AWS Security Reference Architecture (AWS SRA) – IoT

AWS Prescriptive Guidance

AWS Prescriptive Guidance: AWS Security Reference Architecture (AWS SRA) – IoT

Table of Contents

AWS Security Reference Architecture (AWS SRA) – IoT

Global Services Security Team, Amazon Web Services (contributors)

December 2025 (document history)

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

Internet of Things (IoT) refers to the collective network of connected devices and the technology that facilitates communication among devices and between devices and the cloud. IoT implementations pose unique considerations that don't apply to traditional IT deployments. There are three types of IoT implementations: consumer IoT deployments, industrial IoT (IIoT) deployments, and operational technology (OT) deployments. Each of these implementations has a distinct set of security requirements.

- Consumer IoT solution deployments, such as robotic vacuums and other consumer IoT devices, use AWS to handle scale and spikes. These implementations can introduce a new classification of security considerations to address. These security considerations and challenges include, but aren't limited to:
  - Difficulty in managing and securing a wide range of device types at scale
  - Constrained resources such as compute, storage, and network, which limit the availability of robust security features
  - The possible lack of automated update and patching mechanisms
- IIoT solution deployments include implementations by automotive, pharmaceutical, and other manufacturing companies that use AWS IoT SiteWise. These implementations can optimize production processes, reduce costs, and provide a better experience for your customers. However, there are unique security considerations that stem from integration with OT systems, real-time operations, and physical processes.
- IoT deployments that are based on OT or supervisory control and data acquisition (SCADA), such as those adopted by mining, energy, and utilities companies, use various AWS IoT services to improve operational efficiencies and reduce operational cost. These implementations pose additional challenges associated with secure OT and IT convergence. These involve safety-critical systems, proprietary and often legacy industrial protocols, and diverse operating environments.

Note

This guidance focuses on security best practices that are relevant to the growing list of use cases that involve IoT, IIoT, and OT-based solutions on AWS. Future updates will iteratively expand the scope and add guidance to include the full array of relevant AWS services and features for this domain.

In this guide:

About the AWS SRA library

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

This guide is part of a library that provides architectural blueprints and technical guidance for designing and building security architectures on AWS. The library consists of implementation code (AWS SRA code library), a validation tool (SRA Verify), and two complementary categories of guides that cover the core architecture and deep dive architectures.

AWS SRA – core architecture

This guide represents a foundation for the recommended AWS security architecture. It is the starting point that applies to all organizations, regardless of their industry, application type, or any other considerations. This foundation helps you build a strong and scalable architecture on AWS and helps create a strong AWS multi-account security baseline that securely scales as your business grows.

AWS SRA – deep dive architectures

The *AWS SRA – core architecture* guide is complemented by additional publications that provide architectural patterns aligned to specific security capabilities, application types, and compliance or regulatory requirements. These patterns extend the core architecture and should be used in conjunction with the *AWS SRA – core architecture* guide.

The following guides provide architectural patterns aligned to specific security capabilities:

- AWS SRA – identity management provides guidance on how to implement a scalable, robust, and centralized identity and access management solution on AWS.
- AWS SRA – perimeter security discusses architecture patterns and AWS services for implementing edge security in a central account or in individual accounts.
- AWS SRA – cyber forensics describes how to configure an AWS Forensics account as a starting point to develop your organization's forensic capabilities and to help improve your security incident response (IR) preparedness.

The following guides provide architectural patterns for specific application types. You might want to focus on these after you build your baseline security architecture:

- AWS SRA – AI security provides security architectural recommendations for designing and building applications that incorporate generative AI capabilities by using AWS generative AI services.
- *AWS SRA – IoT* (this guide) provides security architectural recommendations for designing and building IoT applications on AWS.

In addition, the following guide describes architectural patterns that are aligned with specific compliance or regulatory frameworks:

- AWS Privacy Reference Architecture (AWS PRA) provides a security architecture for applications that process personal data and must support broad privacy compliance requirements such as the General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), or the Brazilian General Data Protection Law (LGPD). The AWS PRA provides a set of guidelines that are specific to the design and configuration of privacy controls in AWS services.

We recommend that you start with the *AWS SRA – core architecture* guide to understand the foundational architecture and then consult the complementary guides to take advantage of advanced functionality and implementations. For more information about this content set, see AWS Security Reference Architecture.

Tip

To customize the reference architecture diagrams in the AWS SRA library based on your business needs, you can download the following .zip file and extract its contents.

Download the diagram source file (Microsoft PowerPoint format)

IoT for the AWS SRA

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

This guide provides recommendations for using IoT securely in industrial and critical infrastructure environments to improve the productivity and efficiency for users and organizations. It focuses on the use of AWS IoT services based on the AWS SRA holistic set of guidelines for deploying an array of AWS security services in a multi-account environment.

This guidance builds upon the AWS SRA – core architecture to enable IoT capabilities within an enterprise-grade, secure framework. It covers key security controls such as device identity and asset inventory, AWS Identity and Access Management (IAM) permissions, data protection, network isolation, vulnerability and patch management, logging, monitoring, and incident response that's specific to AWS IoT services.

The target audience for this guidance includes security professionals, architects, and developers who are responsible for securely integrating IoT solutions into their organizations and applications.

AWS SRA best practices for IoT

This section explores security considerations and best practices for IoT workloads adapted from the best practices described in the AWS blog post Ten security golden rules for industrial IoT solutions. These AWS SRA best practices for IoT are:

1. Assess OT and IIoT cybersecurity risks.
2. Implement strict separation between OT (or IIoT) environments and IT environments.
3. Use gateways for edge computing, network segmentation, security compliance, and to bridge administrative domains. Harden IoT devices and minimize their attack surface.
4. Establish secure connection with AWS by using AWS Site-to-Site VPN or AWS Direct Connect from the industrial edge. Use virtual private cloud (VPC) endpoints whenever possible.
5. Use secure protocols whenever possible. If you use insecure protocols, convert these into standardized and secure protocols as close to the source as possible.
6. Define appropriate update mechanisms for software and firmware updates.

