum viable data space (MVDS) is a basic version of a data space that contains only enough components to fulfill a specific business need. It typically includes a small number of participants with datasets that are essential for a particular use case or for proof of value. It usually includes only minimal metadata and governance structures.

The purpose of an MVDS is to provide a starting point for data sharing and collaboration, which can then be expanded and refined over time. Typically, an MVDS will include a number of centralized components to accelerate the adoption and exchange of data by participants.

MVDS workflow example

An example of an MVDS might have the following:

- A provider

- A consumer
- A certificate authority
- A centralized identity service

The certificate authority issues digital certificates that serve as cryptographic credentials for participants. These certificates are used by the identity service to verify the identity of the entities involved in data exchange.

The identity service is responsible for managing dynamic attributes related to the participants in the data space. These attributes might include information such as access permissions, roles, and other metadata associated with the participants.

The data exchange uses the following basic workflow:

1. The certificate authority issues certificates to the consumer connector and the provider connector.
2. When the consumer requests data from the provider, the centralized identity service provides data access tokens (DATs) to the consumer and the provider.
3. The provider sends data to the consumer on request.

To deploy and run such an MVDS on AWS, you can use containers within Amazon Elastic Kubernetes Service (Amazon EKS) and other managed services such as Amazon Relational Database Service (Amazon RDS) for databases and AWS Secrets Manager for secrets management.

Operating and maintaining data spaces

The data space authority owns the operation and maintenance tasks. Usually, it delegates those tasks to trusted technical partners. The tasks can include but are not limited to the following:

- **Prioritize standardization, performance, and scalability** – Ensure that standardization is upheld to enable smooth data exchange and collaboration. Decision-makers should commit to adopting common data formats, naming conventions, and protocols.
- **Emphasize user-friendly design and accessibility** – It's crucial to create interfaces and processes that are user-friendly and accessible to both existing and new participants. Provide clear documentation, training resources, and support services to facilitate rapid adoption and ensure that participants can effectively utilize the data space.
- **Establish key success criteria and regularly assess them as performance benchmarks** – Evaluate metrics related to system usage, data compliance, efficiency, user satisfaction, and orientation times. Actively seek positive feedback and participant satisfaction as indicators of success, making continuous improvements based on this input.
- **Establish scaling and failover mechanisms** – This is fundamental in ensuring the uninterrupted functionality and dependable performance of data spaces, particularly in the face of evolving requirements and unexpected challenges.
- **Closely examine milestones and the roadmap proposed for the data space's stable release** – These timelines and objectives should align with the organization's strategic goals and commitments, ensuring that the data space development is on the right track.
- **Align with participant goals** – Ensure that the design and implementation of the data space align with the participants' broader strategic goals. This particularly applies in areas such as sustainability, efficiency, and data-driven decision-making.
- **Monitor continuously system performance, user satisfaction, and compliance with standards** – Be prepared to make necessary adjustments based on feedback and evolving requirements.
- **Evaluate the cost implications** – Track projected costs of the proposed roadmap and the technical or development work to be done. Strive to strike a balance between the investment in data space development and the expected benefits and returns.
- **Consider potential risks and develop mitigation strategies** – This especially concerns technical challenges, scalability issues, and participant orientation difficulties. Take proactive measures to address these risks and ensure the data space's long-term success.

- **Ensure ongoing support and maintenance** – After the initial deployment, have processes and mechanisms in place to keep the data space healthy and up to date.

Joining data spaces

Joining an existing data space presents a compelling opportunity for organizations to become part of a well-established and collaborative ecosystem. By joining a data space instead of building one from scratch, you can use the infrastructure, data resources, and participant network already in place.

Prepare to join a data space

An initial stage of orientation to a data space is focused on learning about the core mission, objectives, and advantages of the data space. This essential orientation process may take various forms, such as attending webinars, reviewing comprehensive documentation, or attending hands-on orientation sessions.

The preparation phase serves as a critical foundation. You want to have a clear understanding that the data space's purpose and support for effective collaboration and data sharing align with your organization's goals. Research and consider the following:

- **The data space landscape and core mission** – Types of data spaces, their focus areas, and the communities they serve
- **Organizational readiness to effectively join and contribute within a data space** – Your organization's data maturity level and the scope of participation
- **The business case for participation** – The benefits of joining a data space, such as improved data quality, increased efficiency, and enhanced collaboration, with defined KPIs and success criteria
- **Roles and responsibilities** – Clear data ownership, access controls, and dispute-resolution mechanisms

To help prepare, use the Checklist for Data Space Readiness provided by Think-it.

Join and participate in a data space

A successful preparation stage helps participants to integrate with the data space, exchange data securely, and collaboratively explore the potential of shared information for their specific use cases.

