AWS Cloud Adoption Framework: Operations Perspective

AWS Whitepaper

AWS Cloud Adoption Framework: Operations Perspective: AWS Whitepaper

Table of Contents

AWS Cloud Adoption Framework: Operations Perspective

Publication date: **November 9, 2022** (*Document revisions* (p. 32))

Abstract

As the proliferation of digital technologies continues to disrupt market segments and industries, adopting Amazon Web Services (AWS) can help you transform your organization to meet changing business conditions and evolving customer needs. As the world's most comprehensive and broadly-adopted cloud platform, AWS can help you reduce business risk; improve environmental, social, and governance (ESG) performance; increase revenue; and improve operational efficiency.

The AWS Cloud Adoption Framework (AWS CAF) uses AWS experience and best practices to help you digitally transform and accelerate your business outcomes through innovative use of AWS. Use AWS CAF to identify and prioritize transformation opportunities, evaluate and improve your cloud readiness, and iteratively evolve your transformation roadmap.

AWS CAF groups its guidance in six perspectives: *Business*, *People*, *Governance*, *Platform*, *Security*, and *Operations*. Each perspective is covered in a separate whitepaper. This whitepaper covers the Operations perspective, which focuses on ensuring that cloud services are delivered at a level that is agreed upon with your business stakeholders.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

Millions of AWS customers (including the fastest-growing startups, largest enterprises, and leading government organizations) are using AWS to migrate and modernize legacy workloads, become data-driven, automate and optimize business processes, and reinvent operating and business models. Through cloud-powered digital transformation, they are able to improve their business outcomes, including reduced business risk, improved ESG performance, increased revenue, and improved operational efficiency.

Organizational ability to effectively use the cloud to digitally transform (organizational cloud readiness) is underpinned by a set of foundational capabilities. A *capability* is an organizational ability to use processes to deploy resources (people, technology, and any other tangible or intangible assets) to achieve a particular outcome. The AWS CAF identifies these capabilities, and provides prescriptive guidance that thousands of organizations around the world have successfully used to improve their cloud readiness and accelerate their cloud transformation journeys.

AWS CAF groups its capabilities in six perspectives:

- Business
- People
- Governance
- Platform
- Security
- Operations

Each perspective comprises a set of capabilities that functionally-related stakeholders own or manage in their cloud transformation journey.

The *Operations* perspective focuses on ensuring that cloud services are delivered at a level that is agreed upon with your business stakeholders. It comprises nine capabilities, as shown in the following figure (*AWS CAF Operations perspective capabilities*). Common stakeholders include infrastructure and operations leaders, site reliability engineers, and information technology service managers.

This whitepaper provides an overview of the benefits of operating in the AWS cloud, and introduces you to the Cloud Operations services and prescriptive guidance that will help you operate efficiently and effectively at scale.

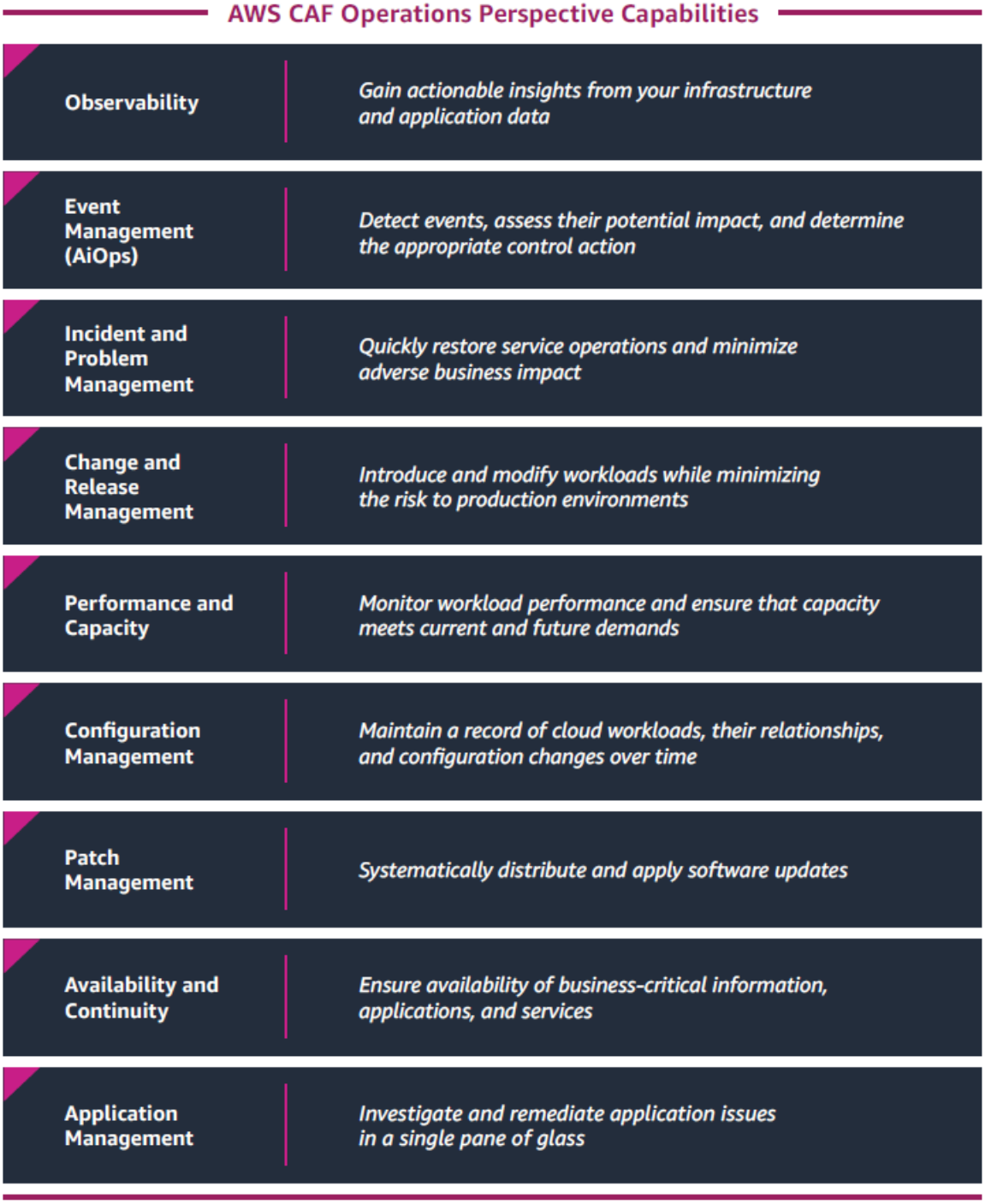
Operations is critical to the success of every organization and its digital transformation. Operational excellence is required to ensure that your transformation achieves its purpose and that applications consistently meet their business outcomes and the expectations of their users.

ITIL and AWS CAF are compatible. Like ITIL, AWS CAF organizes and describes all of the activities and processes involved in planning, creating, managing, and supporting modern IT services. It offers practical guidance and comprehensive guidelines for establishing, developing, and running cloud-based IT capabilities.

AWS and the AWS Partner Network (APN) provide tools and services that can help you along each step of the way. AWS Professional Services is a global team of experts that provides assistance through a collection of AWS CAF aligned offerings that can help you achieve specific outcomes related to your cloud transformation.

AWS Managed Services (AMS) helps you operate your AWS infrastructure more efficiently and securely. AMS can serve as an enabler for organizations that are building up operational rigor and lack the cloud operations staff or in-house AWS skills to achieve operational excellence at scale. Using AWS services and a growing library of automations, configurations, and runbooks, AMS can augment and optimize your operational capabilities in both new and existing AWS environments.

