Building a Cloud Center of Excellence within your organization

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Building a Cloud Center of Excellence within your organization

Table of Contents

Building a Cloud Center of Excellence within your organization

Amazon Web Services (AWS)

November 2023 (document history)

The goal of this guide is to help you build an effective Cloud Center of Excellence (CCoE) unit within your organization and implement governance within this CCoE. The guide also covers example key performance indicators (KPIs) and structures within a CCoE. This guide is intended for Amazon Web Services (AWS) customers who are migrating to the AWS Cloud. This guide is also for AWS customers and AWS Partners who are consulting for other organizations that are moving to the AWS Cloud.

Understanding the CCoE

A CCoE is a group or team that leads other employees and the organization as a whole in cloud adoption, migration, and operation. The CCoE provides guidance on best practices and governance policies within an organization. Many organizations use different terms for the CCoE, such as Cloud Competency Center or Cloud Capability Center.

By centralizing knowledge and expertise from those involved within the CCoE, your organization can improve efficiency, enhance security and compliance practices, and drive innovation. This can help your organization better serve your end customers and stay ahead of market trends.

The CCoE typically has a wide range of responsibilities, including but not limited to the following:

- Defining and implementing the organization's cloud strategy
- Developing and enforcing cloud governance policies
- Providing training and support to cloud users
- Measuring and optimizing cloud costs
- Driving innovation and continuous improvement in the organization's cloud usage

The CCoE also plays a pivotal role in driving and sustaining cultural change within an organization. The CCoE team works with senior leadership to define a clear, compelling vision for the culture that your organization wants to create. The CCoE team creates a comprehensive change plan that

should include specific initiatives, timelines, and key performance indicators (KPIs) to measure progress. A CCoE does the following:

- Develops communication strategies to ensure that employees understand the reasons behind the culture change and how it aligns with the organization's mission and values.
- Creates programs to involve employees in the change process, gather their input, and make them feel like active participants in the cloud-adoption journey.
- Identifies and trains the culture champions within the organization. These individuals help drive cultural change within their teams and act as ambassadors for the new culture.

Within the central CCoE there can be separate workstreams, or *AWS practices*. An AWS practice is usually focused on a specific technology or industry area, and it can apply to one or multiple geographical areas.

In summary, a Cloud Center of Excellence can be also viewed as Culture Center of Excellence driving and sustaining culture transformation within an organization. It's important to recognize that culture transformation is an ongoing process. The CCoE should continuously monitor and evaluate the culture, making adjustments as necessary to ensure that the changes you want are sustained.

What a CCoE can do for an organization

The intended outcomes from a CCoE can be categorized as external-facing or internal-facing:

- **External facing** – In transformational or advisory roles, CCoE team members advise their own customers on how to set up a CCoE or AWS practice, by sharing their industry thought leadership and internal experience.
- **Internal facing** – CCoE team members create accelerators, and they evangelize AWS internally with field, support, and delivery teams.

Note that you can adopt a hybrid approach, sharing best practices and culture transformation inside and outside of your organization.

How a CCoE can help your organization achieve its goals

It's important to understand your organization's goals so that the CCoE can play a crucial role achieving those goals, especially in the context of cloud adoption and digital transformation. Before you set up a CCoE, consider the following:

- An organization needs to be selective and strategic in deciding where to focus time, resources, and efforts to ensure it's aligning with the long-term strategic goals and objectives. It means that you need to analyze that what your organization does really well. What differentiates you from others, and where do you want to invest to further differentiate yourselves from your peers? The answer can be based on market dynamics, customer needs, and emerging trends. As an example, some organizations differentiate themselves by staying at the forefront of technological advancements. For other organizations, providing exceptional customer service and experience can be a significant differentiator.
- Ask yourself, or your organization, why you want to build a CCoE. Is it to prepare your organization internally to accelerate the cloud journey, to help a customer, or both?

  Tip: If you currently are limited in scale or experience, start with an internal transformation. In an internal transformation, you have the most control of the inputs and outputs. You can then share what you learn externally with other customers.

- Most often, you are not starting from scratch. Rather, you will be building on an existing foundation. For example, you might already have personnel with expertise in cloud technologies.

You might have existing training and development resources for enhancing the cloud knowledge and skills of your workforce. You also might have existing relationships with external consulting or technology organizations that can contribute to cloud-adoption and CCoE activities. Use a strategic approach that maximizes existing assets and resources while adapting to changing market dynamics at the same time:

1. Understand business goals – Where does your business see the biggest opportunity for growth? This can be based on your expansion plans, market research, inputs from the field (Sales), and other sources.
2. Assess locations at the regional and global level – Explore opportunities to enter new markets or expand within existing markets. This can involve targeting new customer segments or geographic regions where there is untapped potential.
3. Use existing resources and skills – Look at what skills your organization currently possesses. Your organization can use the assets, knowledge, and infrastructure already in place. This includes your customer base, brand recognition, technology, and people resources. Seek out fearless innovators who want to increase their positive impact on the business. Seed the team from within your organization, and supplement it by upskilling. Finally, use on the hiring of new resources to fill any gaps.

   Tip: Cloud Readiness Assessment and AWS Learning Needs Analysis are good starting points. Your account-management team can provide more information on these AWS offerings. Details are also mentioned in the Reference section.