The orientation process varies in detail and complexity depending on the specific data space and its objectives. Orientation will likely include the following common steps and considerations.

Membership and agreements

- Depending on the data space, your organization might need to submit a membership application.
- Review and sign legal agreements outlining the terms, data governance, security, and responsibilities for data sharing.

Technical integration and high availability

- Select the appropriate technology for the control planes, such as Amazon EKS, and data planes, such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, AWS Glue, and Amazon Kinesis.
- Integrate your organization's systems with the data space's connector technology and data services.
- Set up adequate Service Level Agreements (SLAs) and establish effective processes to ensure the reliability and availability of federated services and data provider endpoints.
- Determine whether data standardization and transformation are necessary to ensure compatibility with the data space's standards.
- Perform data quality and compliance checks.
- Conduct rigorous testing to verify that data can flow securely and without interruption.

Data sharing, collaboration, and innovation

- Your organization begins to share relevant data into the data space. Data is validated, and quality control measures are applied to maintain data integrity.
- Your organization gains access to the data contributed by others, aligning data with your specific use cases. Usage is monitored to ensure compliance with data governance and security policies.
- You are encouraged to explore innovative use cases and use shared data for mutual benefits.
- Networking and collaboration opportunities can lead to partnerships and value-added services.

Compliance and governance

- Regular compliance checks and audits help ensure adherence to data governance standards.
- Governance frameworks for rule enforcement, policies, and data exchange standards are followed as they evolve.

Scaling and growth

- Data standards, security protocols, and governance policies are adhered to as they are adapted to meet changing needs and challenges.
- As trust and participation increase, the data space might expand its ecosystem, including more participants and data sources.
- As the data space ecosystem grows, your organization must strengthen its capacity to use data in a sovereign way to achieve goals and build a data-oriented culture and business practices. This requires training and upskilling.

Challenges and limitations

Depending on multiple factors, there are several challenges and limitations to consider when designing and joining data spaces, including the following 10 most observed:

- **Technical complexity** – Setting up and maintaining a data space requires some technical expertise, especially in areas such as data integration, data governance, and cybersecurity. Organizations that lack skilled professionals to manage these tasks might struggle to get the full benefit from building a data space.
- **Data quality issues** – Data spaces rely on high-quality data to function effectively. However, data quality remains a significant challenge, especially when dealing with legacy systems, disparate data sources, and human error. Ensuring data accuracy, completeness, and consistency across all datasets is crucial but often difficult to achieve.
- **Integration challenges** – Combining data from multiple sources into a single, unified view can be a complex task. Different data formats, schemas, and semantics can create integration challenges that require significant time and resources to resolve.
- **Data privacy and security concerns** – Data spaces must ensure the privacy and security of sensitive information, especially in industries, such as healthcare or finance, that are subject to strict regulations. Implementing robust security measures and maintaining data confidentiality are essential but not always straightforward.
- **Cultural and adoption barriers** – Encouraging collaboration and data sharing across different departments or organizations can be challenging. Some teams or organizations might be hesitant to share their data, citing concerns about intellectual property, competition, or past negative experiences.
- **Scalability limitations** – As data volumes continue to grow, data spaces must scale to accommodate the increase. However, scaling can introduce new challenges, such as managing larger amounts of data, ensuring performance, and maintaining data quality. Those limitations can occur on a governance level as well as on a participant level.
- **Cost and ROI** – Implementing and maintaining a data space does incur some costs, including infrastructure, personnel, and software expenses. Be sure to project and demonstrate a clear return on investment (ROI) for building a data space, especially in the early stages of implementation.
- **Lack of standardization** – The lack of standardization in data formats, schemas, and ontologies can make it difficult for different systems to communicate and share data effectively. Establishing common standards and frameworks can help address these challenges.

- **Change management** – Designing or joining a data space requires significant changes to existing workflows, processes, and culture. Managing this change can be challenging, especially in organizations with entrenched habits or resistance to new technologies.
- **Ethical considerations** – With the increasing emphasis on data-driven decision-making as well as innovative business models based on data, concern is growing about bias. This includes bias in the data exchanged and in the services offered within data spaces. Ensuring fairness, accountability, and transparency in data spaces is critical, but it requires careful consideration and effort.

By acknowledging and addressing these challenges and limitations, your organization can better understand the potential hurdles when building or joining data spaces and develop strategies to overcome them.

Conclusion

This strategy document explored the dynamic landscape of data spaces and their transformative potential as federated networks for trusted data exchange. Data spaces are not only technological solutions. They are also catalysts for positive environmental impact and sustainable development. They play an important role in breaking down barriers, fostering collaboration, and promoting the large-scale sharing of ESG data. The examples of SFC Data Exchange Network and Catena-X illustrated the adaptability of data spaces across industries, underlining data space versatility.