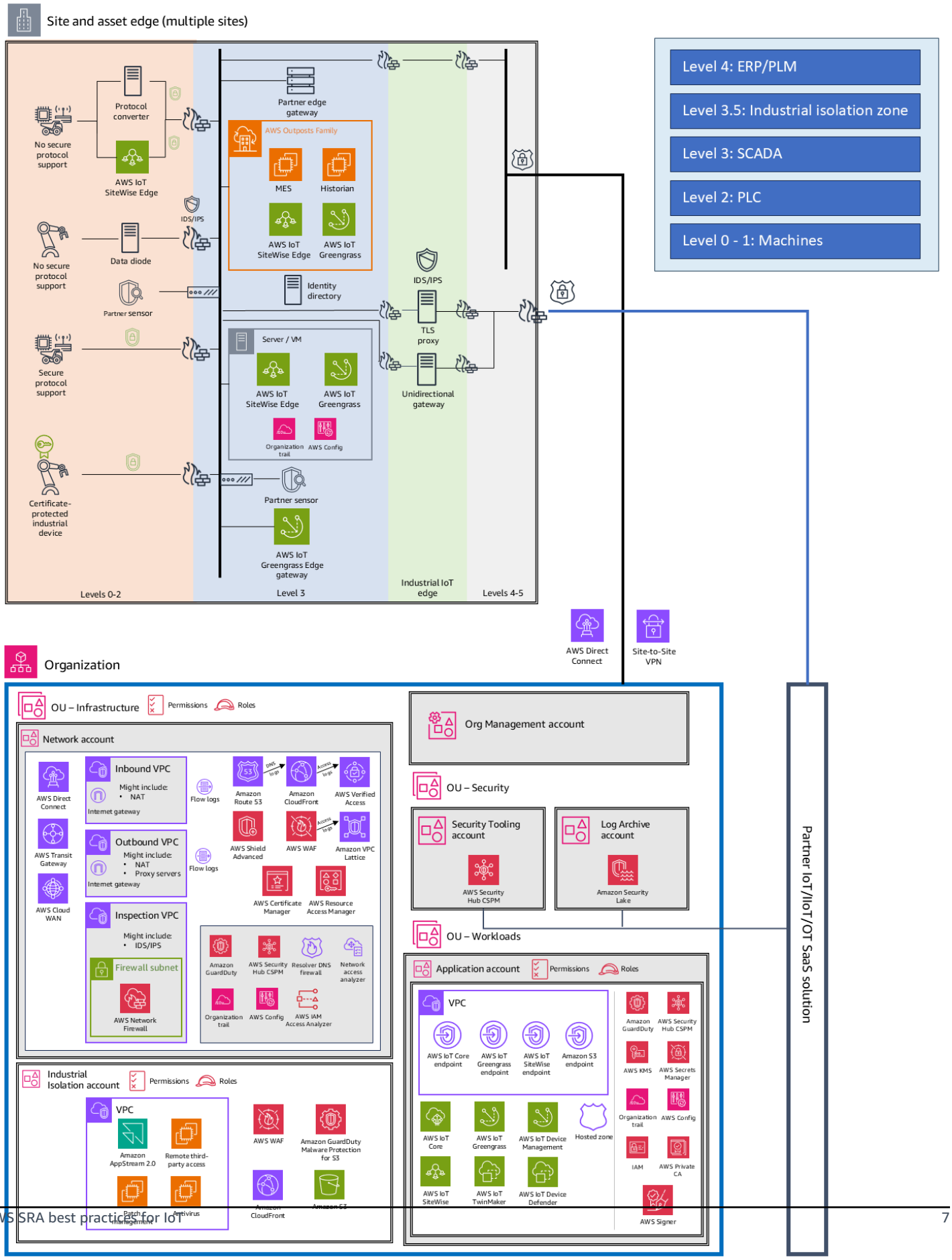
7. Implement device identity lifecycle management. Apply authentication and access control mechanisms.
8. Secure IoT data at the edge and in the cloud by encrypting data at rest and in transit. Create mechanisms for secure data sharing, governance, and sovereignty.
9. Deploy security auditing and monitoring mechanisms across OT and IIoT. Centrally manage security alerts across OT (or IIoT) and the cloud.
10. Create incident response playbooks and a business continuity and recovery plan. Test the plan and procedures.

To implement these best practices, this guidance covers the following capabilities:

- Capability 1. Providing secure edge computing and connectivity (best practices 3, 4, and 5)
- Capability 2. Providing an industrial isolation zone between environments (best practice 2)
- Capability 3. Providing strong device identities and secure device access and management (best practices 6 and 7)
- Capability 4. Providing data protection and governance (best practice 8)
- Capability 5. Providing security monitoring and incident response (best practices 9 and 10)

The following sections of this guidance expand on each capability, discuss the capability and its usage, cover security considerations pertaining to the capability, and explain how you can use AWS services and features to address the security considerations (remediation).

The architecture illustrated in the following diagram is an extension of the AWS SRA diagram depicted in the *AWS SRA – core architecture* guide. It adds the following elements: customer site and industrial IoT edge, industrial isolation zone account, and IoT, IIoT, or OT software as a service (SaaS) security solutions from AWS Partners.


Site and asset edge (multiple sites)
Level 4: ERP/PLM
Level 3.5: Industrial isolation zone
Level 3: SCADA
Level 2: PLC
Level 0 - 1: Machines
Protocol converter
No secure protocol support
AWS IoT SiteWise Edge
Partner edge gateway
AWS Outposts Family
MES
Historian
AWS IoT SiteWise Edge
AWS IoT Greengrass
IDS/IPS
No secure protocol support
Data diode
Partner sensor
Identity directory
IDS/IPS
TLS proxy
Server / VM
AWS IoT SiteWise Edge
AWS IoT Greengrass
Organization trail
AWS Config
Unidirectional gateway
Secure protocol support
Certificate-protected industrial device
Partner sensor
AWS IoT Greengrass Edge gateway
Levels 0-2
Level 3
Industrial IoT edge
Levels 4-5
AWS Direct Connect
Site-to-Site VPN
Organization
OU – Infrastructure
Permissions
Roles
Network account
AWS Direct Connect
AWS Transit Gateway
AWS Cloud WAN
Inbound VPC
Might include:
• NAT
Internet gateway
Outbound VPC
Might include:
• NAT
• Proxy servers
Internet gateway
Inspection VPC
Might include:
• IDS/IPS
Firewall subnet
AWS Network Firewall
Flow logs
Flow logs
Amazon Route 53
Amazon CloudFront
AWS Verified Access
AWS Shield Advanced
AWS WAF
Amazon VPC Lattice
AWS Certificate Manager
AWS Resource Access Manager
Amazon GuardDuty
AWS Security Hub CSPM
Resolver DNS firewall
Network access analyzer
Organization trail
AWS Config
AWS IAM Access Analyzer
Industrial Isolation account
Permissions
Roles
VPC
Amazon AppStream 2.0
Remote third-party access
Patch management
Antivirus
AWS WAF
Amazon GuardDuty Malware Protection for S3
Amazon CloudFront
Amazon S3
Org Management account
OU – Security
Security Tooling account
AWS Security Hub CSPM
Log Archive account
Amazon Security Lake
OU – Workloads
Application account
Permissions
Roles
VPC
AWS IoT Core endpoint
AWS IoT Greengrass endpoint
AWS IoT SiteWise endpoint
Amazon S3 endpoint
AWS IoT Core
AWS IoT Greengrass
AWS IoT Device Management
Hosted zone
AWS IoT SiteWise
AWS IoT TwinMaker
AWS IoT Device Defender
Amazon GuardDuty
AWS Security Hub CSPM
AWS KMS
AWS Secrets Manager
Organization trail
AWS Config
IAM
AWS Private CA
AWS Signer
Partner IoT/IIoT/OT SaaS solution
Legend:
Existing SRA
IoT/OT SRA

The top part of the diagram represents the IIoT edge architecture. This is connected to the AWS Cloud organization in the lower part, which is constructed according to the AWS SRA. For a description of each account noted in the AWS organization in the lower part of the diagram, see .

Note that the isolation zone account is treated as an additional Shared Services account in the AWS SRA structure. This account is used to implement IoT-related networking and communications services, which are used by multiple workload accounts that also contain IoT-related processing. The isolation zone account can be considered a peer to the Networking account in the AWS SRA. It is used to manage shared networking and communications processes that are specific to the IIoT edge environments. In addition to the services shown in the diagram, the isolation zone account includes several common security services such as AWS Security Hub CSPM, Amazon GuardDuty, AWS Config, Amazon CloudWatch, and AWS CloudTrail.