4. Assess job market conditions – Skills that are hard to source coupled with notice periods and unreasonable candidate expectations can lead to hiring challenges. Hiring challenges are common, but proactive and strategic approaches can help organizations overcome these obstacles and secure the talent they need to achieve their goals.

- Identify a sponsor for the CCoE. You might have country, geographical, technology, or business unit–specific priorities that inadvertently compete with one another. When choosing a sponsor, consider the following:

  1. Identify a leader or sponsor who has sufficient influence and is empowered to make decisions. The leader should have authority to mandate the suggested changes. A sponsor with no authority can't ensure that actions will be taken to reach your goals. The sponsor plays a critical role in championing the initiative and ensuring that it aligns with your organization's strategic objectives.
  2. Identify the scope, including geographical boundaries, and the limitations of your CCoE.

3. Modify the charter of your CCoE to define the scope. The sample charter can be referenced from the one mentioned in the section Summarizing the steps for establishing your CCoE. After you have updated the charter, replicate the success across the organization.

- After setting up of a CCoE, measure the results:

  1. Set balanced expectations – Expectations for quick results from a CCoE can be understandable. However, it's essential to balance the speed that you want with the realities of cloud transformation and to scope your CCoE accordingly.
  2. Define short-term and long-term goals – Clearly outline objectives to help stakeholders understand what to expect in the immediate future and over the long haul.
  3. Measure progress – Define key performance indicators (KPIs) to measure the impact of the CCoE's initiatives. It's important to keep goals realistic. A CCoE takes time to build and deliver. It's important to establish a governance process to track and communicate progress to stakeholders regularly.

  Remember that while stakeholders want quick results, a successful CCoE focuses both on immediate gains and on building a foundation for sustained cloud excellence, cost-effectiveness, and agility in the long term. Balancing speed with a strategic and measured approach is key to achieving lasting success in the cloud.

- When setting up a CCoE with a focus on delivering both internal and external outcomes, consider a diverse range of personas to ensure that the CCoE can effectively meet its objectives. Here are some example personas for a CCoE with dual internal and external goals:
  - Persona considerations:
    - External outcomes:
      - Customer-facing cloud evangelists
      - Sales and marketing specialists
      - Customer success managers
      - Partnership and alliances managers
      - Solution architects (for external clients)
    - Internal outcomes:
      - Executive sponsor
      - CCoE leader
      - Practice leaders
      - Cloud architects and engineers

    - Finance and procurement specialists

    The personas are covered in more detail in the CCoE functions section.

    Balancing internal and external outcomes within a CCoE requires clear alignment with the organization's overall business strategy. Each persona needs a comprehensive definition of its specific roles and responsibilities related to internal and external goals. The personas should also support the ability to collaborate effectively across these dimensions to drive success.
- Skill considerations:
    - External outcomes might require resources with a management consulting background.
    - Internal outcomes might require resources with a higher focus on technology consulting.