The exploration of the different aspects of building and operating data spaces, coupled with insights into trust frameworks, connector technologies, and the minimum viable data space (MVDS) concept, provides a practical guide for decision-makers. Yet, it is crucial to underscore the necessity of thoughtful planning for data use post-exchange. This entails envisioning how the shared data will be used for decision-making, innovation, and value creation.

A comprehensive data strategy must encompass considerations for data governance, analytics, and integration into existing workflows. This strategic foresight ensures that the exchanged data not only meets immediate collaboration needs but also aligns with long-term organizational objectives.

In essence, this strategy document serves not only as a guide for implementing data spaces but also as a call to action for decision-makers to consider the full lifecycle of data, from exchange to strategic utilization. As you harness the transformative power of data spaces, foster a forward-looking approach. Beyond collaboration, encompass the intelligent and responsible use of shared data for sustained positive impact and innovation.

Next steps

To embark on your organization's data space journey, reach out to AWS Partner Think-it.

Think-it is a software engineering collective. Their mission is to leverage technology to regenerate our planet and advance human potential. They are pioneers in the operationalization of data space connectors, making sovereign data exchange a reality. Their cutting-edge cross-disciplinary approach is advancing a more sustainable future.

The Think-it initial complimentary offering includes the following:

- The technical modules to build a minimum viable data space (MVDS) so that you can try it out, build ideas, and see for yourself the value you can create. For more information, see the Think-it technical pattern guide.
- A complimentary consultation to guide you through the process and understand your business needs. From there, consultants will provide you with a readiness checklist and scope your next steps, whether you want to customize your orientation to an existing data space or build a new, scalable data space pilot.

Resources

References

- Enabling data sharing through data spaces and AWS (AWS Public Sector blog post)
- Data Act: Commission welcomes political agreement on rules for a fair and innovative data economy
- The European Data Act
- Data spAces for smaRt Energy (DARE)
- Catena-X: Sustainability
- How Does Catena-X Strengthen the Automotive Supply Chain? (Siemens blog post)
- International Data Spaces: Data Spaces Radar
- Gaia-X.eu
- Digital Technologies: The Gaia-X Ecosystem - A Sovereign Data Infrastructure for Europe
- TNO innovation for life: Gaia-X, a European initiative for increased digital sovereignty
- Eclipse Dataspace Components
- European Commission: Preparatory work in view of the procurement of an open source cloud-to-edge middleware platform
- SIMPL: Secure IoT Management Platform
- Post-Platforms Foundation

AWS Partner

- Think-it

Document history

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.

Change	Description	Date
Initial publication	—	February 15, 2024

AWS Prescriptive Guidance glossary

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.

Numbers

7 Rs

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:

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.

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

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

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.

active-passive migration

A database migration method in which 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.

aggregate function

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`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

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.

anti-pattern

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

application control

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

application portfolio

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.

artificial intelligence (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)

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.

asymmetric encryption

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.

atomicity, consistency, isolation, durability (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.

attribute-based access control (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.

authoritative data source

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.

Availability Zone

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.

AWS Cloud Adoption Framework (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 and the AWS CAF whitepaper.

AWS Workload Qualification Framework (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.

B

bad bot

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

BCP

See business continuity planning.

behavior graph

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.

big-endian system

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

binary classification

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?"

bloom filter

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

blue/green deployment

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

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.

botnet

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.

branch

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).

break-glass access

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.

brownfield strategy

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.

buffer cache

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

business capability

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.

business continuity planning (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.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

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.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (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.

chaos engineering

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.

CI/CD

See continuous integration and continuous delivery.

classification

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.

client-side encryption

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

Cloud Center of Excellence (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.

cloud computing

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

cloud operating model

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.

cloud stages of adoption

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

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

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.

CMDB

See configuration management database.

code repository

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 AWS CodeCommit. 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.

cold cache

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.

cold data

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.

computer vision (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, AWS Panorama offers devices that add CV to on-premises camera networks, and Amazon SageMaker provides image processing algorithms for CV.

configuration drift

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.

configuration management database (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.

conformance pack

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.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you 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*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

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

data classification

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.

data drift

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.

data in transit

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

data mesh

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

data minimization

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.

data perimeter

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.

data preprocessing

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.

data provenance

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

data subject

An individual whose data is being collected and processed.

data warehouse

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.

database definition language (DDL)

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

database manipulation language (DML)

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

DDL

See database definition language.

deep ensemble

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.

deep learning

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

defense-in-depth

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.

delegated administrator

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.

deployment

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.

development environment

See environment.

detective control

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*.

development value stream mapping (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.

digital twin

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.

dimension table

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.

disaster

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.

disaster recovery (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.