AWS Enterprise Support customers can benefit from a number of operational workshops and deep dives delivered by AWS Support experts. These services help you review the health of your cloud operations, optimize costs, and scale workloads efficiently.



AWS CAF Operations perspective capabilities

Observability

Gain actionable insights from your infrastructure and application data.

Observability describes how well you can understand what is happening in a system, often by instrumenting it to collect metrics, logs, or traces. In the cloud, observability can be hard to achieve due to sheer system complexity. Whether in data centers or in the cloud, to achieve operational excellence and meet business objectives, you need to understand how your systems are performing. Observability solutions enable you to collect and analyze data from applications and infrastructure so you can understand their internal states and be alerted to, troubleshoot, and resolve issues with application availability and performance to improve the end user experience.

The primary purpose of observability is to enable you to detect and investigate problems, but it also has a secondary purpose that enables you to define and measure Key Performance Indicators (KPIs) and Service Level Objectives (SLOs), such as uptime. For most organizations, important operations KPIs include mean time to detect (MTTD) and mean time to recover (MTTR) from an incident.

As well as providing operations teams with actionable data, observability helps you prioritize your initiatives by enabling you to determine if you are serving the needs of your customers and meeting your business outcomes. Observability can help you find performance improvements in your cloud resources that in turn enable you to reduce costs and improve the customer experience.

“You can’t improve what you don’t measure.” – Peter Drucker

Start

You’ll often see monitoring, tracing, and logging described as the “three pillars of observability”. To start with, you should focus on collecting, visualizing, alerting, and analyzing logs and metrics. In this section, we’ll also discuss using synthetic transactions to check the availability and performance of your applications because they provide powerful insights with minimal effort.

Use Amazon CloudWatch Metrics to centralize data about the performance of your systems. By default, many AWS services provide free metrics for resources, such as Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon Elastic Block Store (Amazon EBS) volumes, and Amazon Relational Database Service (Amazon RDS) instances. Visualizing and alerting on these infrastructure metrics should be your starting point.

Use Amazon CloudWatch Logs to centralize the logs from all of your workloads, in a single, highly-scalable service. You can then easily view logs, search them for specific error codes or patterns, filter them based on specific fields, or archive them securely for future analysis. Many AWS services also provide vended logs that enable you to troubleshoot issues.

If you are at the beginning of your cloud journey or you are migrating to cloud-native observability tools for the first time, you may want to consider your EC2 instances as your starting point, beyond vended metrics and logs provided by AWS. To start collecting logs and metrics from EC2 instances and on-premises servers, install and run the CloudWatch agent using AWS Systems Manager.

Once you have metrics and logs from managed services and EC2 instances, you can begin to visualize them while triggering alerts when thresholds are breached using Amazon CloudWatch alarms. It is important to test your applications prior to deployment in production, to fully understand which metrics and log events indicate that there could be an issue. Use a combination of load, exception, and smoke testing (using synthetic transactions) to establish a baseline where possible. If it is not possible to establish a baseline, or if your application has a predictable but variable load, use anomaly detection to

apply statistical and machine learning algorithms to predict and surface anomalies with minimal user intervention. Use metrics explorer and CloudWatch Metrics Insights to explore your metric data in more detail.

Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view (even those resources that are spread across different Regions). Use CloudWatch dashboards to visualize your infrastructure and applications. Start with AMS and EC2 instances, then consider expanding your visualizations as you mature in your adoption of other managed services, such as serverless services like AWS Lambda and Amazon DynamoDB.

The moment you create your dashboards is also the ideal time to create alerts using CloudWatch alarms, because you will be considering which metrics are important to visualize and measure. A metric alarm watches a single CloudWatch metric, or the result of a math expression based on multiple CloudWatch metrics. The alarm performs one or more actions based on the value of the metric or expression relative to a threshold over a number of time periods. The action can send a notification to an Amazon Simple Notification Service (Amazon SNS) topic, perform an EC2 action or an EC2 Auto Scaling action, or create operational work items (OpsItem) or incident in AWS Systems Manager. At this stage, focus on alerting the appropriate people as quickly as possible.

If you have migrated from on-premises, it's also likely that you have important insights in your logs that you may want to know about. Use CloudWatch log metric filters to search for a pattern in logs in near real-time and expose that data as a metric. For example, an error encountered in a log file could be counted, exposed as a metric using a log metric filter, and then alerted upon using CloudWatch alarms.

When you have application and infrastructure metrics and logs in place, you can also add synthetic transactions. They are easy to configure and can provide critical data on performance and availability. In the absence of a better metric, availability metrics from synthetic canaries can also be used to measure SLOs or agreements. You can use Amazon CloudWatch Synthetics to create canaries (configurable scripts that run on a schedule), to monitor your endpoints and application programming interfaces (APIs). Canaries follow the same routes and perform the same actions as a customer or user, which makes it possible for you to continually verify their experience, even when you don't have any traffic on your workloads.

Advance

Once you have metrics and alerts using CloudWatch alarms, use composite alarms to reduce noise and alert fatigue. A composite alarm includes a rule expression that considers the alarm states of other alarms that you have created. The composite alarm goes into ALARM state only if all conditions of the rule are met. The alarms specified in a composite alarm's rule expression can include metric alarms and other composite alarms.

As you begin to advance in your observability journey, log analytics becomes increasingly important to enable you to troubleshoot infrastructure and application issues. Use CloudWatch Log Insights and Contributor Insights for log analytics. Use CloudWatch Logs Insights to interactively search and analyze your log data in CloudWatch Logs. You can perform queries to help you more efficiently and effectively respond to operational issues and reduce your mean time to recovery (MTTR). You can use Contributor Insights to analyze log data and create time series that display contributor data. You can see metrics about the top-N (for example, top 10) contributors, the total number of unique contributors, and their usage. This helps you find top talkers (contributors to traffic) and understand who or what is impacting system performance. For example, you can find bad hosts, identify the heaviest network users, or find the URLs that generate the most errors.

Now that you have metrics, logs, and alerting in place, it's time to look at complete tracing of transactions through what are likely to become more distributed systems as your business matures. This stage will complete the three pillars of observability. AWS X-Ray receives traces from your application in addition to AWS services your application uses that are already integrated with X-Ray. Instrumenting

your application involves sending trace data for incoming and outbound requests and other events within your application, along with metadata about each request. Many instrumentation scenarios require only configuration changes. For example, you can instrument all incoming HTTP requests and downstream calls to AWS services that your Java application makes.

You can correlate your logs with your traces using software development kits (SDKs) that support this feature, or by injecting your logs with your trace IDs to simplify troubleshooting. View your services and applications in CloudWatch ServiceLens to visualize a complete view of your application with correlated metrics, logs, and traces. Use the visualization and associated traces, metrics, and logs to efficiently pinpoint performance bottlenecks or faults, helping you reduce your MTTR by reducing the time required to understand where an issue occurs. Use ServiceLens metrics, such as latency, to alert you before your customers notice there is a potential issue.

Excel

By now, you should have the three pillars of observability in place, so you can easily troubleshoot and get to the root cause of an issue quickly and efficiently.

However, there is still more that can be done to measure complete application performance and make better use of existing data. To begin with, use CloudWatch Real User Monitoring (RUM) to monitor your web applications. With Amazon CloudWatch RUM, you can perform real user monitoring to collect client-side data about your web application performance from actual user sessions in near real-time. The data collected 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 and understand the user impact, including the number of users impacted, geolocations, and browsers.