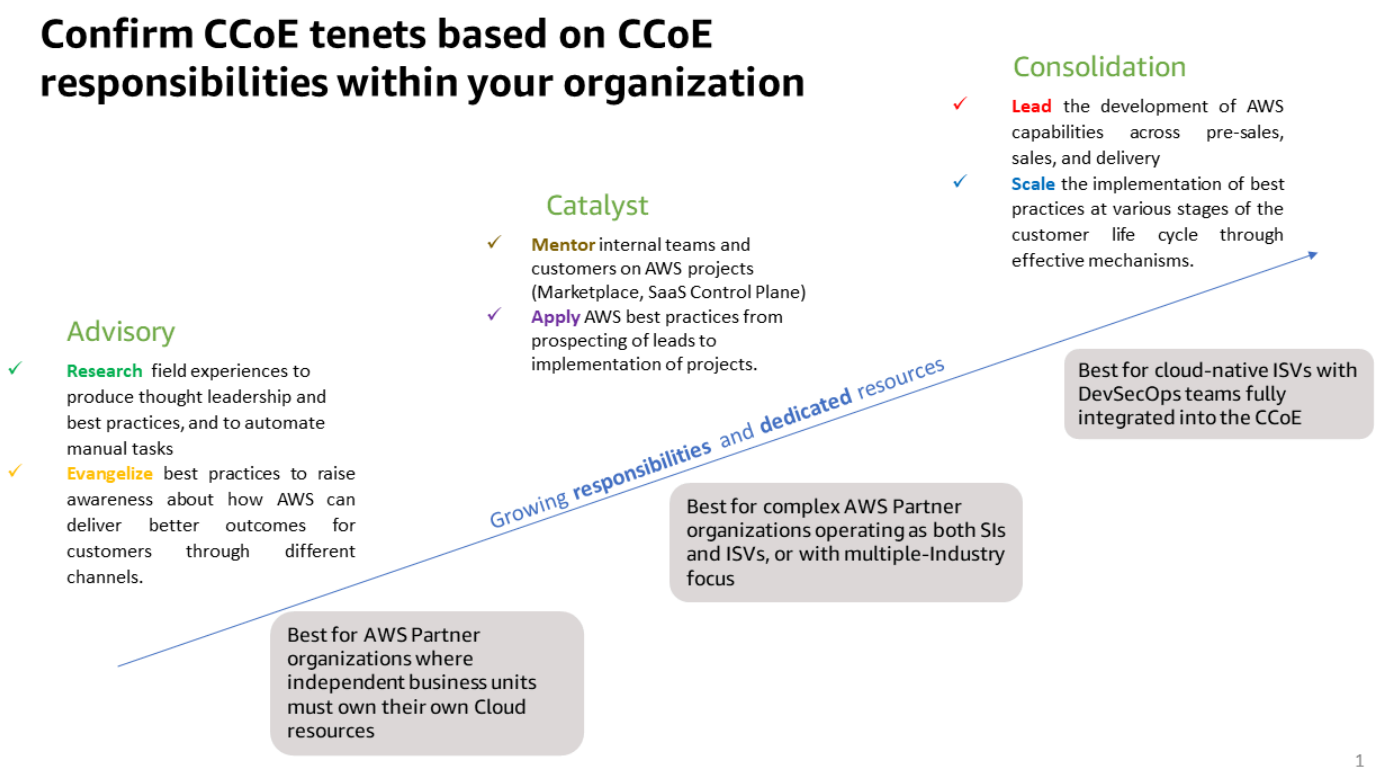
The CCoE phases

Each phase of the CCoE is mapped to the :AWS Cloud Adoption Framework (AWS CAF). The AWS CAF uses AWS experience and best practices to digitally transform and accelerate the business outcomes through innovative use of AWS. The AWS CAF identifies specific organizational capabilities that underpin successful cloud transformations. These capabilities provide best-practice guidance that helps you improve your cloud readiness.

The AWS CAF recommends four iterative and incremental cloud transformation phases:

- **Envision phase** – Demonstrating how the cloud will help accelerate your business outcomes
- **Align phase** – Identifying capability gaps and creating strategies to improve your cloud readiness, ensure stakeholder alignment, and facilitate relevant organizational change-management activities
- **Launch phase** – Delivering pilot initiatives in production and on demonstrating incremental business value
- **Scale phase** – Expanding production pilots and business value to the targeted scale and ensuring that the business benefits associated with your cloud investments are realized and sustained

The following diagram shows CCoE phases that are mapped to different phases of the AWS CAF.



- **Advisory phase** – In this phase, the central CCoE team focuses on gaining organizational awareness and alignment on building a business through AWS. It serves as an early adopter for cloud projects, and it identifies and promotes the value of these engagements within the involved entities. To secure long-term goals for the AWS practice, the central team removes preliminary blockers and identifies early needs such as headcount, skills, and material resources. The Advisory CCoE phase relates to the Envision and Align phases in the AWS CAF.
- **Catalyst phase** – The central CCoE team becomes the AWS champion. It's proactive in driving how the AWS part of the business is run in the context of the organization's overall business strategy, and it supports the other entities with technical developments, AWS enablement, and go-to-market strategies. Its main goals will be defined by the challenges that prompted the formation of the CCoE, which can be different for your business:
  - For AWS customers: To accelerate the migration and modernization of your IT estate to secure AWS Cloud based products and services
  - For AWS Partners: To help your organization reach a profitable status for AWS practices so that they benefit your overall business—for example, by boosting sales and reducing operational costs
  - For AWS customers and AWS Partners: To ensure that the different entities can operate without conflicting interests or processes

The Catalyst CCoE phase relates to the Launch phase in the AWS CAF.

- **Consolidation phase** – Independent practices under the centralized CCoE have reached a volume of AWS projects that makes a positive impact on their profitability, and are self-sufficient in the delivery of such projects. The CCoE shifts to a supporting role, performing tasks that continue to benefit from economies of scale, scope, and knowledge. setting organization standards and best practices, and providing curated training material. To develop specialized expertise (for example, in cloud security and machine learning), consider allocating at least 20 percent of the time on learning and experimenting with new services and new features. The Consolidation CCoE phase relates to the Scale phase in the AWS CAF.

You can analyze your current maturity level, and based on your goals, you can decide where you want to see your organization in short-term and long-term cycles.

The CCoE tenets

A Cloud Center of Excellence (CCoE) typically operates based on a set of tenets or guiding principles that help shape its mission and activities. These tenets provide a framework for the CCoE's operation, and they align its efforts with the organization's broader goals and cloud strategy. While the specific tenets might vary from one organization to another, you can start with the following common CCoE tenets (often known as *REALMS*). Note that these tenets are currently documented from the perspective of AWS Partners, but any AWS customer can define KPIs that support their own cloud journey:

- **Research** means that, based on field experiences and value propositions, AWS Partners can decide which areas to explore, create best practices, and automate manual tasks to deliver business outcomes or benefits to their customers.
  - An example KPI is the number of new solution offerings to be developed in a particular time frame
- **Evangelize** means to share best practices and transfer knowledge across internal teams to raise awareness about how the AWS Partner can deliver better outcomes for its end customers. There can be multiple ways to accomplish this, including internal events, offsites, blog posts, and whitepapers.
  - An example KPI is the number of webinar events, thought leadership materials (for example, blog posts and whitepapers), and training sessions.
- **Apply** involves developing an end-to-end roadmap, from identifying prospecting leads to implementing customer projects.
  - An example KPI is the total number of pilot or proof-of-concept (PoC) implementations.
- **Lead** means leading the development of the AWS Partner's capabilities across presales, sales, and delivery teams through PoC, pilot, minimum viable product (MVP), and first customer wins.
  - An example KPI is the number of customer wins and the win ratio.
- **Mentor** means helping other internal teams and customers to onboard on AWS projects.
  - An example KPI is the implementation of and participation in mentorship programs, communities of practice, and shadowing opportunities.
- **Scale** is about the implementation of best practices at various stages of the end-customer lifecycle to create effective, reusable patterns.