For most customers, a single AWS organization with dedicated OUs for IoT, IIoT, and OT workloads is sufficient. You can separate the OT (or IIoT) environments from IT environments by using a isolation zone and the capabilities provided with AWS Organizations, multiple AWS accounts, VPCs, and networking configurations, as shown in the reference architecture.

Customer site and industrial edge

Customer site and industrial IoT edge refers to the specialized computing infrastructure deployed at industrial and OT environments to enable secure data collection, processing, and connectivity close to the source of data generation. This concept addresses the unique challenges of critical infrastructure environments and industrial settings, and supports distributed operations across multiple sites.

You can apply the Purdue model, which is a reference architecture model for the manufacturing industry, to implement different levels in the context of the customer site and industrial edge as follows:

- **Levels 0-2 – Field devices and local supervisory control:** Industrial equipment, sensors, and actuators are connected by using industrial protocol converters and data diodes. In certain cases, partner edge gateways that run AWS IoT SiteWise Edge are deployed to enable specialized local data acquisition and processing use cases at level 2.
- **Level 3 – Site operations:** Partner appliances and security sensors can be integrated to support asset discovery, vulnerability detection, and network security monitoring. Edge gateways based on AWS IoT Greengrass and AWS IoT SiteWise Edge are deployed to enable local data acquisition and processing.

- **Level 3.5 – Industrial isolation zone:** An industrial isolation zone represents a boundary between IT and OT, and controls the communication between the OT and the IT networks. Cloud access and internet access services such as proxies, firewalls, and unidirectional gateways are deployed to this layer to mediate the required connectivity and data flows.
- **Levels 4-5 – IT network:** Secure connectivity to the cloud is established by using Site-to-Site VPN or Direct Connect. AWS PrivateLink VPC endpoints are used for private access to AWS resources.

AWS organization

A Workloads OU for IoT, IIoT, or OT workloads is created alongside other workload-specific OUs. This OU is dedicated to applications that use relevant AWS IoT services to build and deploy IoT, IIoT, and OT-integrated solutions. The OU contains an Application account (shown in the previous architecture diagram) where you host your solution that provides the required business functionality. Grouping AWS services based on application type helps enforce security controls through OU-specific and AWS account-specific service control policies.

This approach also makes it easier to implement strong access control and least privilege. In addition to these specific OU and accounts, the reference architecture includes additional OUs and accounts that provide foundational security capabilities that apply to all application types. The Org Management, Security Tooling, Log Archive, and Network accounts are discussed in the *AWS SRA – core architecture* guide. These accounts have several additions that pertain to IoT workloads:

- **Network account** includes provisions for Direct Connect, Site-to-Site VPN, and AWS Transit Gateway. It also provides the possibility of creating a global network across operational assets by using AWS Cloud WAN, depending on the chosen approach for connecting to the AWS Cloud. For details, see the Infrastructure OU – Network account section of the in the *AWS SRA – core architecture* guide.
- **Industrial Isolation account** provides the option to deploy services (such as patching, antivirus, and remote access services) that would otherwise be deployed at the customer site or industrial IoT edge (level 3.5). This account supports scenarios that include robust connectivity between the site, the industrial IoT edge, and the AWS Cloud. These services are specific to servicing the IoT industrial edge and can be considered on the *edge side* instead of the *internet side* of a layered networking model.

Hosting services in the Industrial Isolation account on AWS provide enhanced flexibility, scalability, security, and integration capabilities compared with on-premises solutions, and enable more efficient and flexible management of industrial edge operations. For example, you can provide streaming access to your end-user applications by using Amazon WorkSpaces Applications and use Amazon GuardDuty Malware Protection for S3 to provide malware scanning capabilities as part of a secure file exchange solution that spans IT and OT environments. The Industrial Isolation account uses the shared connectivity constructs in the Network account, such as Transit Gateway, to obtain the required connectivity to the desired on-premises resources.

> **Note**
>
> This networking account is labeled *Industrial Isolation* because it serves as a buffer between the industrial IoT edge and the corporate networks that run within AWS accounts that are managed according to the AWS SRA. In this way, the account forms a type of edge between the industrial edge and corporate networking. This is similar to how the Network account in the AWS SRA serves as a buffer between the workloads running in the AWS Cloud (in workload accounts) and both the internet and corporate on-premises IT networks.

Partner IoT, IIoT, and OT SaaS solutions

AWS Partner solutions play a crucial role in helping enhance security monitoring and threat detection across IoT, IIoT, OT, and cloud environments. They complement the native IoT edge and cloud security services from AWS and help provide a more comprehensive security posture through a set of specialized detection and monitoring capabilities. The integration of these specialized OT and IIoT security monitoring capabilities with the broader cloud security offerings from AWS is achieved through services such as AWS Security Hub CSPM and Amazon Security Lake. You can deploy these solutions within your application accounts in your AWS organization. You can also use SaaS solutions that are hosted elsewhere on the internet and managed by third parties. In some cases, these third-party solutions also run on AWS. This scenario can facilitate IAM-based permissions management and AWS-specific network connectivity optimizations. In other cases, the connectivity to these services is configured according to the requirements of the SaaS solution.

These additions enable a more robust, secure, and flexible architecture that's specifically tailored for industrial environments and integrated with the AWS Cloud and AWS IoT services. The IoT components of the AWS SRA architecture address the unique challenges of industrial settings, such as protocol diversity, industrial edge processing requirements, and the need for seamless integration between OT and IT systems.

IoT security capabilities

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

This section discusses secure access, usage, and implementation recommendations for the IoT security capabilities discussed in the previous section.

Important

Use a common framework such as MITRE ATT&CK or ISA/IEC 62443 to conduct a cyber security risk assessment and use the outputs to inform the adoption of relevant capabilities. Your choice depends on your organization's familiarity with these frameworks and the expectations of your regulatory or compliance auditors.

Risk assessment guidance

Whether you're deploying consumer IoT devices, industrial IoT workloads, or operational technologies, you should first evaluate the risks and threats associated with your deployment. For example, one common threat to IoT devices listed in the MITRE ATT&CK framework is Network Denial of Service (T1498). The definition of a denial-of-service (DoS) attack against an IoT device is disallowing status or command and control communications to and from an IoT device and its controllers. In the case of a consumer IoT device, such as a smart bulb, the inability to communicate status or receive updates from a central control location could create problems but would likely not have critical consequences. However, in an OT and IIoT system that manages a water treatment facility, utility, or smart factory, losing the ability to receive commands to open or shut key valves could create a larger impact to operations, safety, and the environment. For this reason, consider the impact of various common threats, understand how they apply to your use cases, and determine ways to mitigate them. Key recommendations include:

- Identify, manage, and track gaps and vulnerabilities. Create and maintain an up-to-date threat model that you can monitor your systems against.
- Maintain an asset inventory of all connected assets and an up-to-date network architecture.