DML

See database manipulation language.

domain-driven design

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.

DR

See disaster recovery.

drift detection

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.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

edge computing

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.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

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.

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.

endpoint

See service endpoint.

endpoint service

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.

enterprise resource planning (ERP)

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

envelope encryption

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.

environment

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

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

epic

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.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

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.

F

fact table

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.

fail fast

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

fault isolation boundary

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.

feature branch

See branch.

features

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.

feature importance

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.

feature transformation

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", "May", "Thu", and "15", you can help the learning algorithm learn nuanced patterns associated with different data components.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

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.

G

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

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.

Gitflow workflow

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.

greenfield strategy

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.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries. *Detective guardrails* detect policy violations and compliance issues, and generate alerts

for remediation. They are implemented by using AWS Config, AWS Security Hub, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

H

HA

See high availability.

heterogeneous database migration

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.

high availability (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.

historian modernization

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.

homogeneous database migration

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.

hot data

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.

hotfix

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.

hypercare period

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.

I

IaC

See infrastructure as code.

identity-based policy

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

idle application

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.

IIoT

See industrial Internet of Things.

immutable infrastructure

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.

inbound (ingress) 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.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the 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 strategy reduces the risks associated with large migrations.

Industry 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.

infrastructure

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

infrastructure as code (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.

industrial Internet of Things (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.

inspection 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.

Internet of Things (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?

interpretability

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.

IoT

See Internet of Things.

IT information library (ITIL)

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

IT service management (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.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (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.

landing zone

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.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

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

See 7 Rs.

little-endian system

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

lower environments

See environment.

M

machine learning (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.

main branch

See branch.

malware

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.

managed services

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*.

manufacturing execution system (MES)

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

MAP

See Migration Acceleration Program.

mechanism

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.

member account

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.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

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

microservice

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.

microservices architecture

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

Migration Acceleration Program (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.

migration at scale

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.

migration factory

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.

migration metadata

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.

migration pattern

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.

Migration Portfolio Assessment (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.

Migration Readiness Assessment (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.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

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.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and

milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

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

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

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?"

mutable infrastructure

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.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

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.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (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.

operational-level agreement (OLA)

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

operational readiness review (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.

operational technology (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.

operations integration (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

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.

organizational change management (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.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic `PUT` and `DELETE` requests to the S3 bucket.

origin access identity (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.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) 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.

P

permissions boundary

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.

personally identifiable information (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.

PII

See personally identifiable information.

playbook

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.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

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).

polyglot persistence

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

portfolio assessment

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

predicate

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

predicate pushdown

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.

preventative control

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*.

principal

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.

Privacy by Design

An approach in system engineering that takes privacy into account throughout the whole engineering process.

private hosted zones

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.

proactive control

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

product lifecycle management (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.

production environment

See environment.

programmable logic controller (PLC)

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

pseudonymization

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.

publish/subscribe (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.

Q

query plan

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

query plan regression

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.

R

RACI matrix

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

ransomware

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

RASCI matrix

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

RCAC

See row and column access control.

read replica

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.

re-architect

See 7 Rs.

recovery point objective (RPO)

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.

recovery time objective (RTO)

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

refactor

See 7 Rs.

Region

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.

regression

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).

rehost

See 7 Rs.

release

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

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

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.

resource-based policy

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.

responsible, accountable, consulted, informed (RACI) matrix

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 *RASCI matrix*, and if you exclude it, it's called a *RACI matrix*.

responsive control

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*.

retain

See 7 Rs.

retire

See 7 Rs.

rotation

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

row and column access control (RCAC)

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

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

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.

S

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.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

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.

security control

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.

security hardening

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.

security information and event management (SIEM) system

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.

security response automation

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.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (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.

service endpoint

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*.

service-level agreement (SLA)

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

service-level indicator (SLI)

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

service-level objective (SLO)

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

shared responsibility model

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.

SIEM

See security information and event management system.

single point of failure (SPOF)

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

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

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.

SPOF

See single point of failure.

star schema

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.

strangler fig pattern

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.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

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

symmetric encryption

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

synthetic testing

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.

T

tags

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.

target variable

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.

task list

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.

test environment

See environment.

training

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 attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don't know the target.

transit gateway

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.

trunk-based workflow

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.

trusted access

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.

tuning

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.

two-pizza team

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.

U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* 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 deep learning systems guide.

undifferentiated tasks

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.

upper environments

See environment.

V

vacuuming

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

version control

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

VPC peering

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

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

W

warm cache

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.

warm data

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

window function

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.

workload

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

workstream

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.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

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.

Z

zero-day exploit

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

zero-day vulnerability

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.

zombie application

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