- Example KPIs are the number of services at released at AWS Marketplace, the number of subscriptions to those services, AWS Competency achievement, AWS Service Delivery Program validation, and movement toward earning the next AWS Services Partner Tier.

The next section discusses each of the tenets in more detail and provides questions to help identify the relevant KPIs that align with the overall business goals.

Evaluating the CCoE KPIs

The previous section introduced the CCoE tenets. Using some questions, this section discusses how you can support your CCoE to work toward those tenets. Later, this will help you derive the relevant list of KPIs to measure the impact of the CCoE.

Research tenet

- **Business goals** – What is your current footprint in terms of geography, industry, and customer segments? For example, is your organization a small or medium-size business, or is it an enterprise? What are your plans for expansion in the next year?
- **AWS practices** – What AWS practices are needed to support your business goals? Skill needs will vary with each practice. Existing skill availability varies. When staffing your CCoE, consider a pyramid-based approach, with varying levels of experience in a given skill area.
- **Skill locations** – How do your current locations and skills availability align? Create an organizational map that shows resources within the practice, including the locations where they are operating.

  Tip: Because notice periods are often substantial and vary by location, we recommend identifying the to-be-hired (TBH) positions up front. Identify resources who perform multiple roles and the time frame in which to re-prioritize their workload. This gives you a view of what the resourcing effort will look like.

- **Resource skill matrix** – Capture the current skill alignments of the CCoE (if already staffed) and your wider organization. This will help you plan for resourcing appropriately.

  Tip: To identify the current footprint and potential training needs, perform an AWS Learning Needs Analysis exercise. To learn more about this exercise and it how can be conducted for your organization, reach out to your AWS Enablement Manager. You can also use any organization-wide tagging of skills that might already be in place (drawn from the HR resource onboarding process).

Evangelize tenet

- **Communication plan** – Set up mechanisms to engage field teams and evangelize the CCoE: Your field teams (local CEOs, business unit leads, profit and loss (P&L) leads, account leads, sales, presales, bid, and pricing) must view your CCoE as a collaborative partner in helping your customers. Field teams need to understand how the CCoE can help them in this process.

  Internal roadshows or town-hall sessions are a good vehicle for engagement. Newsletters and internal portals can also help disseminate information to your field teams. Plan for both one-time and ongoing engagements with the field teams.
- **Asset usage** – The CCoE will lead the effort in developing assets to help reduce delivery cost, provide your workforce with relevant skills, and support the sales and bid processes. It's important to define a process to track the usage of these assets by the field teams. This will tell you what is working, what is not, and what needs to change.

  You can systemically track downloads of assets and views of pages. Incentivize field teams to ask the CCoE questions (for example, use a points system). The CCoE Project Management Office (PMO) can follow up and ask for feedback.
- **Feedback mechanism** – Define a process that the field teams can follow to provide feedback to the CCoE. Also define how the CCoE can advertise or market their assets internally. Examples include how many ideas or how much feedback a team or resource is contributing. Marketing mechanisms include an existing web portal, customer satisfaction (CSAT) scoring, and real-time feedback.
- **Usage encouragement** – Think of how you will incentivize your field teams to collaborate with the CCoE. The CCoE should not be considered an extension of your delivery team. Instead, they should be aligned with your field teams and empowered to evangelize when delivering value for your customer.

  Tip: To encourage field teams and the CCoE to work with each other, use non-monetary incentive options. Examples include thank-you cards, email from senior leadership, and vocal recognition in team meetings.

Apply tenet

- **Feedback flywheel** – Define a mechanism to capture inputs from your field teams. Field teams should have a processes for sharing lessons learned and field experiences with the CCoE team so that the CCoE can incorporate the information into their asset roadmap.

  Tip: Supplement the offline feedback from the field teams with regularly scheduled meetings to ensure that the CCoE and field teams are fully aligned.

- **Information dissemination** – How will the AWS business practice and the CCoE team propagate the best practices, assets, and other deliverables to the field teams?
- **Bid process and presales support** – How will the CCoE support the bid and presales teams during request for proposal (RFP) responses?

  Tip: The CCoE can own the solution and provide subject-matter expert (SME) inputs and estimation inputs.

Lead tenet

- **Delivery consulting** – CCoE resources can help accelerate the delivery phase for your customers through limited-duration consulting to your existing delivery teams.

  Tip: Define a *loaning* process for CCoE resources to temporarily assist delivery teams. The loaning process can include the percentage of time spent on the consultation.

- **Engagement model** – How long will a CCoE member remain engaged to support a delivery team? Is the engagement short, medium, or long term? Such a consulting or engagement model should not be more than few weeks. CCoE resources are not replacements for your delivery team.

Mentor tenet

- **Community of practice** – To create a community of practice, foster mentoring opportunities. This will create an inclusive atmosphere and encourage other employees to learn more and

contribute. This can include programs such as aspiring area of depth, where employees can pursue their interests and build their careers while they help your organization and the customer.