- Segment your systems based on their risk assessment. Some IoT and IT systems might share the same risks. In this scenario, use a predefined zoning model with appropriate controls between them.
- Follow a micro-segmentation approach to isolate the impact of an event.
- Use appropriate security mechanisms to control information flow between network segments.
- Understand the potential effects of indirect impact on communications channels. For example, if a communications channel is shared with some other workload, a DoS event on that other workload could affect the network communications of the IIoT or OT workload.
- Regularly identify and review security event minimization opportunities as your solution evolves.

In OT or IIoT environments, consider partitioning the system under consideration (SuC) into separate zones and conduits in accordance with ISA/IEC 62443-3-2, Security Risk Assessment for System Design. The intent is to identify assets that share common security characteristics in order to establish a set of common security requirements that reduce cybersecurity risk. Partitioning the SuC into zones and conduits can also help reduce overall risk by limiting the impact of a cyber incident. Zone and conduit diagrams can assist in detailed OT or IIoT cybersecurity risk assessments and help in identifying threats and vulnerabilities, determining consequences and risks, and providing remediations or control measures to safeguard assets from cyber events.

Recommended AWS services

When you build your environment in the AWS Cloud, use foundational services such as Amazon Virtual Private Cloud (Amazon VPC), VPC security groups, and network access control lists (network ACLs) to implement micro-segmentation. We recommend that you use multiple AWS accounts to help isolate IoT, IIoT, and OT applications, data, and business processes across your environment, and use AWS Organizations for better manageability and centralized insight.

For more information, see the Security Pillar – AWS Well-Architected Framework and the AWS whitepaper Organizing your AWS environment using multiple accounts.

Capability 1. Providing secure edge computing and connectivity

This capability supports best practices 3, 4, and 5 from the AWS SRA best practices for IoT.

The AWS shared responsibility model extends to the industrial IoT edge and into environments where devices are deployed. In the environments where devices are deployed, often called *IoT edge locations*, customers' responsibilities are much broader than they are in the cloud environment. Security of the IoT edge is the AWS customer's responsibility and includes securing the edge network, the edge network perimeter, and devices in the edge network; securely connecting to the cloud; handling software updates of edge equipment and devices; and edge network logging, monitoring, and auditing, as key examples. AWS is responsible for AWS-provided edge software such as AWS IoT Greengrass and AWS IoT SiteWise Edge, and AWS edge infrastructure such as AWS Outposts.

Rationale

As industrial operations increasingly adopt cloud technologies, there's a growing need to bridge the gap between traditional OT systems and modern IT infrastructure. This capability addresses the necessity for secure, low-latency processing at the edge while also ensuring robust connectivity to AWS Cloud resources. By implementing edge gateways and secure connectivity methods, organizations can maintain the performance and reliability required for critical industrial processes while they take advantage of the scalability and advanced analytics capabilities of cloud services.

This capability is also essential for maintaining a strong security posture in IIoT and OT environments. OT systems often involve legacy devices and protocols that might lack built-in security features and become vulnerable to cyber threats. By incorporating secure edge computing and connectivity solutions, organizations can implement crucial security measures such as network segmentation, protocol conversion, and secure tunneling closer to the data source. This approach helps protect sensitive industrial data and systems and also enables compliance with industry-specific security standards and regulations. Additionally, it provides a framework for securely managing and updating edge devices, which further enhances the overall security and reliability of IIoT and OT deployments.

Security considerations

The implementation of secure edge computing and connectivity in IoT, IIoT, and OT solutions presents a multifaceted risk landscape. Key threats include inadequate network segmentation between IT and OT systems, security weaknesses in legacy industrial protocols, and the inherent limitations of edge devices that have limited resources. These factors create potential entry points and avenues for threat propagation. The transmission of sensitive industrial data between edge devices and cloud services can also introduce risks of interception and manipulation, and insecure cloud connections can expose systems to internet-based threats. Additional concerns include

the potential for lateral movement within industrial networks, lack of visibility into edge device activities, physical security risks for remotely located infrastructure, and supply chain vulnerabilities that can introduce compromised components. Collectively, these threats underscore the critical need for robust security measures in edge computing and connectivity solutions for industrial environments.

Remediations

Data protection

To address data protection concerns, implement encryption for data in transit and at rest. Use secure protocols such as MQTT over TLS, HTTPS, and WebSockets over HTTPS. For communications with IoT devices, and generally within IoT industrial edge environments, consider using secure versions of industrial protocols such as CIP Security, Modbus Secure, and Open Platform Communications Unified Architecture (OPC UA) with security mode enabled. When secure protocols aren't natively supported, employ protocol converters or gateways to translate insecure protocols into secure ones as close to the data source as possible. For critical systems that require strict data flow control, consider implementing unidirectional gateways or data diodes. Use AWS IoT SiteWise Edge gateways with OPC UA security mode for industrial data sources, and use AWS IoT Greengrass for secure local MQTT broker configurations. When protocol-level security isn't possible, consider implementing an encryption overlay by using VPNs or other tunneling technologies to protect data in transit.

In the context of the AWS SRA for IoT, IIoT, and OT environments, secure protocol usage and conversion should be implemented at multiple levels:

- Level 1. By using an AWS IoT SiteWise Edge gateway connected to an industrial data source that supports OPC UA with security mode.
- Level 2. By using an AWS IoT SiteWise Edge gateway combined with a partner data source that supports legacy protocols to achieve required protocol conversion.
- Level 3. By using a secure local MQTT broker configuration with MQTT brokers that are supported through AWS IoT Greengrass.

Identity and access management

Implement robust identity and access management practices to mitigate unauthorized access risks. Use strong authentication methods, including multi-factor authentication where possible, and

apply the principle of least privilege. For edge device management, use AWS Systems Manager for secure access and configuration of edge computing resources. Use AWS IoT Device Management and AWS IoT Greengrass for secure management of IoT devices. When you use AWS IoT SiteWise gateways, employ AWS OpsHub for secure management. For edge infrastructure, consider AWS Outposts as a fully managed service that consistently applies best practices to AWS resources at the edge.

Network security

Secure connectivity between the industrial edge and the AWS Cloud is a critical component for the successful deployment of IoT, IIoT, and OT workloads in the cloud. As shown in the AWS SRA, AWS offers multiple ways and design patterns to establish a secure connection to the AWS environment from the industrial edge.

The connection can be achieved in one of three ways:

- By setting up a secure VPN connection toAWS over the internet
- By establishing a dedicated private connection through AWS Direct Connect
- By using secure TLS connections to AWS IoT public endpoints

These options provide a reliable and encrypted communication channel between the industrial edge and the AWS infrastructure, in alignment with the security guidelines outlined in the National Institute of Standards and Technology (NIST) Guide to Operational Technology (OT) Security (NIST SP 800-82 Rev. 3) which warrants the need to "use secure connections … between network segments, such as between a regional center and primary control centers and between remote station and control centers."

After you establish a secure connection to workloads running in AWS and to AWS services, use virtual private cloud (VPC) endpoints whenever possible. VPC endpoints enable you to connect privately to supported Regional AWS services without using the public IP addresses of these AWS services. This approach further helps enhance security by establishing private connections between your VPC and AWS services, and aligns with NIST SP 800-82 Rev. 3 recommendations for secure data transmissions and network segmentation.

