Logging and monitoring guide for application owners

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Logging and monitoring guide for application owners

Table of Contents

Logging and monitoring guide for application owners

John Buckley, Amazon Web Services (AWS)

January 2023 (document history)

A *workload* is a collection of resources and code that delivers business value, such as a customer-facing application or a backend process. A workload might consist of a subset of resources in a single AWS account, or it might span multiple AWS accounts. In the cloud, an *application* is a type of workload. It might be deployed exclusively in the cloud environment, or it might be also be supported by local, on-premises hardware. Many publications focus on logging and monitoring cloud infrastructure and are intended for security teams. This guide is intended for application owners and focuses on effective and efficient approaches for logging and monitoring applications in the AWS Cloud.

This guide helps you set logging and monitoring at an appropriate level so that you can identify and respond to anomalies quickly. It also helps you make sure that your application logs support detailed analysis and resolution of any issues.

Although this guide is written with AWS Cloud deployments in mind, you can apply these principles to applications running on premises or on other cloud provider infrastructure.

Targeted business outcomes

After reading this guide, you should be able to understand:

- The types of events that are commonly logged for applications
- The event attributes (such as the who, what, and when) that you should consider logging
- The types of data that you should consider excluding from logs, such as data that might compromise your security posture or personally identifiable information
- How to set logging and monitoring at an appropriate level for your application
- Who should be able to manage and access your application logs
- The AWS services and features that you can configure to monitor and log your applications in the AWS Cloud
- How to use the log data from your application and AWS services and features to triage problems and diagnose issues

About logging and monitoring for applications

Logging, monitoring, alerting, and reporting are different security processes that work together to provide visibility into the health and performance of your application. It is critical that you create and maintain a detailed record of actions and events for your application so that you can monitor, alert, and report based on the recorded activity.

Application logging is the process of collecting the events generated by your application and recording them in one or more log files. This history of events can help you perform security and performance analyses, track resource changes, and troubleshoot application issues.

Application monitoring is the process of assessing the overall performance and health of your application. You should be able to monitor the application's frontend and backend constantly. Because applications hosted on the cloud are highly distributed, logging and monitoring tools can help you quickly troubleshoot performance issues or identify and remediate security threats in real time. Log data is a critical input for monitoring.

Observability is similar to monitoring, but it introduces ways to measure application behavior using different parameters, and it allows for complex correlations. An example is measuring the HTTP success rate on a particular day, for a set of users in a specific geographical region. For more information, see Monitoring and Observability (AWS marketing).

Ultimately, the goal of application owners is to maintain secure, healthy applications and positive user experiences with those applications. By implementing logging and monitoring, your developers and operations teams can more quickly plan for and troubleshoot application issues.

The level of logging and monitoring required varies for each application. Factors that can affect the monitoring and logging levels include organizational policies and procedures, level of security risk the application poses, the criticality of the application to business operations, and the sensitivity of the data managed by the application. In general, applications that are public or customer-facing require a higher level of monitoring and logging than applications that are used internally in the organization. This guide includes general information and recommendations, and you should customize your approach based on the requirements of your application.

Note

The standards or procedures in your organization might mandate specific logging and monitoring attributes. An example is passing user permissions into an enterprise

entitlement review system. Make sure that your logging and monitoring plan addresses the requirements of your organization.

Logging for applications in the AWS Cloud

For logging applications in the AWS Cloud, review common types of events, the event attributes, and best practices.

This section includes the following topics:

Event types

One of the most important considerations when establishing an application logging strategy is deciding which events and actions to log. Although the requirements of your organization and application might affect this decision, we recommend that you always log the following if they apply to your application:

- **Input validation failures** – Examples include protocol violations, unacceptable encodings, and invalid parameter names and values.
- **Output validation failures** – Examples include database record set mismatches and invalid data encoding.
- **Identity authentication successes and failures** – Log authentication activities, but do not log user names and passwords. Because users can accidentally type their passwords into a user name field, we recommend that you don't log user names. This might unintentionally expose credentials and result in authorized access. Implement security controls for any logs that contain authentication data.
- **Authorization (access control) failures** – For related authorization systems, log failed access attempts. You can monitor this log data for patterns that might indicate an attack or issues with the authorization system in the application.
- **Session management failures** – Examples include modifying session cookies or tokens. Applications often use cookies or tokens to manage user states. Malicious users can attempt to modify cookie values to gain unauthorized access. Logging tampered session tokens provides a way to detect this behavior.