As your adoption of AWS matures, you will probably find yourself using more serverless services and containers. CloudWatch Lambda Insights is a monitoring and troubleshooting solution for serverless applications running on AWS Lambda. The solution collects, aggregates, and summarizes system-level metrics, including CPU time, memory, disk, and network. It also collects, aggregates, and summarizes diagnostic information, such as cold starts and Lambda worker shutdowns to help you isolate issues with your Lambda functions and resolve them quickly.

Use CloudWatch Container Insights to collect, aggregate, and summarize metrics and logs from your containerized applications and microservices. CloudWatch automatically collects metrics for many resources, such as CPU, memory, disk, and network. Container Insights also provides diagnostic information, such as container restart failures, to help you isolate issues and resolve them quickly. You can visualize a map of your container resources, CloudWatch alarms related to metrics that Container Insights collects, and performance metrics.

At this point in your journey, you may be considering how to optimize cost when collecting high cardinality metrics at scale. The CloudWatch embedded metric format enables you to ingest complex high-cardinality application data in the form of logs, and to generate actionable metrics from them. You can embed custom metrics alongside detailed log event data, and CloudWatch automatically extracts the custom metrics so that you can visualize and alarm on them, for near real-time incident detection. This can help to reduce costs by reducing the requirement to call the PutMetricData API, and also help simplify the collection of metrics for Lambda and containers.

You may also want to consider some open-source options, particularly for the collection of metrics and traces. At the time of writing, logs are not available as part of OpenTelemetry; instead, use the CloudWatch agent or FluentBit to send logs to CloudWatch logs. You can use the AWS Distro for OpenTelemetry to collect metrics and traces in a variety of source formats from Amazon EC2, AWS Lambda, Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), and on-premises infrastructure, and send them to managed services such as X-Ray and CloudWatch as well as Amazon Managed Service for Prometheus. This gives your developers more options whilst maintaining data in a centrally managed location.

If you want to visualize data from more sources than CloudWatch and X-Ray, or if you want to manage dashboards for non-IAM users, you can use Amazon Managed Grafana to create dashboards from AWS data sources such as Amazon CloudWatch, Amazon OpenSearch Service, AWS X-Ray, AWS IoT SiteWise, Amazon Timestream, and Amazon Managed Service for Prometheus as well as many popular open-source, third-party, and other cloud data sources.

Event management (AIOps)

Detect events, assess their potential impact, and determine the appropriate control action.

An *event* is an observation of an action, occurrence, or change of state. Events can be planned or unplanned and they can originate internally or externally to the workload. For example, an advertising promotion for a retail site is a planned event that originates externally to the workload. In contrast, a component failure is an unplanned event that originates internally to the workload. Events that require a response are called *incidents*. By using the right measurement and threshold for normal operating condition, an event can be detected when the threshold is breached.

Using advanced machine learning techniques, you can reduce operational incidents and increase service quality. Artificial Intelligence for IT Operations (AIOps) can help you increase service quality by grouping related incidents, predict incidents before they happen, and classify new incidents and insights.

Start

Efficient and effective management of planned and unplanned operational events is required to achieve operational excellence. Application classification determines the criticality of workloads based on business impact and customer experience, simplifying event handling and prioritization of concurrent events.

Establish fine-grained alerts and thresholds with differing actions based on criticality. For example, applications serving critical user-flow of publishing documents should have rapid escalations, while less impactful slow disk saturation of less critical application might be addressed during business hours.

Modern applications, such as those running on microservices architectures, generate large volumes of data in the form of metrics, logs, and events. Use Amazon CloudWatch to collect, access, and correlate this data on a single platform from across all your AWS resources, applications, and services that run on AWS and on-premises servers, helping you break down data silos so you can easily gain system-wide visibility and quickly resolve issues.

CloudWatch simplifies the collection of technical metrics as it natively integrates with more than 70 AWS services (including Amazon EC2, AWS Lambda, Amazon ECS, Amazon EKS, Amazon DynamoDB, Amazon S3) and automatically publishes detailed 1-minute metrics and custom metrics with up to one-second granularity so you can dive deep into your logs for additional context. You can also use CloudWatch in hybrid cloud architectures by using the CloudWatch agent or API to monitor your on-premises resources.

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

CloudWatch and CloudTrail enable you to explore, download, archive, analyze and visualize your events, and respond to account activity across your AWS infrastructure. You can identify who or what took which action, what resources were acted upon, when the event occurred, and other details to help you analyze and respond to activity in your AWS account.

Advance

Once workloads have been designed to provide information necessary to understand their internal state (metrics, logs, and traces), you should seek to ensure that the correct events are being propagated. Track

events at the correct level of granularity and implement a mechanism to review and update thresholds, limits, and event handling rules. Centrally collect and store events of interest to help teams to view, investigate, and resolve them. Centralized and standardized experience improves timely management, detection and remediation as well as creating a lower barrier to onboard new operations engineers responsible for managing events.

Business and operational metrics derived from desired business outcomes enable you to understand the health of your workload, prioritize operations activities, and respond to events. Establishing metric baselines helps to improve operations, investigation, and intervention. Use established runbooks for well-understood events, and use playbooks to aid in investigation and resolution of issues. Prioritize responses to events based on their business and customer impact. Ensure that, if an alert is raised in response to an event, there is an associated process to be followed, with a specifically identified owner. Define, in advance, the personnel required to resolve an event and include escalation triggers to engage additional personnel as it becomes necessary, based on urgency and impact. Identify and engage individuals with the authority to decide on the course of action where there will be a business impact from an event response not previously addressed or authorized.

You can use CloudWatch anomaly detection to detect anomalous behavior in your environments. When you enable anomaly detection for a metric, CloudWatch applies machine learning (ML) algorithms to the metric's past data to create a model of the metric's expected values. The model assesses both trends and hourly, daily, and weekly patterns of the metric. Use CloudWatch to set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

Several AWS services publish CloudWatch metrics can be used to gain system-wide visibility into resource utilization, application performance, and operational health. However, with distributed systems that use these services, your application telemetry should capture information to enable situational awareness. Instrumentation requires explicit code that records how long tasks take, how often certain code paths are executed, metadata about what the task was working on, and what parts of the tasks succeeded or failed. Further, it may be important to follow the flow of a request using a trace ID, as the request enters the system and passes through various systems before it is fulfilled. AWS X-Ray makes it easy to analyze the behavior of your distributed applications by providing request tracing, exception collection, and profiling capabilities.

AWS Systems Manager OpsCenter provides a central location to view, investigate, and resolve OpsItems related to AWS resources. OpsCenter is integrated with Amazon EventBridge and CloudWatch and designed to reduce mean time to resolution for issues impacting AWS resources. OpsCenter aggregates information from AWS Config, CloudTrail logs, and Amazon EventBridge events, so you don't have to navigate across multiple console pages during your investigation.

Excel

Customers seeking to accelerate business goals such as availability, MTTD, and MTTR, are often challenged with identifying the correct KPIs. Production applications can experience a wide variety of issues, and proactively identifying all potential operational problems is a time-consuming and challenging task. This is also increasingly common in modern microservice-based architectures with distributed and decoupled components. Metrics and logs need to be gathered from workloads, which humans then need to assess and hypothesize on potential operational problems and resolutions. Knowing what metrics to measure and the purpose they serve, as well as implementing alert governance to separate signal from noise can help reduce alert fatigue for the operators. Too much noise can cause alert fatigue that can lead to alerts being missed or ignored, or to responses being delayed.

Amazon DevOps Guru can be used to provide ML-powered service insights that make it easy for developers and operators to automatically detect issues that can improve application availability and performance. Amazon DevOps Guru, which can be enabled at the AWS account or CloudFormation stack level, can detect issues by correlating metric anomalies, traces, changes, and log events triggered

by incidents. The service produces insights which are a collection of identified anomalies containing observations, recommendations, and analytical data that you can use to improve your operational performance.