You can configure VPC endpoint policies to control and limit access to only the required resources, applying the principle of least privilege. This helps reduce the attack surface and minimize the risk of unauthorized access to sensitive IoT, IIoT, and OT workloads. If the VPC endpoint for the required service isn't available, you could establish a secure connection by using TLS over the public

internet. The best practice in such scenarios is to route these connections through a TLS proxy and a firewall, as shown previously in the Infrastructure OU – Network account section of the *AWS SRA – core architecture* guide.

Some environments might have requirements to send data in one direction to AWS while physically blocking traffic in the opposite direction. If your environment has this requirement, you can use data diodes and unidirectional gateways. Unidirectional gateways consist of a combination of hardware and software. The gateway is physically able to send data in only one direction, so there is no possibility of IT-based or internet-based security events pivoting into the OT networks. Unidirectional gateways can be a secure alternative to firewalls. They meet several industrial security standards, such as the North American Electric Reliability Corporation Critical Infrastructure Protection (NERC CIP), the International Society of Automation and International Electrotechnical Commission (ISA/IEC) 62443, the Nuclear Energy Institute (NEI) 08-09, the U.S. Nuclear Regulatory Commission (NRC) 5.71, and CLC/TS 50701. They are also supported by the Industry IoT Consortium's Industrial Internet Security Framework, which provides guidance on protecting safety networks and control networks with unidirectional gateway technology. NIST SP 800-82 states that using unidirectional gateways might provide additional protections associated with system compromises at higher levels or tiers within the environment. This solution enables regulated industries and critical infrastructure sectors to take advantage of cloud services on AWS (such as IoT and AI/ML services) while preventing remote events from penetrating back into protected industrial networks. OT devices that are behind the data diode and unidirectional gateway need to be locally managed. The data diode function is a networking-related function. The data diodes and unidirectional gateways, when deployed into the AWS environment to support the IoT industrial edge, should be deployed into the Industrial Isolation networking account so they are embedded between levels in the OT network.

Capability 2. Providing an industrial isolation zone between environments

This capability supports best practice 2 from the AWS SRA best practices for IoT.

Organizations are increasingly connecting OT and IIoT systems to cloud environments. This convergence brings numerous benefits but also introduces unique security challenges. It also requires strict separation between OT, IIoT, and IT environments to limit the potential for attacks to OT or IT systems from affecting business systems for critical infrastructure. A single AWS organization that includes multiple AWS accounts can meet the requirements for implementing this strict separation by using an Industrial Isolation account and separate OUs, separate AWS

accounts, and careful configuration of networking between accounts (separate VPCs, AWS Transit Gateway routing, and network inspection firewalls). This approach provides a secure foundation for integrating industrial systems with cloud services while maintaining the strict security and operational requirements that are inherent to OT environments. By implementing this capability, organizations can take advantage of the scalability and advanced services provided by AWS while preserving the integrity, availability, and security of their critical industrial operations.

Rationale

Establishing a separate OU within the AWS organization that is dedicated to IoT, IIoT, and cloud-connected OT workloads helps enhance security by enabling segregation from traditional IT environments. This approach allows organizations to:

- Directly apply OT security principles and standards to the AWS environment.
- Accommodate different risk toleration between OT and IT teams.
- Limit potential impact of security incidents.
- Enable clear separation of duties between OT and IT personnel.

When you use a dedicated OU for IoT, IIoT, and OT along with segregated networking by using separate VPC configurations to connect VPCs that span multiple accounts, the OU should have the following characteristics:

- Segregated network architectures should be provided for both the IoT (or OT or IIoT) and the industrial isolation workloads.
- The OT or IIoT environment within the landing zone should be designed to align with the security requirements that are outlined in ISA/IEC 62443 and NIST SP 800-82 for industrial control systems and operational technology.
- The Industrial Isolation account should act as a dedicated security perimeter between the OT (or IIoT) environment and the IT environment, and should follow the NIST SP 800-82 guidance on network segmentation and the use of demilitarized zones.
- The landing zone should have segregated identities or roles, defined within the identity infrastructure, which are separate from IT identities or roles. You can implement these as separate identity center assignments within the AWS IAM Identity Center instance for the AWS organization, to manage access and permissions for the OT (or IIoT) and Industrial Isolation account resources in parallel with the IT environment.

- The identity and access management policies in the landing zone should be tailored to the unique needs and risk profiles of the OT, IIoT, and industrial isolation components, which might differ from traditional IT environments.
- The OU should also host services and resources that facilitate secure communication, remote access, and data exchange between the OT (or IIoT) and IT domains, while maintaining strict access controls and monitoring mechanisms.

This separation also creates the opportunity for further enhancements to the security posture of these workloads, by integrating relevant IIoT services and features that are available on AWS, such as AWS IoT Core, AWS IoT Greengrass, AWS IoT Device Defender, AWS IoT Device Management, AWS IoT SiteWise, and AWS IoT TwinMaker. These services help provide secure connectivity, data management, and analytics capabilities that are tailored for the OT and IIoT environments.

For example, the ISA/IEC 62443 standard defines the security requirements for industrial automation and control systems, and NIST SP 800-82 provides guidance on securing industrial control systems, including recommendations for network architecture, remote access, and patch management. By aligning the design and configuration of the dedicated OT portions of the organization with the ISA/IEC 62443 standards and the NIST SP 800-82 guide, organizations can ensure that security controls such as network segmentation, access management, and device hardening are implemented consistently across all components of their AWS landing zone. This can help organizations bridge the gap between traditional IT security and the specific requirements of cloud-connected OT and IIoT systems.

Additional benefits include:

- **Isolation of OT and IT workloads**: Separate OUs, AWS accounts, and networking configurations allow for better isolation of OT and IT workloads, and ensure that the security, access controls, and resource configurations can be tailored to the specific requirements of each domain. This helps mitigate the risk of cross-contamination, reduces the scope of impact, and ensures that the unique needs of OT and IT systems are addressed.
- **Tailored configurations**: By using distinct OUs, AWS accounts, and networking configurations, you can configure each environment independently to meet the specific technical requirements of your OT and IT teams. This includes the ability to apply different security controls, such as network ACLs, security groups, and IAM policies, as well as resource-level configurations such as instance types, storage options, and backup/restore mechanisms.
- **Simplified governance and compliance for showing segregation of duties (SoD)**: Maintaining separate OUs, AWS accounts, and networking configurations simplifies the application of

different compliance frameworks, security standards, and regulatory requirements to the OT, IIoT, and IT environments. For OT and IIoT systems, this might include compliance with standards such as ISA/IEC 62443 and NIST SP 800-82, which have specific requirements for secure OT and IIoT system design, deployment, and maintenance. In contrast, the IT systems might have to comply with standards such as ISO 27001 and Payment Card Industry Data Security Standard (PCI DSS).