- **Application errors and system events** – Examples include syntax and runtime errors, connectivity problems, performance issues, error messages from third-party services, file system errors, virus detection for file uploads, and configuration changes.
- **Application state** – Starting or stopping the application and its related resources.
- **Logging state** – Starting, stopping, or pausing logging.
- **Use of higher-risk functionality** – Examples include network connection changes, adding or deleting users, changing privileges, assigning users to tokens, adding or deleting tokens, using system administrative privileges, access by application administrators, all actions performed by users with administrative privileges, accessing payment cardholder data, using data encryption keys, changing encryption keys, creating and deleting of system-level objects, submitting user-generated content (especially file uploads), and importing and exporting data (including reports).
- **Legal and other opt-ins** – Examples include permissions for mobile phone capabilities, terms of use, terms and conditions, personal data usage consent, and permissions to receive marketing communications.

In addition to the recommended attributes, for your application, consider what additional attributes might provide useful data for monitoring, alerting, and reporting. Examples include:

- Sequencing failures
- Attributes that help you assess user behavior that violates your organization's acceptable use policy
- Data changes
- Attributes required to comply with standards or regulations, such as preventing financial crimes, limiting equity trading, or collecting health or other personal information.
- Attributes that help you identify suspicious or unexpected behavior, such as attempts to perform unauthorized actions
- Configuration changes
- Application code file or memory changes

Event attributes

Each log entry needs to include sufficiently detailed information for monitoring and analysis. You could log full content data, but it's more efficient to log an extract or summary properties. The application logs must record the *when*, *where*, *who*, *what*, and *which* of each event. The properties

for these will be different depending on the architecture, class of application, and the host system or device.

When logging date and time stamps, use Coordinated Universal Time (UTC) and the internationally recognized date and time formats in ISO 8601 (ISO website).

> **Note**
>
> Consider using a network time synchronization service to help ensure accurate time stamps. Amazon provides the Amazon Time Sync Service, which is used by many AWS services, including Amazon Elastic Compute Cloud (Amazon EC2). Amazon Time Sync Service uses a fleet of satellite-connected and atomic reference clocks in each AWS Region to deliver accurate current time readings of the UTC global standard through Network Time Protocol (NTP). For more information, see Keeping Time with Amazon Time Sync Service (AWS blog post).

The following event attributes are commonly included in logs.

Attribute category	Event attribute	Description
When	Logging date and time	Record the date and time that the event was added to the log.
	Event date and time	Record the date and time that the event occurred. This might be different than the logging record, such as when logging is delayed because the client applicati on is hosted on a remote device that is periodically or intermittently online.
	Event identifier	Log a user name, account number, or other unique attribute that ensures the

		event can always be identifie d.
Where	Application identifier	Log the application name and version.
	Application address	Log the cluster or hostname, server IPv4 or IPv6 address, port number, workstation identity, and local device identifier.
	Service	Log the service name and protocol.
	Geolocation	Log the geographic locations of the user.
	Window, form, or page	Log the entry point URL, HTTP method for a web application, or dialogue box name where the action was taken.
	Code location	Log the script or module name.
Who (human or machine user)	Source address	Log the user's device identifie r, IP address, cellular or radio frequency (RF) tower ID, or mobile telephone number.
	User identity	If the user is authenticated or otherwise known, log the user database table primary key value, user name, or license number.

	User type classification	Log the type of user, such as public, authenticated, CMS, search engine, authorized penetration tester, or uptime monitor. For more informati on about uptime monitors, see Cautions and exclusions in this guide.
	Request HTTP headers or HTTP user agent	(Web applications only) Log HTTP request header information, including the HTTP user-agent string, because these values affect the information that the client sends to the server.
What	Type of event	Log whether the event is informational, a warning, or an error.
	Severity of event	Classify the event severity, such as high, medium, and low.
	Security event flag	If the log contains data not related to security events, create a flag for security-related events to help you identify them.
	Event description	(Optional) Include a brief description of the event.