- **Crowdsourcing knowledge** – How do you ensure the benefits of CCoE are not limited just to those working on the requests for proposal (RFPs), but are available to all employees? One way is to use a mechanism such as an Answer portal, where technical questions could be submitted by any employee. CCoE resources can review questions and provide feedback.
- **Training the trainer for CCoE** – To make the CCoE a force multiplier for itself, use a *train the trainer* approach. After you have staffed motivated resources for the CCoE, you can consider developing an approach in which experts in one skill can gradually upskill themselves in other areas.

  Tip: To support upskilling, use shadowing and reverse shadowing.

Scale tenet

- **CCoE front door** – What is the mechanism for the field teams to gain access to the CCoE resources? How do you plan to scale the CCoE operations efficiently? Consider creating a dedicated Project Management Office (PMO) to handle the day-to-day operations of the CCoE. The PMO resources can handle any undifferentiated heavy lifting in the CCoE operations.
- **Self-service mechanisms** – What types of self-service mechanisms can you put in place for the field teams to find information? For example, what assets, collateral, and past experiences will help the field during the sale and delivery stages?

  Tip: Use Amazon Bedrock to build customized generative AI solutions to help your field teams quickly access your CCoE assets.

- **CCoE scope** – What are the plans for incorporating the other functions (for example, Legal, Fin-ops, Contracting, and Account Leadership) into the scope of the CCoE? Typically, these are existing functions within organizations. Having them under the CCoE banner promotes consistency and a one-team behavior.
- **CCoE footprint** – How do you plan to expand the size of your CCoE? We recommend planning for growth based on your business's growth. Because the CCoE is a strategic investment, align its growth with your overall targets. After you finalize headcount projections, you can plan for hiring and lateral movements.

- **Incentivizing innovation** – Think about how to incorporate an incentive mechanism to encourage CCoE resources to innovate continuously.
- **Performance management of CCoE resources** – The resources who are part of your CCoE should be able to grow within your organization while being part of the CCoE. Review your current performance-management practices in light of the roles that CCoE resources are expected to perform, and make adjustments as needed.
- **Recognition of CCoE resources** – Establish a plan for recognizing performance and success within this part of the organization.

CCoE functions – Engineering and business

The CCoE functional scope can be separated into engineering functions and business functions. Clearly define which functions are in scope of the CCoE based on specific goals and priorities.

Engineering functions

The engineering functions of the CCoE help your organization maximize the technical benefits of using AWS Cloud services. They relate to the implementation of a series of functions and best practices reflecting your technical knowledge:

- **Cloud infrastructure**
  - Core networking capabilities to integrate the corporate network with AWS
  - Setup of AWS Control Tower landing zones, accounts, AWS Identity and Access Management (IAM) roles and policies, and federation with the corporate directory
  - Infrastructure as code (IaC) using standardized, automated deployments of integrated primitives with configuration management
- **Architecture alignment**
  - Development and publication of cloud reference architectures aligned with the enterprise architecture
  - Breakdown and analysis of technical requirements mapped against cloud reference architectures and the roadmap
  - Enterprise cloud vision, strategy, roadmap, and delivery
- **Operations**
  - Monitoring infrastructure, and providing best practices and operational insights
  - Resiliency mechanisms and best practices to provide patch management, backup, and restore capabilities
  - Providing CI/CD infrastructure, with best practices for building development, security, and operations (DevSecOps) teams
  - Software delivery, including ownership of the AWS Marketplace listing process
- **Security, risk, and compliance**
  - Management of cloud workload security, including threat and vulnerability management, security information and event management, IAM policy management, network security, and secrets and encryption

- Management of security incident response, quarantine, analysis, and forensics
- Risk management, addressing the security, risk, and compliance needs of cloud migrations
- Compliance management, providing advisory services on the implementation of robust security, risk, and compliance solutions for the cloud migrations

- **Technical excellence**
  - Capability uplift, including training and certification to demonstrate required knowledge and skills obtained
  - Exploration and expertise in new technical areas relevant for the core business
  - Creation of training plans for all personas in the organizational business units
- **Cloud optimization**
  - Optimizing the performance and cost-effectiveness of the organization's cloud environment
  - Identifying opportunities to improve performance, reduce costs, and right-size resources