Each insight contains a list of the metrics and events that were used to identify the unusual behavior, as well as specific recommendations that can help you improve the performance of your application. These insights can be surfaced directly within OpsCenter dashboards (as OpsItems) and integrated into a workflow for team visibility; you can also perform actions such as running runbooks (Systems Manager automation documents). Examples of automatically detected operational issues include missing or misconfigured alarms, early warning signs of resource exhaustion, and code and configuration changes that could lead to outages. DevOps Guru insights can notify engineers using SNS and perform customized actions using Lambda functions for further workflow integration, such as with ticketing systems or AWS Config.

Incident and problem management

Quickly restore service operations and minimize adverse business impact.

Monitored and observable workloads provide insight into system health and business function. For workloads to adhere to SLAs and maintain uptime, develop plans for identifying, responding to, and resolving incidents with workloads. AWS provides services that teams can use to manage minor and major incidents in an automated and repeatable way. This ensures MTTD and MTTR are reduced, and unplanned business disruption is kept to a minimum.

Start

Amazon CloudWatch metrics can be displayed on dashboards and integrated with alarms to send notifications to engineers. Workloads should have dashboards and alerts in place, with alerts notifying teams, not individuals. You can use SNS to send low-priority notifications to engineers through email and high-priority incidents through SMS. Alerts should notify before a problem affects users as this enables engineers to assess and remediate issues before they negatively impact business outcomes. Alerts should be created in response to a specific problem; this ensures that all alerts are relevant and engineers are conditioned to act on them immediately.

When creating alerts, always ask these three questions:

- Why am I monitoring this metric?
- Who should be notified when the threshold is breached?
- What is the business impact of a breach?

Runbooks and playbooks should be used to diagnose and resolve incidents; initially, runbooks and playbooks can be stored in knowledge base systems or repository READMEs. *Runbooks* are a set of predetermined steps or pieces of code that perform actions, such as daemon restarts or scaling activities. Playbooks are orchestrated steps which should be used to perform complex procedures or fault investigation.

Ensure all workloads have support contacts and that engineers understand when and how to engage with one another. Repeat incidents should be investigated by a problem management team who can work with stakeholders to identify why the issue has been occurring and develop a remediation plan. After major incidents, ensure stakeholders come together to discuss the incident in detail to prevent or reduce the impact of a future occurrence.

Advance

Alerts should have human-readable messages with embedded diagnostic information so engineers can mitigate impact and diagnose faster, reducing MTTR. CloudWatch alarms can automatically cue Amazon EC2 Auto Scaling actions, reboot EC2 instances, create AWS Systems Manager OpsItems, and cue incident playbooks using AWS Systems Manager Incident Manager.

AWS Systems Manager enables immediate and automated resolution of issues. AWS provides over 300 Systems Manager documents that automate common operational actions, such as running Ansible playbooks, modifying Auto Scaling group parameters, performing Secure Shell Protocol (SSH) troubleshooting steps, and installing patches. Custom documents can run commands on Amazon

EC2 instances or make API calls to the AWS control plane. Documents are JSON formatted, version controlled, and can be shared across AWS accounts, providing a mechanism to control runbooks centrally. Scripts and knowledge base articles can be translated into automation documents, which can be triggered in response to events to provide an immediate resolution.

Major incidents cannot be resolved by one team, and coordinated steps often need to be taken to restore normal business operations. In these scenarios, you can use AWS Systems Manager Incident Manager to orchestrate an incident. Major incidents require planned processes regarding stakeholder management, incident communications, instant messaging, incident roles, and more. Engineers also need access to dashboards, documentation, tools, historical incident details, scaling processes, and the means to investigate the issue. With AWS Systems Manager Incident Manager, this information can be kept inside one AWS Systems Manager document.

CloudWatch alarms can cue an Incident Manager playbook that starts an incident and begins an escalation plan to contact incident responders. All Incident Management information is visible with quick links for instant messaging groups. Incident commanders and engineers can walk through the steps in the playbook until the incident has been resolved. Teams can use an AWS provided playbook or create their own, with necessary customizations. These can be kept in version control systems to allow changes to be tracked over time and to ensure that one source of truth exists for how to manage incidents.

Incident Manager provides a post-incident analysis document that you can use after a major incident. This template contains AWS suggested best practices for post-incident analysis and can be modified as needed to suit your requirements. Processes, questions, timelines, follow-up actions, and summaries of the incident can be tracked centrally to prevent or reduce the impact of a future occurrence. After an incident you should perform a post-incident analysis to identify how you can detect and diagnose quicker, and what adjustments could be made to the metrics used for detection. Always ensure follow-up actions are tracked and completed; this can be done through tickets inside OpsCenter, and also synchronized with third-party tools such as Jira and ServiceNow using AWS Service Management Connectors.

Excel

Alerts and runbooks should be kept on a one-to-one mapping. Alerts should cue because of a genuine business reason and have a specific process to resolve the problem. Understanding workloads' common incident scenarios is what drives the plan for immediate and automated remediation. A known response to a CloudWatch alarm could be for an engineer to manually run a runbook that restarts daemons or cues, scaling events. Within AWS, this is a process that could be executed immediately in response to an event using Systems Manager Automation. This model has no human involvement, ensures responses are immediate, and has reduced margin for error. The configuration for the alarm, the triggering steps, and the runbooks can all be version controlled. This provides a central source of truth, a view of all configuration changes, and a repeatable pattern for other engineers to adopt.

AWS Fault Injection Simulator (FIS) can perform controlled chaos experiments to test workload resilience and engineers' ability to respond to events. You can create Fault injection templates to perform many actions, such as stopping instances, throttling API actions, or stressing CPUs. These activities provide insights into gaps associated with people, process, and tools, leading to reduced MTTR and MTTD, while also raising teams' familiarity with incident management.

Some customers may be constrained by skills and/or staff to provide 24x7x365 proactive monitoring and incident management for their AWS accounts and resources. AMS can provide that capability.

To continuously respond to issues and resolve them instantly, you need to monitor the correct metrics, define relevant KPIs, understand failure scenarios, and have defined remediation runbooks. Initially, these runbooks can be run by humans, but as confidence and maturity grows, CloudWatch and Systems Manager can implement remediations automatically. Keep all runbooks, alarms, and configuration inside version control; this provides centralized sources of truth, auditable changes, and templated usage. To

ensure incident response and remediation is fast, prepare for incidents in advance by creating Incident Manager playbooks with links, documentation, procedures, and contacts.

Change and release management

Introduce and modify workloads while minimizing the risk to production environments.

If you are migrating to the cloud, it's important to understand that the pace of change will accelerate dramatically. This pace of change acts as an enabler for the business to achieve agility and allow it to be competitive with its peers. It also means that you need to rethink the way in which IT operates. Project teams no longer have to liaise with procurement teams to generate purchase orders and then place them with suppliers to await delivery and installation. Services can now become available within minutes or even seconds.

Start

The first thing to keep in mind is that change management is not designed to minimize business risk; instead, the process should ensure that overall business risk is optimized. A good change management process in any environment should enable the delivery of business value whilst protecting the business by balancing risk against business value, and it should do so in a way that maximizes productivity and minimizes wasted effort or cost for all participants in the process.

Every change should deliver business value, and the change management processes should be geared towards enabling that delivery. Rather than acting as a gatekeeper, the process should enable developers to fulfil their function in adding business value, using the products they deploy to production environments.