Action or intent	Log the original intended purpose of the request, such as logging in, refreshing the session ID, logging out, or updating a profile.
User or application response	Log the user's or application's response to the event, such as a status code, custom text messages, stopping the session, or administrator alerts.
Result status	Log whether the action was successful, such as success, fail, or defer.
Result reason	Log the reason the status occurred. For example, a sign in request might fail because the user is not authenticated in the database.
Extended details	Log any additional information associated with the event, such as a stack trace, system error messages, debug information, and the HTTP request body.
HTTP response status code	(Web applications only) Log the HTTP response status code returned to the user, such as `200` or `301`. For more information, see Logging levels in this guide.

Which	Resources affected	Log which resources were acted upon.
	Object	Log affected components or other objects, such as a user account, data resource, file, URL, or session ID.
	Name of resource	Log the names of affected resources.
	Resource tags	Log the tags assigned to the affected resources. For more information about tags, see Tagging AWS resources (AWS General Reference).
Other	Analytical confidence	Record the logging service's confidence in the event detection, such as assigning a low, medium, or high rating or a numeric value.
	Internal classifications	Log any internal classific ations for standards or compliance adherence.
	External classifications	Log any external classific ations for standards or compliance adherence, such as NIST Security Content Automation Protocol (SCAP).

Logging best practices

Logging levels

Be careful not to log an excessive amount of data. Logs should capture useful and actionable data. Excessive logging can negatively impact performance, and it is can also increase logging storage and processing costs. Excessive logging can also result in issues and security events going undetected.

Logging HTTP response status codes can generate a significant amount of log data, especially 200-level (success) and 300-level (redirection) status codes. We recommend that you consider logging only 400-level (client-side errors) and 500-level (server-side errors) status codes.

Application logging frameworks provide different levels of logging, such as **info**, **debug**, or **error**. For development environments, you might want to use verbose logging, such as including **info** and **debug**, to help your developers. However, we recommend that you disable **info** and **debug** levels for production environments because these can generate excessive logging data.

Cautions and exclusions

- Ensure that the data you are logging is legally permitted, particularly in jurisdictions where your organization operates.
- Don't exclude any events from known users (such as other internal systems), trusted third parties, search engine robots, uptime or process monitors, and other remote monitoring systems. However, you can include a classification flag for each of these in the recorded data. Consider that log files generated by your application might be used by parties, such as third-party log monitoring solutions or external service providers, that are not authorized to view any sensitive data that the application processes.
- The following attributes should not be recorded directly in the logs. Remove, mask, sanitize, hash, or encrypt the following:
  - Application source code
  - Session identification values (consider replacing this with a hashed value if you need to track session-specific events)
  - Access tokens
  - Sensitive personal data and some forms of personally identifiable information (PII), such as health information or government-issued identifiers
  - Authentication passwords

- Database connection strings
- Encryption keys and other primary secrets
- Bank account or payment card holder data
- Data of a higher security classification than the logging system is permitted to store
- Commercially sensitive information
- Information that is illegal to collect in the relevant jurisdictions
- Information that a user has either opted out of or hasn't explicitly consented to collection
- Information that the consent to collect has expired

Special data types

Sometimes, the following data can also be recorded in logs. While it can be useful for investigative and troubleshooting purposes, it can reveal sensitive information about the system. You might need to anonymize, hash, or encrypt these data types before the event is recorded:

- File paths
- Internal network names and addresses
- Non-sensitive personal data, such as personal names, telephone numbers, and email addresses

Use data anonymization if the individual's real identity is not required in the log or if the risk is considered too great.

Access and change management

- Non-administrative users should not be able to disable logging of events, especially those that are necessary to meet compliance requirements.
- Only administrative users should be able to pause or stop logging services or modify configurations.
- If your logging service has a log file integrity validation feature, enable it. This helps you detect modification, deletion, or forging of log files. For more information about this feature in AWS services, see Using CloudTrail in this guide.
- Logging changes must be intrinsic to the application, such as made automatically by the application based on an approved algorithm, or follow an approved change management processes, such as when you change configuration data or modify the source code.