Business functions

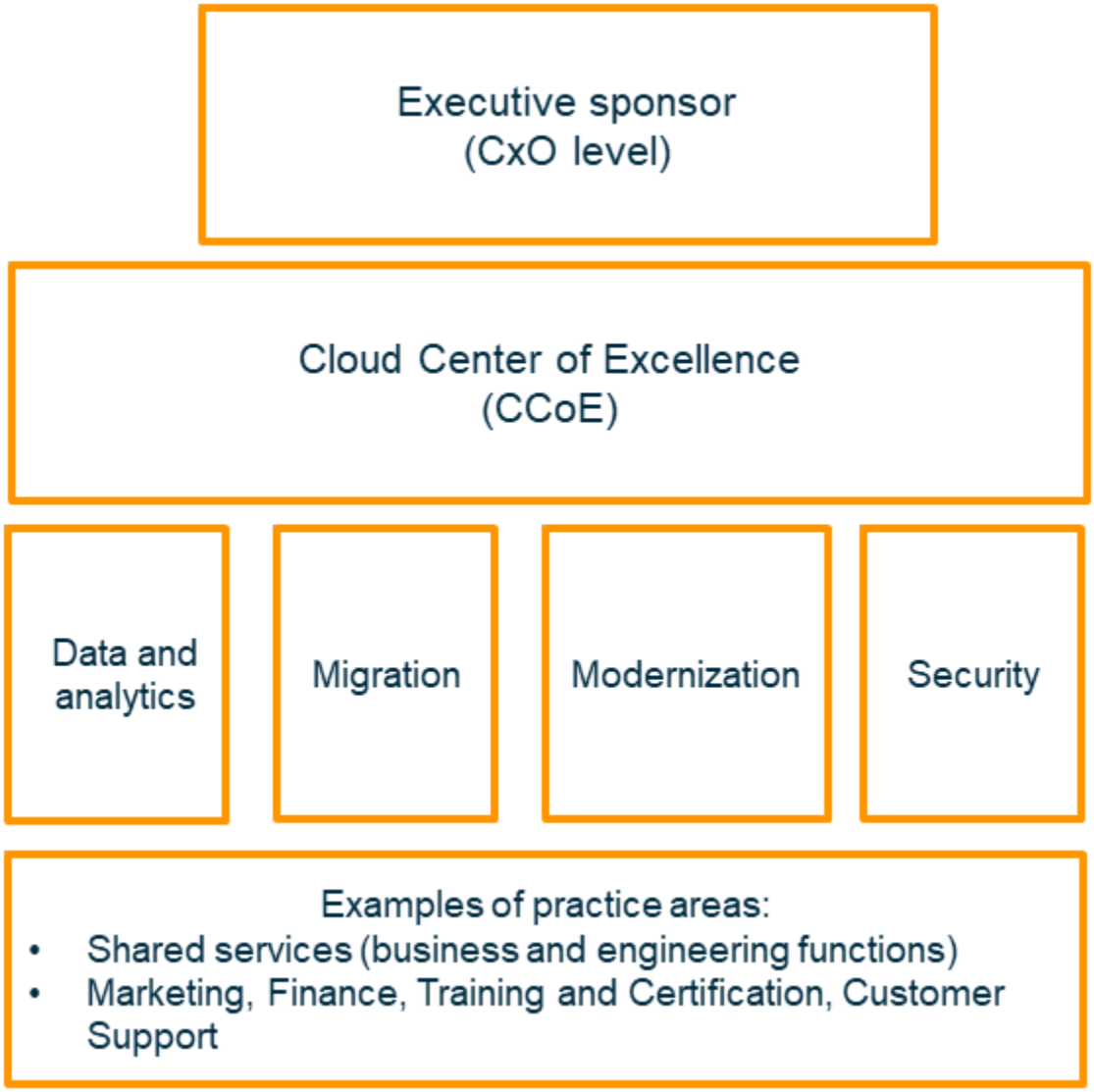
The business functions of the CCoE help your organization accelerate your business and optimize the benefits of using AWS Cloud services:

- **Sales cycle acceleration**
  - Creation of field ready kits, including first-call decks, sales briefs, solution briefs
  - Support for the entire sales cycle, from lead generation to contract signing
  - Enablement, including awareness sessions and sales-team coaching on cloud solutions
- **Marketing**
  - Creation of case studies, blog posts, videos, and technical content that supports other marketing activities (for example, advertising, email marketing, positioning, influencer marketing)
  - Events to increase brand awareness and generate leads by supporting the organization of and participation in events with AWS
- **Delivery support**
  - Migration of legacy services toward cloud-native services, optimizing the new application's user-onboarding process
  - Implementation of agile delivery frameworks and removal of roadblocks

  - Cloud services expertise to support deployments, consolidate lessons learned, and help identify risks and opportunities
- **Finance management**
  - Continuous optimization of cloud asset allocation compared with usage, and implementation of AWS tools for reporting and cost optimization
  - Self-service dashboards, such as the Cost Intelligence Dashboard, so that external customers have visibility into the cost of your solution, and internal stakeholders can access cloud-consumption metrics
  - Invoice management – Breakdown of the cloud invoice to allocate spend at the business-unit level
- **Project Management Office (PMO)**
  - Market studies and technology watch to support portfolio management
  - Project management, including identification of synergies between different cloud projects
  - Centralized governance with a view of all cloud initiatives
  - Coordination of all engagements with AWS. For AWS Partners, acquisition of specific competencies and service delivery designation from the AWS Partner Network.