The key concepts of change management remain the same in the cloud. Change delivers business value, and it should be efficient. Agile methodologies and the automation capabilities of the cloud go hand in hand with the core principles of change management, as they are also designed to deliver business value quickly and efficiently. Nevertheless, there are some key areas that may require existing change processes to be modified to adapt to new methods of delivering change.

Widening the scope of a standard change is the starting point for managing change in the cloud. Without doing this, you risk the change management process becoming a bottleneck for delivering business value. It's always worth considering the business impact and risk of not implementing a change, or introducing a delay, keeping in mind that the purpose of managing change is to optimize business risk.

If automation, pipelines, and deployment methods are in place, it may be possible to reconsider the approach to standard changes. A standard change is where:

- There is a defined trigger to initiate the change request.
- Actions are well known, documented, and proven
- Authority is given in advance (or pre-authorized).
- The risk is usually low.

If the appropriate automation, testing, and deployment strategies are put in place, this should result in a scenario where large, infrequent, and risky changes are transformed into small, frequent, low-risk changes. By understanding the risk-reduction strategies enabled by the cloud, it should be possible, and it may even be necessary, to widen the scope of a standard change to include deployments that would have previously been considered as normal changes due to the risks associated with them in traditional IT environments.

As changes become more frequent due to agile methodologies and increased automation, there is a risk that change management becomes overburdened with normal changes, which can lead to delaying

changes due to bandwidth constraints. Important details might be missed as changes are not properly scrutinized due to resource constraints. Both of these scenarios introduce business risk that change management aims to optimize. In an environment of small, frequent changes, standard changes should become the new normal so that proper scrutiny can be given to normal changes, optimizing business risk and enabling the delivery of business value.

Advance

Changes become safer the more that you automate them, and as outcomes become more predictable. Outside of development and deployment, the easiest place to start is with patching and standardization of items, such as agent installation and configuration.

Automate the deployment of patches to operating systems (OS') and applications without any human intervention and with automated rollback using AWS Systems Manager Patch Manager. Doing so will allow you to take automated actions, before and after deployment of patches, such as implementing post-deployment testing. This kind of automation allows you to continue to think differently about change management, because the risks have been dramatically reduced and the process has been proven. Inherently risky procedures that can be automated, such as patching, can be transformed into standard changes, because the risk has been significantly lowered and proven rollback strategies are in place.

At this point, you should consider integrating your existing IT Service Management (ITSM) tooling, such as ServiceNow or Jira Service Desk with cloud services including AWS Config, AWS Systems Manager Change Manager, AWS Systems Manager Incident Manager, and AWS Systems Manager OpsCenter to make it easier to manage and record incidents and changes that take place in the cloud.

Use infrastructure as code tools such as AWS CloudFormation or AWS Cloud Development Kit (AWS CDK) to deploy applications and infrastructure as code. This helps ensure consistency of deployments, reducing risk even further.

Excel

To minimize risk, you should make deployments small and frequent. Because this will inevitably increase the number of changes, it's vital that your deployment processes adapt to ensure repeatability and consistency throughout the lifecycle of your applications. You should automate your deployment and testing using an automated pipeline. This decreases the requirement for manual testing, and reduces the likelihood that your changes will need to be scrutinized by a change advisory board (CAB).

Operations must be able to support a new release or service before it is made available to the end user. With the correct tooling, this process can be largely automated, including the creation of documentation, provisioning of automated runbooks and playbooks, and building predefined and automated patching plans. This process can be made even more robust using the correct tooling to ensure that only pre-approved services are used. Use AWS CodePipeline to automate deployment, testing, documentation and the provisioning of runbooks, playbooks, dashboards, alerts, and patching schedules.

The focus of a test manager should be to automate service acceptance testing as much as possible. This is made easier in the cloud with a wide variety of tools that are available for both validation and testing.

- Use AWS Device Farm to test your application on multiple devices and platforms.
- Use synthetic monitoring to monitor your endpoints and APIs.
- Use test events to create unit tests for your Lambda functions.
- Use Lambda, AWS Fargate, or Amazon EC2 to automate pipelines using existing tooling or new tools designed for the cloud.

- Add unit tests and synthetic transactions to your CI/CD pipeline.

Performance and capacity management

Monitor workload performance and ensure that capacity meets current and future demands.

To ensure that your applications fulfil their business purpose, it's essential to measure performance and ensure that you do not reach capacity limits. Although the AWS Cloud can allow you to scale to unparalleled levels, it's important to understand that there are performance considerations and service quotas that need to be measured and acted upon.

Start

Many AWS services publish metrics to Amazon CloudWatch that should form the basis of the absolute minimum metrics you should be monitoring and alerting upon. As stated in the observability section of this whitepaper, you should also monitor and alert upon metrics collected from AWS using the CloudWatch agent or AWS Distro for OpenTelemetry for Amazon EC2 instances as well as managed services. For large scale applications, ensure that you load test your application in pre-production environments to gauge where you may reach performance limits.

While the cloud offers virtually infinite scalability, even for the largest organizations and applications, it's important to remember that managed services have quotas (formerly referred to as limits) that are designed to help guarantee the availability of AWS resources, and prevent accidental provisioning of more resources than needed. You must anticipate these quotas by running load tests in pre-production environments, to anticipate demand in production. These tests are vital to ensure that you do not encounter any unanticipated service quotas or hit any limits encountered by the design of your application.

Use Amazon CloudWatch metrics that are provided by AMS, as well as metrics provided by your EC2 instances through the CloudWatch agent, to ensure that your application will respond according to your business requirements. It is vital that you test these against service quotas in pre-production before deploying to production, but you must also continuously monitor these metrics in production. It is possible, and often desirable, that your actual demand will outstrip your anticipated demand. If this is the case, you need to have mechanisms that alert you to such changes so that you can respond accordingly.

AWS Service Quotas is a service that enables you to view and manage your quotas for AWS services from a central location. Along with looking up the quota values, you can also request a quota increase, monitor the usage of specific services API actions, and create alerts for them directly from the Service Quotas console. AWS Trusted Advisor gives you additional insight as to whether or not you are approaching or breaching limits.

Advance

To make full use of the tools available to you to measure performance, you need to monitor your metrics and make use of AWS services that give you enhanced insights. To gain full visibility into your applications' performance, you need to implement distributed tracing, one of the three pillars of observability.

AWS X-Ray is a distributed tracing system that can help you gain insights into how your applications communicate between each other, measure performance of functions or lines of code, and provide analytics capabilities correlated with other signals, such as metrics and logs. Additionally, CloudWatch Log Insights, Lambda Insights, Container Insights, and Metric Insights enable you to use enhanced insights into how your application is performing.

While load testing gives you an idea of how well your application sustains its performance at certain amount of traffic and its associated infrastructure capacity, your infrastructure will probably not always be performing at that level. This is why it's vital that you implement Auto Scaling for horizontal scaling wherever possible to allow your infrastructure to scale according to the traffic. AWS Auto Scaling helps your application scale by monitoring and adjusting capacity using metrics and user-defined thresholds.

Multiple AWS services provide serverless offerings, where AWS shifts the shared responsibility model from you to AWS. AWS takes most of the work such as scaling, servers/systems patching, and management. This lets you focus on your business use case and spend more time on innovation. When working with databases, AWS offers more than 15 purpose-built, fully managed database engines to choose from according to your use cases. Orchestrating containers at scale can be daunting and that's why AWS offers Amazon EKS, Amazon ECS, AWS Fargate, and other container services to make it easier to manage your underlying infrastructure. Choosing the right service and architecture for the job helps you achieve better performance and less management overhead.