AWS services for logging and monitoring

This guide focuses on logging and monitoring applications deployed in the AWS Cloud. You can use AWS services to implement your logging and monitoring plan, or you can use them to augment your current solutions. For example, if you are troubleshooting an issue with your application, you might:

- Triage the application logs with the VPC Flow Logs feature in Amazon Virtual Private Cloud (Amazon VPC) and view the network traffic that corresponds to the issue.
- Use AWS CloudTrail to view the API calls that correspond to the issue event times.
- Review the logs in Amazon CloudWatch Logs to check for CPU spikes that correspond to the issue event times.

You can deploy the following AWS services and features for logging and monitoring your application:

- AWS CloudTrail helps you audit the governance, compliance, and operational risk of your AWS account by recording the actions taken by a user, role, or an AWS service. For more information about using this service to log or monitor events for your application, see CloudTrail in this guide.
- Amazon CloudWatch helps you analyze logs and, in real time, monitor the metrics of your AWS resources and hosted applications. You can also use the ServiceLens feature to monitor the health of your application or use the Synthetics feature to create canaries that monitor your endpoints and APIs. For more information about using this service to monitor your application, see CloudWatch in this guide.
- Amazon CloudWatch Logs helps you centralize the logs from all your systems, applications, and AWS services so you can monitor them and archive them securely. For more information about using this service to log events for your application, see CloudWatch Logs in this guide.
- The VPC Flow Logs feature of Amazon Virtual Private Cloud (Amazon VPC) captures information about the IP traffic going to and from network interfaces in your VPC. For more information about using this service to log events for your application, see VPC Flow Logs in this guide.
- AWS X-Ray collects data about requests that your application serves, and it helps you view, filter, and gain insights into that data to identify issues and opportunities for optimization. For more information about using this service to monitor your application, see X-Ray in this guide.

Application logging and monitoring using AWS CloudTrail

AWS CloudTrail is an AWS service that helps you enable operational and risk auditing, governance, and compliance of your AWS account. Actions taken by a user, role, or an AWS service are recorded as *events* in CloudTrail. Events can include actions taken in the AWS Management Console, AWS Command Line Interface (AWS CLI), and AWS SDKs and APIs.

Using CloudTrail

CloudTrail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a CloudTrail event. You can easily view recent events in the CloudTrail console by going to **Event history**.

For an ongoing record of activity and events in your AWS account, you create a *trail*. You can create trails for a single AWS Region or for all Regions. Trails record the log files in each Region, and CloudTrail can deliver the log files to a single, consolidated Amazon Simple Storage Service (Amazon S3) bucket.

You can configure multiple trails differently so that the trails process and log only the events that you specify. This can be useful when you want to triage events that occur in your AWS account with events that occur in your application.

> **Note**
>
> CloudTrail has a validation feature that you can use to determine whether a log file was modified, deleted, or unchanged after CloudTrail delivered it. This feature is built using industry standard algorithms: SHA-256 for hashing and SHA-256 with RSA for digital signing. This makes it computationally infeasible to modify, delete or forge CloudTrail log files without detection. You can use the AWS CLI to validate the files in the location where CloudTrail delivered them. For more information about this feature and how to enable it, see Validating CloudTrail log file integrity (CloudTrail documentation).

Use cases for CloudTrail

- **Compliance aid** – Using CloudTrail can help you comply with internal policies and regulatory standards by providing a history of events in your AWS account.

- **Security analysis** – You can perform security analysis and detect user behavior patterns by ingesting CloudTrail log files into a log management and analytics solutions, such as CloudWatch Logs, Amazon EventBridge, Amazon Athena, Amazon OpenSearch Service, or another third-party solution.
- **Data exfiltration** – You can detect data exfiltration by collecting activity data on Amazon S3 objects through object-level API events recorded in CloudTrail. After the activity data is collected, you can use other AWS services, such as EventBridge and AWS Lambda, to trigger an automatic response.
- **Operational issue troubleshooting** – You can troubleshoot operational issues by using the CloudTrail log files. For example, you can quickly identify the most recent changes made to the resources in your environment, including creation, modification, and deletion of AWS resources.