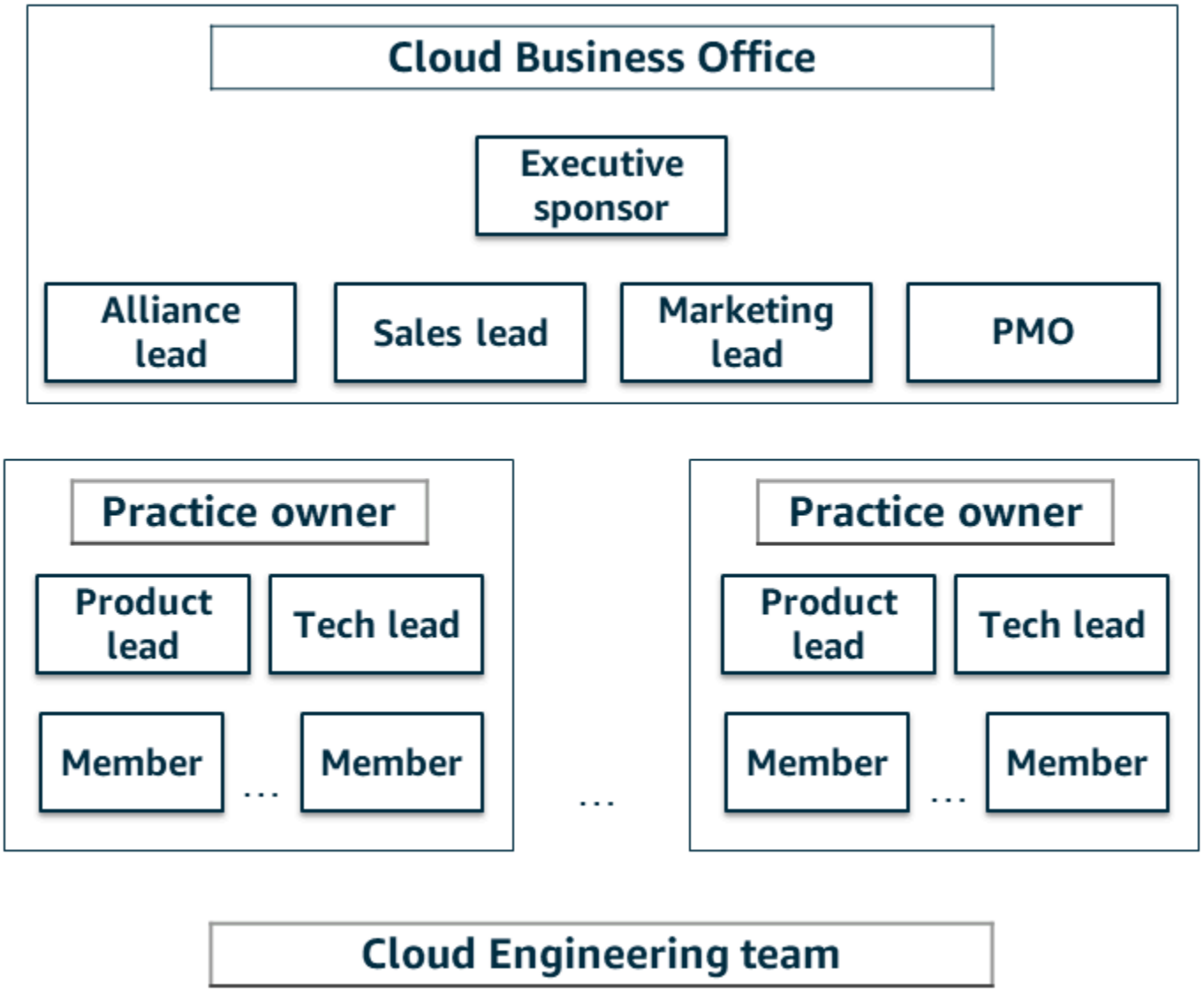
Example CCoE structure

The following diagram shows an example CCoE organizational structure.



Under the shared services, you can choose different engineering functions and business functions to complement the different practice areas. In the diagram, the practice areas are Marketing, Finance, Training and Certification, and Customer Support.

Under each practice area, the expectation is to have a single practice owner work with the Product Technical Leads and the Delivery team members who will deliver the projects. Each practice owner will be responsible for the targets and KPIs for their individual practice and will report to the Cloud Business Office (CBO) team, as shown in the following diagram.



The CBO is the central hub of the CCoE. It's responsible for developing and implementing the cloud strategy, developing and enforcing cloud governance policies, and managing the cloud budget. The CBO also oversees the work of the Cloud Engineering team.

The Cloud Engineering team is responsible for the technical aspects of the organization's cloud environment. This includes designing, migrating, and operating cloud workloads. The Cloud Engineering team also works to ensure the security and compliance of the cloud environment.

Summarizing the steps for establishing a CCoE

Setting up a Cloud Center of Excellence (CCoE) is a strategic initiative that can help your organization effectively plan, govern, and optimize your cloud-adoption efforts. A CCoE is a cross-functional team responsible for driving cloud best practices, innovation, and governance within an organization. You can use the following example steps to set up a CCoE. However, it's important to note that the steps might vary, depending on the maturity and needs of your organization.

1. **Define objectives and goals** – Start by clearly defining the objectives and goals of your CCoE. Understand why you are establishing it and what you hope to achieve. Common objectives include cost optimization, security, compliance, and innovation.
2. **Build a cross-functional team** – Assemble a team of experts from various departments, including IT, security, finance, compliance, and operations. The team should represent a range of skills and knowledge related to cloud technologies.
3. **Identify leadership and accountability** – Appoint a CCoE leader or manager who will be responsible for its success. Make sure that this leader has the authority to make decisions and can drive cloud initiatives.
4. **Create a charter** – Develop a charter or mission statement that outlines the purpose, scope, responsibilities, and authority of the CCoE. Share this with the organization to set clear expectations. The following table provides an example charter that you can modify depending on your specific scenarios.