- **Scalability and flexibility**: Independent OUs, AWS accounts, and networking configurations provide the ability to scale each environment as needed, without the risk of unintended impacts on the other domain. This allows for more efficient resource allocation, testing processes, and deployment processes that are tailored to the specific requirements of the OT (or IIoT) and IT teams.
- **Reduced complexity**: Separating the OT and IT environments into distinct OUs, AWS accounts, and networking configurations helps reduce the overall complexity of the AWS infrastructure, and makes it easier to manage, monitor, and troubleshoot each domain independently. This can lead to improved operational efficiency and reduced risk of cross-domain issues.
- **Specialized tooling and processes**: The OT (or IIoT) and IT teams might require different tools, automation scripts, and operational processes to effectively manage their respective environments. Separate OUs, AWS accounts, and networking configurations enable the implementation of specialized tooling and workflows that are optimized for the unique needs of each domain. For example, OT or IIoT teams might require specific industrial control system (ICS) monitoring and management tools whereas IT teams focus on traditional IT management platforms.
- **Improved disaster recovery and business continuity**: Maintaining separate OUs, AWS accounts, and networking configurations enhances your organization's ability to ensure business continuity and effective disaster recovery. This is particularly important for OT and IIoT systems, which might have stricter uptime and availability requirements compared with IT systems.

Security considerations

The integration of OT or IIoT systems with cloud environments introduces potential security risks that this capability aims to address. Primarily, it mitigates the threat of lateral movement between IT and OT networks, which could lead to a potential compromise of industrial control systems and other significant OT workloads. Without proper segmentation, a threat actor with malicious intent who gains unauthorized access to the IT network could potentially pivot to the OT network and

gain unauthorized access to critical OT systems, which might lead to safety incidents, production downtime, or environmental damage.

Additionally, this capability addresses the risks associated with the unique operational requirements and legacy protocols often found in OT environments. Many industrial systems use proprietary or outdated protocols that lack built-in security features, which make them vulnerable to interception, manipulation, and exploitation when exposed to broader networks. By providing separate OUs, AWS accounts, networking configurations, and an Industrial Isolation account, organizations can implement appropriate protocol conversions, access controls, and monitoring solutions that are specifically tailored to these OT and IIoT communications, to reduce the attack surface and the potential for unauthorized access or data exfiltration.

Remediations

Data protection

Latency-sensitive industrial processes and real-time control systems might struggle with the higher network latency inherent in a cloud-based architecture, especially when connecting OT or IIoT equipment over a wide-area network to a remote AWS Region. Additionally, many industrial protocols used in OT environments, such as Modbus, Distributed Network Protocol 3 (DNP3), and proprietary SCADA protocols, were not designed with cloud connectivity in mind. Transmitting these insecure and often unencrypted traffic over public networks introduces a significant risk of interception, tampering, and exploitation. To mitigate these concerns, implement secure protocol conversion for legacy industrial communications before transmission over wide-area networks. Deploy specialized OT and IIoT network traffic monitoring and threat detection solutions in both on-premises and cloud environments to identify and respond to potential data breaches or unauthorized access attempts. Regularly review and update data protection measures to maintain alignment with evolving OT and IIoT security standards and best practices.

Identity and access management

Establish dedicated IAM Identity Center permission sets and identity center assignments for OT or IIoT access management that are separate from IT systems. Check for strict separation of concerns or duties in the IAM Identity Center assignments. Configure IAM policies that are specific to OT or IIoT requirements and ensure that the principle of least privilege is applied. Implement strong authentication mechanisms, such as multi-factor authentication, for accessing OT or IIoT resources in the cloud. Regularly audit and review access permissions to maintain a secure posture.

Network security

Design the OT or IIoT network architecture to align with NIST SP 800-82 guidance on segmentation and industrial isolation implementation. Configure security groups and network ACLs to enforce strict traffic control between OT (or IIoT), industrial isolation, and IT networks. Implement AWS IoT security services, such as AWS IoT Device Defender, to enhance the protection of connected industrial assets. Establish secure VPN or AWS Direct Connect links for communication between on-premises OT networks and the AWS Cloud. Regularly conduct network security assessments and penetration testing to identify and address potential vulnerabilities in the OT or IIoT network architecture.

> **Note**
>
> In some situations, such as those that involve critical infrastructure or highly regulated or segregated OT environments, or cases where there are requirements for strict separation between OT and IT teams with no common chains of command, you can deploy a separate AWS organization with a landing zone for IoT, IIoT, or OT workloads. In this deployment model, you can configure selective network connectivity between the two separate AWS organizations. However, this model duplicates effort in identity and access management, organization management, security configuration, and logging and monitoring activities, and should be considered only if you can't meet the requirements by using a single AWS organization with separate or dedicated OUs for IoT, IIoT, or OT workloads.

Capability 3. Providing strong device identities and secure device access and management

This capability supports best practices 6 and 7 from the AWS SRA best practices for IoT.

In the rapidly evolving landscape of IoT, IIoT, and OT, ensuring the security and integrity of connected devices is paramount. This capability focuses on implementing robust device identity lifecycle management and secure update mechanisms. It is crucial for maintaining the trustworthiness of devices throughout their operational lifespan, from initial deployment to retirement, while ensuring that they remain current with the latest security patches and firmware updates.

Rationale

Devices that form part of IoT, IIoT, and cloud-connected OT solutions continuously interact with one another and with cloud services to exchange data, and, in some cases, to facilitate critical processes. The security of these devices is not just a technical requirement but a core business imperative. Strong device identities form the foundation of this security framework and enable reliable authentication and authorization. Devices, ranging from factory floor sensors to smart grid gateways, must conclusively establish their authenticity when they access on-premises data sources, network resources, or cloud services. This establishment of trust is essential to help prevent unauthorized access and potential compromises that could result in operational disruptions or data breaches.

The dynamic nature of IoT and IIoT environments also necessitates an active approach to device management. Devices require regular updates with the latest security patches and firmware to address newly discovered vulnerabilities and to enhance functionality. A comprehensive identity and management system facilitates the secure and timely distribution of these updates across device fleets. Additionally, it enables fine-grained access control and ensures that each device operates under the principle of least privilege to access only the resources that are necessary for its designated function. This system manages the entire lifecycle of device identities, from initial provisioning through potential repurposing or recommissioning, to eventual decommissioning.

Security considerations

The implementation of strong device identities and secure management practices addresses several critical security risks. Device impersonation poses a significant threat, because attackers can potentially gain unauthorized access to sensitive systems by mimicking legitimate devices. This risk is compounded by weak authentication mechanisms and overly permissive access controls, which can lead to unauthorized access to devices and associated cloud resources.

Outdated software and firmware present another substantial challenge. Unpatched devices remain susceptible to known security flaws and create potential entry points for malicious actors. The update process introduces additional risks, because insecure update mechanisms can be used for supply chain attacks and enable the distribution of malicious code across device fleets. Furthermore, inadequate protection of device credentials, including cryptographic keys and certificates, can result in widespread system compromise if these credentials are obtained by unauthorized parties. The implementation of this capability helps mitigate these risks by establishing a robust framework for device authentication, authorization, and lifecycle management.

Remediations

Data protection

Implement cryptographic signing and verification for all software and firmware updates to help ensure authenticity and integrity. Use AWS Signer for code signing capabilities to help ensure the trust and integrity of code that's created for IoT devices. Store updates securely by using Amazon Simple Storage Service (Amazon S3) with appropriate permissions, access roles, and encryption settings, such as server-side encryption by using AWS managed keys or customer managed keys. Implement version control and rollback capabilities by using AWS IoT Jobs and AWS IoT Device Management Software Package Catalog to maintain version history and to revert to previous versions if necessary.