Excel

Use testing to evaluate performance and capacity limits, and use scaling mechanisms to help you sustain your customer traffic and growth. You should adopt flexible architectures that enable you to scale globally. Extending your infrastructure to multiple regions can give you that extra mile of capacity. Using the tools discussed in the Observability section of this whitepaper, such as CloudWatch RUM, you can understand the performance impact on your customers around the globe and deploy accordingly.

Testing your applications should not be a one-time event before going to production. Continuously testing your applications helps you detect customer impact when issues occur and can surface issues before technical metrics become available. Use CloudWatch Synthetics and CloudWatch RUM as described in the Observability section to continuously monitor application performance, including when you have no active users. Build experimentations and design your applications around failures to help recover quickly. AWS FIS is a fully managed service to help you run experiments safely and easily implement chaos engineering. Chaos engineering is the practice of stressing an application in testing or production environments by creating disruptive events, such as sudden increase in CPU or memory consumption, observing how the system responds, and implementing improvements.

Configuration management

Maintain an accurate and complete record of all your cloud workloads, their relationships, and configuration changes over time.

To ensure that you have reliable and scalable systems, it is important to manage application configurations, particularly as complexity increases.

Start

Most software teams already organize their application source code by separating out executable code from application configuration data. When writing application code, a software engineer will determine what values in the application may change over time and separate those values out into a distinct configuration file. As the application development continues, an engineer may simply update a variable in configuration data without the need to recompile the code.

However, separating configuration data from code does not go far enough. While this separation does help teams to move faster, the next step is to separate the deployment of configuration data from the deployment of code. An example of this includes using AWS Systems Manager Parameter Store for managing application logging configuration data, including the file paths on the Amazon EC2 instance where the AWS Systems Manager Agent (SSM Agent) should read log data from the Amazon CloudWatch log group, and the Amazon S3 bucket to which a copy of those logs should be sent. Most significantly, you can also include a parameter to store the current verbosity level that the application should use to write to the logs. In this way, your application can typically run with an ERROR level of logging, keeping the volumes of data generated (and associated storage and processing overheads) low. Yet, in response to certain error conditions, or using a manual update of the parameter, you can set a more verbose level, helping support any subsequent investigations.

The first step beyond this initial separation of configuration from code is to change the application to continually poll configuration data at runtime. The application reads config data and adjusts accordingly, without a restart. If there is a change to the configuration data, the application will read that change and adjust its behavior in near real-time. An example of this includes putting parameters such as throttling limits into configuration data. Once a DevOps engineer wants to adjust throttling limits to react to the current conditions of the application, they can simply update only the configuration data variable. The app will read that new data, and, without restarting, adjust throttling limits.

Advance

While it's important to manage configurations of applications, the OS and other software on the instances also need to be managed to ensure the successful running of the workload. AWS Systems Manager State Manager provides a mechanism to do this immediately or to a schedule, with support for a variety of playbook formats, including Ansible playbooks, Chef recipes, or Managed Object Format (MOF) files. The tasks you can carry out in this way vary, but examples include updating the AWS paravirtual (PV) drivers on Microsoft Windows EC2 instances, forcing changes to logging configurations, installing packages, and updating dependencies.

As you begin to manage your application, infrastructure as code (IaC) and configuration management becomes even more important. Building secure, reliable, and efficient images for your EC2 instances is an important task. EC2 Image Builder provides a scalable means of creating your golden images. Once you've created your new Amazon Machine Image (AMI), your various application environments also need

to be instructed to make use of it. As you create your first AWS CloudFormation template to describe you application environment, you might initially hardcode definitions for EC2 instances or Auto Scaling groups. Later, you might improve things by moving such configurations to a parameter that still needs to be manually updated. Finally, you might consider using AWS Systems Manager Automation actions to automate the distribution of golden images across teams.

Excel

Using AWS AppConfig, you can make updates to configuration data in a validated and controlled way. AWS AppConfig lets you configure, validate, and deploy your application configuration data. You can choose among multiple sources of configuration data, including AWS Systems Manager Parameter Store, Amazon S3, or within AppConfig itself. Using AWS AppConfig validators, configuration data can be syntactically or semantically validated prior to deployment (thus, avoiding typos that could cause outages). Additionally, configuration data can be rolled out gradually or immediately, allowing teams to limit the blast radius of changes. And finally, if bad configuration changes are pushed out, tripping a CloudWatch alarm, AWS AppConfig can automatically roll back the latest configuration and restore the previous version.

There are three questions you should ask yourself when considering how to manage your application's dynamic configuration:

- **Will this value need to change in the future, and if so, how frequently?** It might seem obvious, but carefully consider if this value will need to be updated in the future. Think of some scenarios in which the value might need to change. Likelihood of change is a key indicator of the need to externalize configuration. Every creative engineer can probably brainstorm a scenario in which a value in the code could change, but how likely is that scenario?
- **Is there a scenario where, in a crisis or a feature launch, it would be helpful to change this value quickly without redeploying and restarting the app?** Many teams use operational configuration to toggle things such as logging verbosity, so they can see more details about problems in production. Updating dynamic configuration to address critical issues can be helpful for operations.
- **Is the value deterministic?** That is, is the value important only at startup or at runtime, too?

An approach known as continuous configuration provides the means of adjustment of software behavior that allows teams to move faster and safer. Instead of changing software with new release through a CI/CD pipeline, changes are made only to configuration data, which can happen immediately and can be scoped to limit its impact. AWS AppConfig has a capability called feature flags that provides one kind of continuous configuration. Engineers will develop a new feature, but hide it behind a feature flag so that end users cannot yet access it. Engineers can update the value of the flag to allow access to the new feature for a limited set of users. Over time, the flag can be adjusted to allow for more users to access the feature. Teams can measure operational impact as more users access the new functionality. This technique helps limit blast radius for operational or reputational impact, and if the new feature results in negative consequences, the feature can be immediately rolled back with a simple configuration change.

Patch management

Systematically distribute and apply software updates.

Patch management is the process of distributing and applying updates to software. A systematic approach to patch management will ensure that you benefit from the latest updates while minimizing risks to production environments.

If you are involved in application or infrastructure operations, you understand the importance of an OS patching solution that is flexible and scalable enough to meet the varied requirements from your application teams. In a typical organization, some application teams use an architecture that involves immutable instances, whereas others deploy their applications on mutable instances.

Immutable instance patching involves applying the patches to the AMIs that are used to provision the immutable EC2 application instances. Mutable instance patching involves an in-place patch deployment to running instances during a scheduled maintenance window.

Start

To get started with patch management on AWS, you first need to ensure that your Amazon EC2 instances are set up to register with AWS Systems Manager. Additionally, you can register hybrid environment resources, such as on-premises servers, edge devices, and virtual machines (VMs) with AWS Systems Manager, including VMs in other cloud environments. By registering your hybrid environment resources with Systems Manager, you can use a single tool to automate patching and other remote operations across your environment.

AWS Systems Manager Patch Manager uses patch baselines, which include rules for auto-approving patches within days of their release, in addition to a list of approved and rejected patches. Patch Manager provides predefined patch baselines for each of the OS' supported by Patch Manager. You can use these baselines as they are currently configured (you can't customize them), or you can create your own custom patch baselines. Custom patch baselines allow you greater control over which patches are approved or rejected for your environment. After the appropriate approval rules have been identified, custom patch baselines should be deployed across accounts and AWS Regions to ensure patch criteria for managed nodes is consistent and in-line with your security standards.

Ensure all managed nodes are performing patch scans on a scheduled basis. To quickly enable daily patch scans across your AWS Organization, use the Quick Setup Host Management configuration. Establish standard patch installation periods during well-defined maintenance windows. You can enable Systems Manager Explorer to aggregate patch compliance states, in addition to other operational data sources, and display data across multiple accounts and Regions.