Mission statement	Governance	Deliverables	KPIs
• Codify patterns in use or planned. Patterns include standard Amazon Machine Image (AMI) images, configuration management, and AWS CloudForm ation templates.	• Weekly meetings • Monthly reporting to the CCoE PMO	**3 months** • AWS Control Tower landing zone as a foundation to onboard business units and applicati ons • Reference architectural patterns with	**3 Months** • Architectural patterns exist with clear annotations. **6 months** • Reusable products in AWS Service Catalog **12 months**

- Publish patterns to the enterpris e AWS Service Catalog.
- Identify and prioritize future patterns.

approved AMIs and baked-in security

6 months

- Self-service catalogs
- Monitoring and logging
- CI/CD and automated testing
- Cloud Migration and Applicati on Lifecycle playbooks
- Prioritized backlog of additiona l architecture patterns

12 months

- Solution built using CI/CD pipelines and DevOps tooling for next-gen products
- Extensive infrastru cture support for the majority of your use cases

- Additional architectural patterns to work on are prioritized in the backlog.

5. **Develop cloud expertise** – Provide training and resources to the CCoE team members to enhance their cloud expertise. Ensure that they are up to date with the latest cloud technologies and best practices.
6. **Establish a governance framework** – Define cloud governance policies and procedures to help ensure compliance, security, and cost control. This might include creating cloud usage policies, access controls, and resource-tagging standards.

7. **Manage costs** – Implement cost-management practices to monitor and control cloud spending. Set up budgets, use cost allocation tags, and regularly review cloud bills for optimization opportunities.
8. **Manage security and compliance** – Develop security and compliance guidelines specific to your organization's needs. Implement security best practices, conduct regular security audits, and confirm compliance with industry standards and regulations.
9. **Define cloud architecture and best practices** – Encourage teams to follow these guidelines when designing and building cloud-based applications and infrastructure.
10. **Innovate and automate** – Foster innovation by exploring new cloud services and technologies that can benefit your organization. Encourage automation to improve efficiency and reduce manual processes.
11. **Collaborate and communicate** – Facilitate communication and collaboration between the CCoE and other departments or teams in the organization. Regularly share updates, successes, and lessons learned.
12. **Share knowledge** – Create a knowledge-sharing platform or repository where best practices, documentation, and case studies related to cloud adoption can be stored and accessed by the organization.
13. **Measure and define KPIs** – Define KPIs to measure the success of your CCoE. These KPIs can include cost savings, security incidents, compliance levels, and adoption rates.
14. **Continuously improve** – Continuously review and improve the processes, policies, and practices of the CCoE based on feedback and changing organizational needs.
15. **Report regularly** – Provide regular reports and updates to senior leadership to demonstrate the value and impact of the CCoE on the organization's cloud adoption journey.
16. **Promote feedback and adaptation** – Encourage feedback from stakeholders. Be ready to adapt and evolve the CCoE's strategy and activities based on changing business requirements and technology trends.

Do's and don'ts

The following lists provide quick reminders of best practices to use when building a CCoE for your organization.

Do's

- Do align the CCoE's goals and initiatives with the organization's broader business objectives.
- Do appoint a capable and empowered leader to oversee the CCoE. This leader should have the authority to make decisions and drive cloud initiatives.
- Do facilitate communication and collaboration between the CCoE and other departments or teams. Regularly share updates and seek input from stakeholders.
- Do establish a robust cloud-governance framework that includes policies, procedures, and best practices for security, compliance, and cost management.
- Do encourage knowledge sharing within the CCoE and across the organization. Create a repository of best practices, documentation, and case studies.
- Do facilitate communication and collaboration between the CCoE and other departments or teams. Regularly share updates and seek input from stakeholders.
- Do define key performance indicators (KPIs) to measure the success of the CCoE's initiatives. Use these KPIs to demonstrate value to leadership.

Don'ts

- Don't proceed without clearly defined objectives and scope for the CCoE. Vague or overly broad goals can lead to confusion.
- Don't operate the CCoE in isolation. Collaboration and communication with other departments are essential for success.
- Don't focus solely on short-term goals. A successful CCoE should have a long-term vision for cloud excellence.

Conclusion

Establishing a Cloud Center of Excellence (CCoE) is not just a trend. It's a strategic move that can transform the way organizations approach cloud adoption. CCoEs provide a structured framework for achieving better governance, enhanced security, cost optimization, and continuous innovation in the cloud. While challenges might arise along the way, with the right leadership, a cross-functional team, and a commitment to best practices, these challenges can be overcome.

As you consider the potential benefits of a CCoE for your organization, remember that successful cloud adoption is an ongoing journey.

Whether you're a cloud enthusiast or a decision-maker looking to drive digital transformation, your proactive steps today can shape a more agile and resilient future for your organization. Start by sharing this article with your colleagues and engaging in the conversation about the power of the CCoE.

Resources

- Cloud Transformation Maturity Model: Guidelines to develop effective strategies for your cloud adoption journey
- AWS Learning Needs Analysis
- AWS Cloud Adoption Framework

Contributors

Contributors to this guide include:

- Rishi Singla, Senior Partner Solutions Architect, AWS
- Guillaume Goutaudier, Senior Enterprise Architect, AWS
- Shankar Subramaniam, Senior Enterprise Architect, AWS
- Steve Drew, Senior Enterprise Architect, AWS
- Jonathan Cornell, Manager, Partner Enterprise Architecture, AWS

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	—	November 15, 2023

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. This migration scenario is specific to VMware Cloud on AWS, which supports virtual machine (VM) compatibility and workload portability between your on-premises environment and AWS. You can use the VMware Cloud Foundation technologies from your on-premises data centers when you migrate your infrastructure to VMware Cloud on AWS. Example: Relocate the hypervisor hosting your Oracle database to VMware Cloud on 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 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.