Develop and implement a robust update strategy that includes gradual rollouts to catch defects and to ensure that all devices of the same type aren't affected simultaneously. Design the update process to be responsive to vulnerabilities and to be scalable for managing updates across large fleets of diverse devices. Use AWS IoT Jobs and AWS IoT Device Management for scalable and secure distribution of updates. Implement monitoring and logging of update processes to detect anomalies and maintain audit trails. Make sure that update mechanisms are resilient to intermittent connectivity and resource constraints that are common in IoT environments. Consider implementing cancel, rollback or fallback, and failed update handling procedures.

Identity and access management

Provision devices that have unique identities by using X.509 certificates or other strong credentials. Implement a comprehensive device identity lifecycle management system that covers provisioning, rotation, and revocation of credentials. Use the security features in AWS IoT Core for device authentication and authorization. Use AWS Private Certificate Authorityto provision and manage device certificates. Use AWS Certificate Manager (ACM) to manage server keys or certificates for applications. Employ Amazon Cognito to manage user identities that are associated with device management interfaces. Use AWS Secrets Manager to securely store and manage device secrets, and encrypt them by using AWS Key Management Service (AWS KMS). Implement hardware-protected modules such as Trusted Platform Modules (TPMs), where available, to establish a root of trust on devices.

Network security

Use secure communication protocols such as MQTT over TLS for device-to-cloud communications. Where possible, implement AWS PrivateLink VPC endpoints for secure configuration management

and update downloads. Apply network segmentation to isolate IoT and IIoT devices from other critical network assets. Use AWS IoT Device Defender to continuously audit and monitor the security posture of your device fleet, including checking for compliance with security best practices such as the principle of least privilege and unique identity per device.

Capability 4. Providing data protection and governance

This capability supports best practice 8 from the AWS SRA best practices for IoT.

Capability 4 addresses the critical need to secure IoT and IIoT data throughout its entire lifecycle, from edge devices to cloud storage and processing systems. It encompasses robust encryption mechanisms for both data at rest and data in transit as well as establishing thorough data governance practices.

Rationale

Industrial systems can generate, process, and store vast amounts of sensitive information, including proprietary manufacturing processes, equipment performance data, and critical operational telemetry. Unauthorized access to, or manipulation of, this data can result in significant consequences that range from intellectual property theft to operational disruptions and safety incidents. Implementing robust encryption and data governance practices addresses these risks directly. It helps safeguard valuable information assets and helps ensure the continuity of industrial operations.

Security considerations

The implementation of robust data protection and governance measures addresses several security risks in IoT, IIoT, and OT environments. Primary concerns include unauthorized access to sensitive data that's stored on IoT devices and edge gateways, and the interception of data during transmission between devices and cloud systems.

Remediations

Data protection

Data at rest encryption: Information that's stored on deployed devices such as sensors or cameras might seem harmless, but when the physical control of a device isn't guaranteed, that information

can be a target for unauthorized actors. Examples include cached videos on consumer cameras, proprietary machine learning (ML) models in industrial applications, and configuration data for operational environments. For deployed devices, the best practice is to encrypt all data that's stored at rest when possible. This includes:

- Device storage: Encrypt local storage on IoT devices by using hardware-based encryption (when available) or strong software encryption.
- Edge gateways: Implement full-disk encryption on edge gateways and local servers.
- Cloud storage: Use AWS-managed encryption services for data that's stored in the cloud, as described in the AWS KMS section in the Application account of the *AWS SRA – core architecture*.

Implement mechanisms for clearing information that's stored in devices. This might be necessary when devices are repurposed or sold and change ownership.

Data in transit encryption: Encrypt all data in transit, including sensor and device, administration, provisioning, and deployment data. Nearly all modern IoT devices have the capacity to perform encryption of network traffic, so take advantage of that ability and protect both data plane and control plane communications. This practice helps ensure both the confidentiality of the data and the integrity of monitoring signals. For protocols that can't be encrypted, consider whether an edge device that's closer to the IoT asset can accept the communication and convert it to a secure protocol before sending it outside the local perimeter.

Key practices include:

- Use TLS for all MQTT and HTTP communications (that is, use MQTTS and HTTPS). Secure communications are recommended regardless of the network packet routing path, whether it's confined to the AWS backbone or not.
- Implement secure MQTT for IoT messaging, including at the edge.
- Use AWS Site-to-Site VPN, AWS PrivateLink, and AWS Direct Connect for secure communication between on-premises components and AWS. These services provide more predictable network routing or packet encapsulation compared with internet-accessible API endpoints.

Capability 5. Providing security monitoring and incident response

This capability supports best practices 9 and 10 from the AWS SRA best practices for IoT.

Capability 5 focuses on implementing comprehensive security monitoring and incident response mechanisms across IoT, IIoT, OT, edge, and cloud environments. This capability encompasses the deployment of logging and monitoring mechanisms, centralized management of security alerts, and the creation of incident response playbooks and business continuity plans that are tailored to the unique challenges of hybrid OT and IT architectures.

Rationale

The integration of OT, IoT, and IIoT technologies with traditional IT systems and cloud services introduces new attack vectors and expands the overall cyber attack surface. Security events can originate in OT environments and propagate to IT systems, or they can originate in IT systems and propagate to OT environments. This makes it critical to implement comprehensive security monitoring across the full attack surface. Implementing this capability enables organizations to:

- Establish a unified view of security across OT, IoT, IIoT, edge, and cloud environments.
- Detect and respond to security anomalies and threats in real time.
- Maintain operational continuity in the face of cyber incidents.
- Enhance overall cybersecurity resilience and reduce the potential impact of security breaches.

Moreover, the development of incident response playbooks and business continuity plans that are specifically tailored to cloud-connected OT and IIoT workloads ensures that organizations can effectively manage and recover from security incidents. This proactive approach minimizes downtime, helps protect against financial losses, and safeguards an organization's reputation in the event of a security breach or operational disruption.