Advance

Update your machine images using EC2 Image Builder, and include components to update and test patches before rolling out to production for your immutable resources.

For mutable resources, notify users in advance with the details of the upcoming updates, and allow them to defer patches when other mitigating controls are available. Establish standard processes to remediate zero-day vulnerabilities or specific patch installation using an install override list.

Schedule centralized multi-account and multi-Region patch scan or install operations using AWS Systems Manager Automation. Deploy resource data syncs in each account and Region where managed instances

are registered to aggregate patch compliance, association compliance, and inventory metadata details to a single S3 bucket.

You can deploy Systems Manager resource data syncs across your accounts and Regions to send inventory metadata, association compliance, and patch compliance data collected from all of your managed nodes to a single Amazon S3 bucket. Resource data sync then automatically updates the centralized data when new inventory data is collected. With all inventory data stored in a target Amazon S3 bucket, you can use services such as Amazon Athena and Amazon QuickSight to query and analyze the aggregated data.

To store patch compliance for long term storage or for auditory compliance and regulatory requirements, you can record using AWS Config and evaluate patch compliance using Config rules. AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. This includes how the resources are related to one another, and how they were configured in the past, so you can see how the configurations and relationships change over time.

Excel

To automate vulnerability management, enable Amazon Inspector. Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for vulnerabilities. Amazon Inspector automatically discovers and scans Amazon EC2 instances and container images residing in Amazon Elastic Container Registry (Amazon ECR) for software vulnerabilities and unintended network exposure. If your AWS environment has multiple accounts, you can centrally manage your environment through a single account by using AWS Organizations and designating an account as the delegated administrator account for Amazon Inspector. You can resolve Inspector findings using patch install operations or install override lists.

For application workloads that require customized multi-step custom patch processes, use Patch Manager lifecycle hooks. Patch lifecycle hooks enable pre-patching and post-patching hooks that allow custom, customer-specified steps to be run at different phases of the patching workflow. For example, consider a customer with a custom-built application that requires a procedure to start and stop the application. The customer can use patch lifecycle hooks to run a pre-patching custom script to safely shut down the application before performing the patching process. After patching is complete and the server has been rebooted, you can run a post-patching custom script to start the application and perform validation testing, to ensure it is operating as expected before signaling success of the overall patching process.

Availability and continuity management

Ensure availability of business-critical information, applications, and services.

Availability and continuity management, while included as one operational capability, have separate implications on the operations of your critical workloads. Continuity management deals with how your workload is architected to respond to a one-off event that causes a serious negative impact to your business. To maintain business continuity, you must work backwards from your organization's objectives to develop a strategy for avoiding loss of data and reducing downtime where your workload is not usable for customers. Availability focuses on more common smaller scale disruptions, such as network issues, software bugs, component failures, and load spikes. Availability and disaster recovery, which helps facilitate continuity management, make up the resiliency of your workloads.

Start

When getting started with your cloud adoption journey, you should focus on weaving in availability objectives for each application hosted in the cloud. Build highly available architecture by deploying Amazon EC2 instances to multiple Availability Zones (AZs) and using AWS Auto Scaling to facilitate self-healing architecture. Ensure high availability of your most critical data by enabling backups to another AWS Region using Amazon S3 Cross-Region Replication (CRR).

Another aspect of ensuring availability is monitoring the health of your workloads. Ensure monitoring is enabled through Amazon CloudWatch on all existing and newly provisioned instances to collect metrics and logs related to operational KPIs. Establish baseline monitoring to assess early indicators of performance degradation that will eventually impact availability of applications and business continuity.

When getting started with continuity management, focus on ensuring processes are in place to recover from failures. To begin with, enable AWS Backup for all relevant services. Adjust backup plans and retention period for critical environments to optimize cost. You can back up Amazon EC2 instances used by your workload as AMIs. Set up disaster recovery plans for your most business critical applications and adapt the plans based on your Recovery Time Objectives (RTOs) and Recovery Point Objectives (RPOs).

You can also use services such as AMS to accelerate the implementation and orchestration of high availability and business continuity in your AWS environment through automation flywheels and operational experience. AMS provides proactive monitoring and incident management (infrastructure and security) in your AWS accounts. AMS then goes further to develop a deeper operational understanding of the typical use cases, issues, and top failure and recovery scenarios and helps you perform failover tests.

Advance

Once your initial availability and continuity management capabilities have been operationalized, you should consider further improving them to optimize your business objectives and lower your business risk. Granular and context-based monitoring helps make informed decisions about the availability of your workloads. This includes correlation of multiple data points from Amazon CloudWatch and AWS Health Dashboard. This helps you understand workload health against business outcomes and declare outages to ensure business continuity objectives are met.

Refine the backups that were set up as part of your cloud foundations to holistically assess the strategy to balance business impact and risk; for example, consider using Amazon S3 Glacier Deep Archive to help reduce cost. Refine continuity management procedures through inspection of disaster recovery strategies for critical applications.

You should work with multiple stakeholders in the organization to understand your RTO and RPO requirements for each application and potential impact to end users. For the most stringent RTO requirements, you can implement automated failovers based on health checks. This should be examined regularly to ensure false signals are not triggering the failover as that can be expensive and add its own availability risks. Based on requirements and considerations for cost and complexity, you should select the most suitable disaster recovery strategies: backup and restore, pilot light, warm standby, or multi-site active/active.

Automate your disaster recovery plans to reduce manual burden and margin of error. This includes automating redeployment of infrastructure, configuration, and application code in a recovery region using AWS CloudFormation and AWS CodePipeline. Manage configuration drift in the recovery Region using AWS Config and AWS service quotas to continuously monitor AWS resource configurations. To help drive automation in business continuity procedures, use services such as Amazon Route 53 for global routing, AWS Global Accelerator for traffic routing for multi-Region applications, and Amazon CloudFront for traffic management during recovery. Test your continuity management strategy through operational game days to ensure that orchestration of failover is triggered in a timely manner and that the recovery is completed in line with your RTO and RPO requirements. Operational game days are also valuable for testing key stakeholder engagement and escalation process within the organization.

Excel

As you mature your cloud operations and ensure that your availability and continuity management capabilities are meeting the operational KPIs for the business, you should further fine-tune the overall operating mechanisms that lend themselves to achieving high availability. Adopt the Correction of Error (COE) process to improve the quality of operations by documenting and driving problem management of recurring issues. Define a standardized way to document critical root causes and ensure they are reviewed and addressed. Use the mechanism of iterative improvement (flywheel) to increase your operational efficiency and improve the business continuity KPIs.

For your most business-critical workloads, push the limits of operational excellence standards by including chaos engineering. Business owners can use AWS FIS to simulate the most complex and thoughtfully induced failures, and as a result help teams identify monitoring blind spots and performance bottlenecks. This exercise can help improve the availability of the most complex and distributed applications. Look to mature your continuity management plan, where disaster recovery is only a subset of this plan. Review other aspects of your business that could be impacted by a one-off failure event and develop mechanisms to periodically undertake business impact analysis and risk assessment to verify that the RTO and RPO objectives are aligned with the business objectives of your workload.

Application management

Investigate and remediate application issues in a single pane of glass.

An application is a logical group of AWS resources that you want to operate as a unit. Whether you take an approach of "you build it, you run it" or have separate operations teams, it is often simpler to view information and take operations actions in the context of an application rather than on individual components. Applications often have different requirements for cost, monitoring, compliance, automation, and many other factors. Being able to view information, make decisions, and take actions within the context of the application and its requirements simplifies operational oversight and accelerates the remediation of application issues.