Best practices for CloudTrail

- Enable CloudTrail in all AWS Regions.
- Enable log file integrity validation.
- Encrypt logs.
- Ingest CloudTrail log files into CloudWatch Logs.
- Centralize logs from all AWS accounts and Regions.
- Apply lifecycle policies to S3 buckets containing log files.
- Prevent users from being able to turn off logging in CloudTrail. Apply the following service control policy (SCP) in AWS Organizations. This SCP sets an explicit deny rule for the `StopLogging` and `DeleteTrail` actions across the organization.

```
{
"Version": "2012-10-17",
"Statement":
       [
                 { "Action":
                     [
                     "cloudtrail:StopLogging",
                     "cloudtrail:DeleteTrail"
                      ],
                      "Resource": "*",
                      "Effect": "Deny"
                  }
        ]
```

```
}
```

Application logging and monitoring using Amazon CloudWatch

Amazon CloudWatch monitors your AWS resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track *metrics*, which are variables you can measure for your resources and applications.

Using CloudWatch

CloudWatch is, essentially, a metrics repository. An AWS service, such as Amazon EC2, puts metrics into the repository, and you retrieve statistics based on those metrics. If you put your own custom metrics into the repository, you can also retrieve statistics on these metrics. For more information, see Using CloudWatch metrics (CloudWatch documentation).

You can also configure *alarms*, which automatically initiate actions on your behalf. An alarm watches a single metric over a specified time period and performs one or more specified actions, based on the value of the metric relative to a threshold over time. For example, the alarm might send a notification to an Amazon Simple Notification Service (Amazon SNS) topic. You can also add alarms to dashboards. For more information, see Using CloudWatch alarms (CloudWatch documentation).

The CloudWatch console automatically displays metrics about every AWS service you use. You can create additional, custom dashboards to display metrics and alarms for your applications. For more information, see Using CloudWatch dashboards (CloudWatch documentation).

CloudWatch automatically supports cross-Region functionality. You do not need to take any extra steps to display metrics from different AWS Regions in a single account on the same graph or dashboard. You can achieve cross-account functionality by implementing cross-account observability (CloudWatch documentation).

For more information and detailed guidance about using CloudWatch to log and monitor workloads in the AWS Cloud, see Designing and implementing logging and monitoring with Amazon CloudWatch (AWS Prescriptive Guidance).

Use cases for CloudWatch

- **Application health monitoring** – CloudWatch ServiceLens enhances the observability of your services and applications by enabling you to integrate traces, metrics, logs, alarms, and

other resource health information into one place. ServiceLens integrates CloudWatch with AWS X-Ray to provide an end-to-end view of your application to help you more efficiently pinpoint performance bottlenecks and identify impacted users. For more information, see Using ServiceLens to monitor the health of your applications (CloudWatch documentation).

- **Synthetic monitoring** – You can use CloudWatch Synthetics to create canaries, configurable scripts that run on a schedule, to monitor your endpoints and APIs. Canaries follow the same routes and perform the same actions as a customer, which makes it possible for you to continually verify your customer experience even when you don't have any customer traffic on your applications. Canaries check the availability and latency of your endpoints and can store load time data and screenshots of the UI. They monitor your REST APIs, URLs, and website content, and they can check for unauthorized changes from phishing, code injection, and cross-site scripting. For more information, see Using synthetic monitoring (CloudWatch documentation).
- **User monitoring** – With CloudWatch RUM, you can perform real user monitoring to collect and view client-side data about your web application performance. The data includes page load times, client-side errors, and user behavior. You can use the collected data to quickly identify and debug client-side performance issues. For more information, see Using CloudWatch RUM (CloudWatch documentation).
- **Anomalous behavior detection** – When you enable *anomaly detection* for a metric, CloudWatch applies statistical and machine learning algorithms. These algorithms continuously analyze metrics of systems and applications, determine normal baselines, and surface anomalies. For more information, see Using CloudWatch anomaly detection (CloudWatch documentation).
- **Feature validation and A/B experiments** – You can use Amazon CloudWatch Evidently to safely validate new features by serving them to a specified percentage of your users while you roll out the feature. You can also conduct A/B experiments to make feature design decisions based on evidence and data. For more information, see Perform launches and A/B experiments with CloudWatch Evidently (CloudWatch documentation).