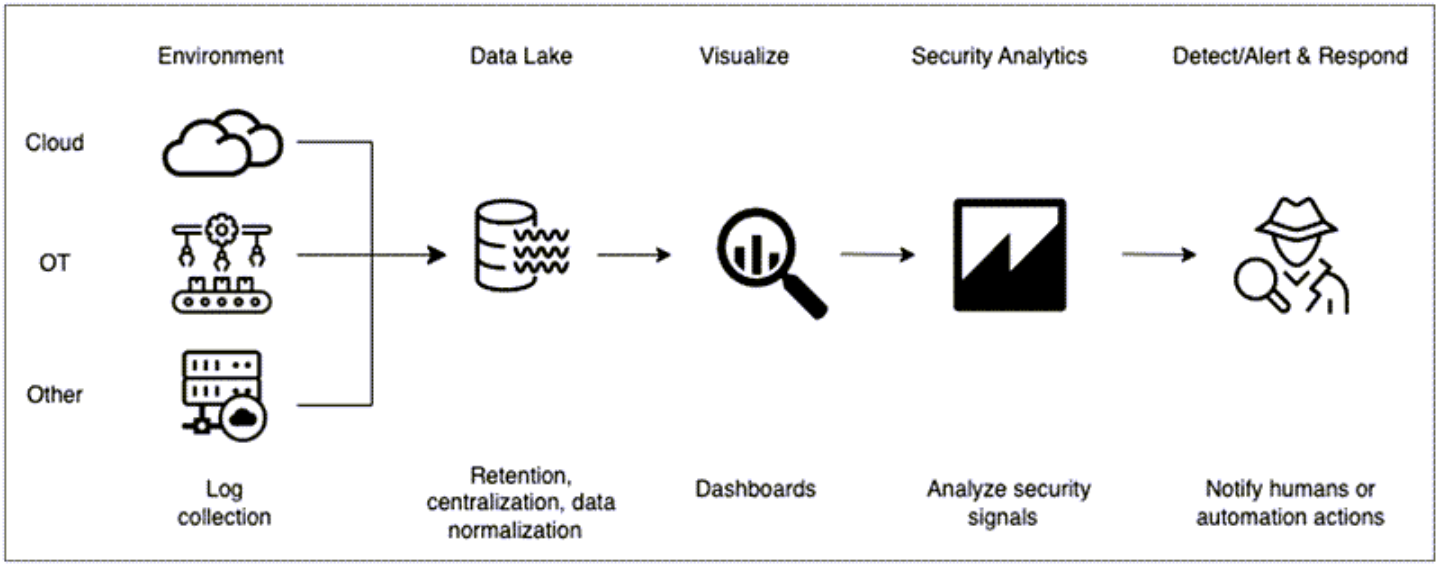
Security considerations

The primary consideration addressed by this capability is the risk of delayed detection of security incidents due to siloed monitoring of OT and IT environments. This might be compounded by the inability to correlate security events across these diverse technology stacks. This fragmentation often results in insufficient visibility into industrial network traffic and anomalies, and leaves critical systems exposed to undetected events. Furthermore, the interconnected nature of modern industrial systems creates the potential for cascading failures, where a security event in one area can rapidly propagate across interconnected OT and IT systems, and can amplify the impact of an incident.

Another significant concern is the incompatibility of traditional response procedures when dealing with hybrid OT/IT security incidents, which require specialized knowledge and coordinated action

across multiple domains. This is particularly critical given the increasing threat of cyberphysical events that target industrial processes. Additionally, the unique nature of interconnected OT and IIoT systems often means that recovery mechanisms after a security incident might be insufficient and might potentially lead to prolonged downtime and operational disruptions.

The following illustration shows a unified System and Organization Controls (SOC) architecture for IT and OT systems.



Remediations

Security logging and monitoring

Use centralized AWS Security Hub CSPM and Amazon Security Lake services to capture and handle events that are relevant to IoT, IIoT, and cloud-connected OT solutions in combination with the rest of your AWS organization. Use separate concerns, responsibilities, IAM permission sets, and identity center assignments to identify the teams that can change the configurations for the AWS accounts that are dedicated to OT, IIoT, and Industrial Isolation account resources. All security events can be sent to Security Hub CSPM to gain a centralized view of security findings across your OT, IoT, IIoT, edge, and cloud environments. Review the logging and monitoring recommendations in the Log Archive account section of the *AWS SRA – core architecture*.

Implement a unified SOC by integrating IT and OT security data in Security Lake, which can provide broad visibility across the IT and OT environments and enable coordinated threat detection, faster incident response, and immediate sharing of indicators of compromise (IoCs) between environments. This allows for better understanding of threat paths and origins across OT, IoT, IIoT, edge, and cloud environments. The Partner IoT, IIoT, and OT SaaS solutions section shows how OT

and IIoT security monitoring solutions from AWS Partner Network (APN) providers and others can be used to complement the IoT edge and cloud security services provided by AWS.

Incident response

Begin by identifying potential incident scenarios that are specific to your deployment, such as IoT device or edge gateway compromise, operational data breaches, or disruptions to industrial processes. For each scenario, create detailed response procedures (playbooks) that outline steps for detection, containment, eradication, and recovery. These playbooks should clearly define roles and responsibilities, communication protocols, and escalation procedures. Test these playbooks by using tabletop exercises. These exercises test the procedures and educate the teams that will have to implement the procedures under the pressure of an actual ongoing incident.

Implement continuous health checks and monitoring systems to detect anomalies before they escalate into major incidents. Automate initial response actions where possible to contain events quickly and to return systems to a known good state. As your IoT environment matures, regularly review and update these playbooks to address new threats and incorporate lessons learned from previous incidents or simulations.

For business continuity and disaster recovery, define clear parameters for system behavior during failures or disruptions. Determine whether systems should fail open or closed, if recovery should be automatic or require human intervention, and the conditions under which manual controls should be enabled or disabled. These decisions should be based on the criticality of the systems and potential impact on safety, operations, and the environment. Test your continuity and recovery plans to ensure that they perform as expected under various scenarios.

Contributors

The following individuals contributed to this guide.

Authoring:

- Avik Mukherjee, AWS Senior Security SA
- Ryan Dsouza, AWS Principal Guidance Lead SA
- Tim Hahn, AWS Senior Delivery Consultant

Reviewing:

- Eric Rose, AWS Principal Security SA
- Manoj Kumar, AWS Delivery Consultant

Technical writing:

- Handan Selamoglu, AWS Senior Technical Writer

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 as standalone guide	Converted from a chapter in the AWS SRA – core architecture guide to an individual guide.	December 22, 2025

AWS Prescriptive Guidance glossary

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

Numbers

7 Rs

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

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.

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

A

A2A (Agent-to-Agent)

A stateful protocol for agent-to-agent collaboration supporting task delegation and state transfer.

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 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.

Agent

An AI system that can autonomously reason, plan, and take actions using tools to achieve goals.

Agent Ops

Operational practices for building, testing, deploying, and running AI agents in production at scale.

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.

Citizen Developer

A business user who creates AI applications using no-code/low-code platforms without specialized technical skills.

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 Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

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, Amazon SageMaker AI 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.

EDI

See electronic data interchange.

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.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

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.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

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.

FM

See foundation model.

foundation model (FM)

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

FM gateway

A centralized intermediary that controls and normalizes access to foundation models. Also known as an *LLM gateway*.

G

generative AI

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

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.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision

software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

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 CSPM, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

guardrails (AI)

Safety mechanisms that filter, validate, and constrain agent inputs and outputs to help ensure responsible and safe AI behavior.

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.

holdout data

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

human-in-the-loop (HitL)

A workflow pattern where agent execution pauses for human review and approval at critical decision points.

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 language model (LLM)

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

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.

LLM

See large language model.

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.

MCP

See Model Context Protocol.

Model Context Protocol (MCP)

A stateless protocol for agent-to-tool communication.

MCP server

A service that exposes one or more tools through the Model Context Protocol.

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.

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

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

prompt chaining

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

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).

RAG

See Retrieval Augmented Generation.

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.

Retrieval Augmented Generation (RAG)

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

rotation

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

row and column access control (RCAC)

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

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

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

S

SAML 2.0

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

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

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

security by design

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

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.

Shadow AI

Unauthorized AI applications built or used outside of governed channels within an organization.

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.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

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.

tool

A function or API that an agent can invoke to perform operations in external systems.

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.

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.

zero-shot prompting

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

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.