Application logging and monitoring using Amazon CloudWatch Logs

Amazon CloudWatch Logs enables you to centralize the logs from all of your systems, applications, and AWS services that you use, in a single, highly scalable service. You can then easily view them, search them for specific error codes or patterns, filter them based on specific fields, or archive them securely for future analysis. You can see all of your log events, regardless of their source, as

a single and consistent flow of events ordered by time. You can query them and sort them, group them by specific fields, create custom computations, and visualize log data in dashboards.

Using CloudWatch Logs

In CloudWatch Logs, log events are organized into log streams and log groups. A *log stream* is a sequence of log events that share the same source. More specifically, a log stream is generally intended to represent the sequence of events coming from the application instance or resource being monitored. *Log groups* define one or more log streams that share the same retention, monitoring, and access control settings. Each log stream must belong to at least one log group. For more information, see Working with log groups and log streams (CloudWatch Logs documentation).

You can use CloudWatch Logs Insights to search and analyze your log data in Amazon CloudWatch Logs. You can perform queries to help you more efficiently and effectively respond to operational issues. If an issue occurs, you can use CloudWatch Logs Insights to identify potential causes and validate deployed fixes. For more information, see Analyzing log data with CloudWatch Logs Insights (CloudWatch Logs documentation).

You can search and filter the log data coming into CloudWatch Logs by creating one or more *metric filters*. Metric filters define the terms and patterns to look for in log data as it is sent to CloudWatch Logs. CloudWatch Logs uses these metric filters to turn log data into numerical CloudWatch metrics that you can graph or set an alarm on. For more information, see Creating metrics from log events using filters (CloudWatch Logs documentation).

Use cases for CloudWatch Logs

- **Monitoring CloudTrail logs** –– You can create alarms in CloudWatch and receive notifications of particular API activity, as captured by CloudTrail, and use the notification to perform troubleshooting. For more information, see Sending CloudTrail Events to CloudWatch Logs (CloudTrail documentation).
- **Logging AWS API calls** – If you have a third-party monitoring solution in place, you can use CloudWatch Logs to log AWS API calls. You set up the third-party monitoring service to evaluate this log and the application-level APIs.
- **Configuring log retention** – By default, logs in CloudWatch Logs are kept indefinitely and never expire. You can adjust the retention policy for each log group, keeping the indefinite retention, or choosing a retention period between one day and 10 years.

- **Archiving and storing logs** – You can use CloudWatch Logs to store your log data in highly durable storage. The CloudWatch Logs agent sends both rotated and non-rotated log data into the log service. You can then access the raw log data when you need it.

Application logging and monitoring using VPC Flow Logs

VPC Flow Logs is a feature of Amazon Virtual Private Cloud (Amazon VPC) that helps you capture information about the IP traffic going to and from network interfaces in your VPC.

Using VPC Flow Logs

You can create a flow log for a virtual private cloud (VPC), a subnet, or a network interface. If you create a flow log for a subnet or VPC, each network interface in that subnet or VPC is monitored. For more information, see Work with flow logs (Amazon VPC documentation).

Flow log data for a monitored network interface is recorded as flow log records. A *flow log record* represents a network flow in your VPC. By default, each record captures a network IP traffic flow that occurs within an aggregation interval. Each record is a string with fields separated by spaces. A record includes values for the different components of the IP flow, for example, the source, destination, and protocol. When you create a flow log, you can use the default format for the flow log record, or you can specify a custom format. For more information, see Flow log record examples (Amazon VPC documentation).

Flow logs don't capture the following information:

- Traffic generated by instances when they contact the Amazon Domain Name System (DNS) server. If you use your own DNS server, then all traffic to that DNS server is logged.
- Traffic generated by a Windows instance for Amazon Windows license activation.
- Traffic to and from `254.169.254`, for instance metadata.
- Traffic to and from `254.169.123`, for the Amazon Time Sync Service.
- Dynamic Host Configuration Protocol (DHCP) traffic.
- Traffic to the reserved IP address for the default VPC router.
- Traffic between an endpoint network interface and a Network Load Balancer network interface.

Flow log data can be published to several AWS services, including Amazon CloudWatch Logs. After you create a flow log, you can retrieve and view the flow log records in CloudWatch Logs in the

log group that you configure. For more information, see Publish flow logs to CloudWatch Logs (Amazon VPC documentation).

Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency. You can create or delete flow logs without any risk of impact to network performance.

Use cases for VPC Flow Logs

- Diagnose overly restrictive security group rules
- Monitor the traffic that is reaching your application instance
- Determine the direction of the traffic

Application logging and monitoring using AWS X-Ray

AWS X-Ray collects data about requests that your application serves, and it helps you view, filter, and gain insights into that data to identify issues and opportunities for optimization.

Using X-Ray

AWS X-Ray receives traces from your application and, if they're integrated with X-Ray, from the AWS services that your application uses. X-Ray samples and visualizes requests on a service graph when they flow through your application components. X-Ray generates trace identifiers so that you can correlate a request when it flows through multiple components, which helps you view the request from end to end. You can further enhance this by including annotations and metadata to help uniquely search for and identify the characteristics of a request.

We recommend that you configure each server or endpoint in your application with X-Ray. X-Ray is implemented in your application code by making calls to the X-Ray service. X-Ray also provides AWS SDKs for multiple languages, including instrumented clients that automatically send data to X-Ray. The X-Ray SDKs provide patches to common libraries used for making calls to other services (for example, HTTP, MySQL, PostgreSQL, or MongoDB).

For more information, see Tracing applications with AWS X-Ray (AWS Prescriptive Guidance).

Use cases for X-Ray

- **Application analysis and debug** – Trace data can help you debug the application by providing an end-to-end view of the request so that you can identify bottlenecks and troubleshoot issues. The

X-Ray service map is a visual tool that helps you identify where errors are occurring, connections with high latency, or traces for unsuccessful requests.

- **Performance analytics** – The Analytics console is an interactive tool for interpreting trace data to quickly understand how your application and its underlying services are performing. The console helps you explore, analyze, and visualize traces. You can also compare trace sets with different conditions, for root cause analysis.

Frequently asked questions

Can I use my current monitoring service?

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and application owners. It provides data and actionable insights to help you monitor your applications, respond to system-wide performance changes, and optimize resource utilization. However, if you have an established monitoring service in place, you do not need to replace it.

How do I stop the log files from being tampered with?

You can enable log file integrity validation. It is good practice to manage and store your logs in a dedicated AWS account and restrict access to that account. For more information, see Using CloudTrail in this guide.

Do I have to maintain separate log files for each application?

No, you can consolidate the log data from multiple applications into the same log file. However, make sure that a unique identifier for each application is recorded in the log stream.

Resources

AWS documentation

- AWS CloudTrail documentation
- AWS CloudWatch documentation
- AWS CloudWatch Logs documentation
- Amazon VPC Flow Logs documentation
- AWS X-Ray documentation
- Designing and implementing logging and monitoring with Amazon CloudWatch (AWS Prescriptive Guidance)

AWS marketing

- AWS CloudTrail
- Amazon CloudWatch
- Centralized Logging on AWS (AWS Solutions)
- Monitoring and Observability (AWS Cloud Operations)
- How to Monitor your Applications Effectively (AWS Startups)

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	—	January 6, 2023

AWS Prescriptive Guidance glossary

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

Numbers

7 Rs

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

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

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

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

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

aggregate function

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

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

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

anti-pattern

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

application control

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

application portfolio

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

artificial intelligence (AI)

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

artificial intelligence operations (AIOps)

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

asymmetric encryption

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

atomicity, consistency, isolation, durability (ACID)

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

attribute-based access control (ABAC)

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

authoritative data source

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

Availability Zone

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

AWS Cloud Adoption Framework (AWS CAF)

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

AWS Workload Qualification Framework (AWS WQF)

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

B

bad bot

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

BCP

See business continuity planning.

behavior graph

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

big-endian system

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

binary classification

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

bloom filter

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

blue/green deployment

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

bot

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

botnet

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

branch

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

break-glass access

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

brownfield strategy

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

buffer cache

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

business capability

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

business continuity planning (BCP)

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

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

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

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

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

chaos engineering

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

CI/CD

See continuous integration and continuous delivery.

classification

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

client-side encryption

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

Cloud Center of Excellence (CCoE)

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

cloud computing

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

cloud operating model

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

cloud stages of adoption

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

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

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

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or 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.

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.

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.

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.

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.

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.

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.