Start

To get started with application management, use AWS Systems Manager Application Manager, Service Catalog AppRegistry, AWS CloudFormation, or AWS Launch Wizard to specify the AWS resources that define your application. Applications can be defined as custom applications using a combination of tags, AWS Resource Groups, or AWS CloudFormation stacks. Application Manager automatically imports metadata about resources that were created by AWS CloudFormation, AWS Launch Wizard, Amazon ECS, and Amazon EKS. Application Manager then displays those resources in predefined categories.

Once you have defined your application, you can view information about it. The Application Manager overview displays a summary of Amazon CloudWatch alarms, OpsItems, Amazon CloudWatch Application Insights, and AWS Config compliance and runbook history. To start with, focus on cost and monitoring.

Use the alarms that you have already created following the Observability (p. 4) guidance in this document to automatically visualize alarms in Application Manager. Enable Application Insights with the application that you have created to visualize additional alarms created automatically and to view insights using intelligent problem detection, metric anomalies, and log errors.

Log groups related to your application are listed in Application Manager, allowing you to see all of these log groups in one place, with a link to the log group itself for further analysis. The resources view in Application Manager lets you view all of the AWS resources associated with your application as well as viewing the cost of your application.

Advance

Once you have organized your applications and set up patch management (p. 21), you will be able to view AWS Systems Manager State Manager association compliance information, as well as other State Manager associations related to your application. If you have enabled AWS Config, you will be able to see Config rules compliance and Config Resource compliance, as well as the number of configuration changes that have been made to that resource.

OpsItems (p. 8) can also be viewed in the context of your application. The OpsItems tab displays OpsItems for resources in your application.

Excel

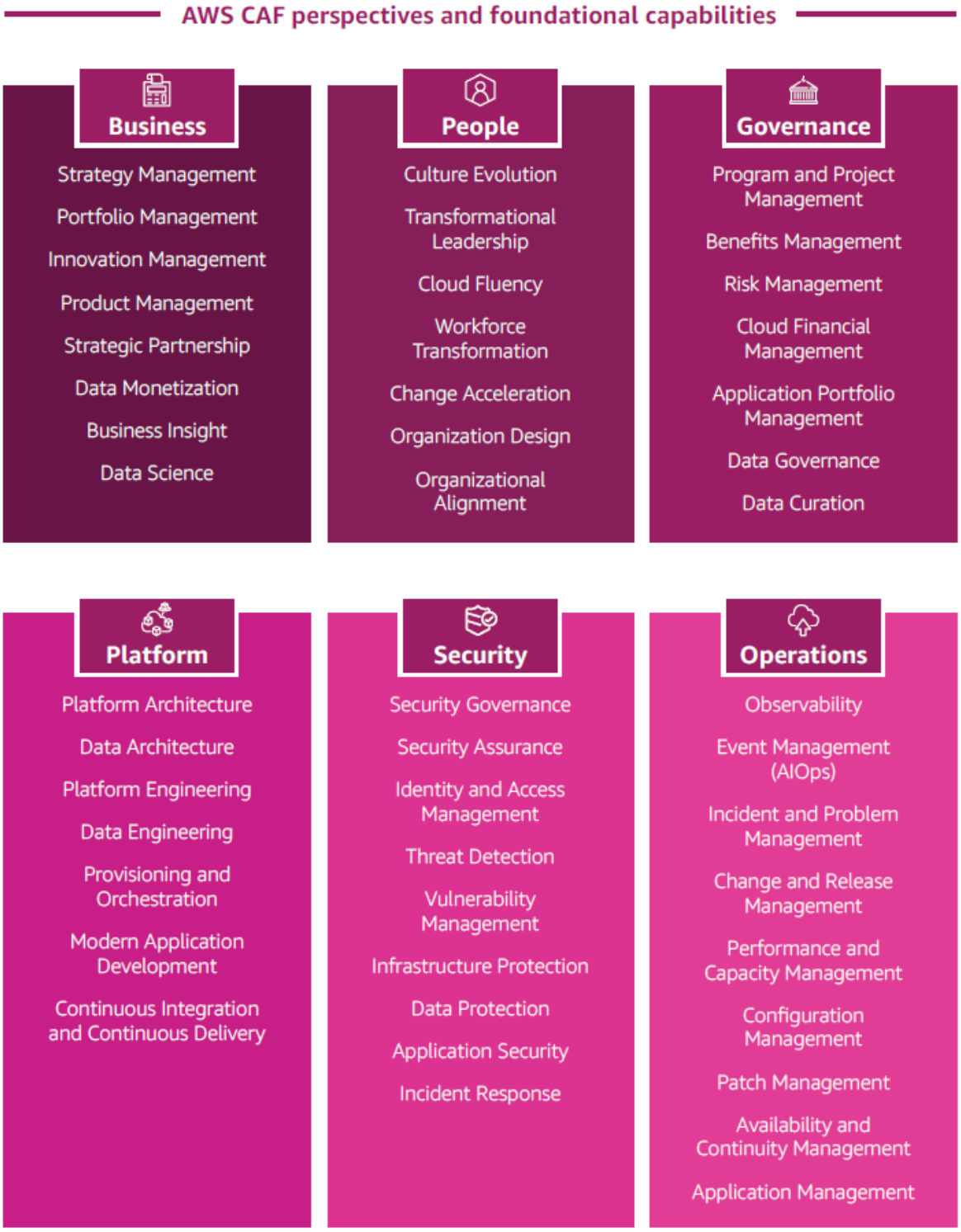
As described in the incident and problem management (p. 11) section, you can remediate issues in your application using Automation runbooks. You can start any runbook filtered by the type of resource used

in your application, or you can choose the name of the resource in your application, which will then filter runbooks by that resource type.

Conclusion

Operational excellence is critical to ensuring that your applications achieve their business objectives. A running theme through all of these sections is to build initially and to advance into automation and excel with full automation. There is conflicting data on this subject, but everyone has experienced major incidents as a direct result of change. Change can make up a large proportion of your major incidents. Automation of operations removes human error, and reduces the risk of change causing an incident. With techniques such as blue/green or canary deployments, this risk can also be further mitigated. By following the guidance in this document and automating your operations activities, you will see improvements to availability, security, and performance of your applications, while reducing costs.

Appendix: AWS CAF perspectives and foundational capabilities



AWS CAF perspectives and foundational capabilities

Contributors

Contributors to this document include:

- Alex Livingstone, Principal Worldwide Cloud Operations Specialist
- Arvind Raghunathan, Senior Specialist Technical Account Manager – Cloud Operations
- Chris Pates, Senior Specialist Technical Account Manager – Cloud Operations
- Erik Weber, Senior Specialist Solutions Architect – Operations Management
- Pragya Purohit, Specialist Solutions Architect – Cloud Foundations
- Rodrigue Koffi, Specialist Solutions Architect – Observability
- Saša Baškarada, Worldwide Lead AWS CAF
- Steven Tyson - Senior Specialist Technical Account Manager – Cloud Operations

Further reading

For additional information, refer to:

- Building dashboards for operational visibility
- Operational Readiness Reviews (ORR)
- Instrumenting distributed systems for operational visibility
- Change Management in the Cloud – AWS Whitepaper
- *Disaster Recovery of Workloads on AWS: Recovery in the Cloud* (AWS whitepaper)
- AWS Architecture Center
- AWS Case Studies
- AWS Cloud Adoption Framework
- AWS Cloud Adoption Framework (eBook)
- AWS General Reference
- AWS Knowledge Center
- AWS Prescriptive Guidance
- AWS Quick Starts: Partner solutions
- AWS Security Documentation
- AWS Solutions Library
- AWS Training and Certification
- AWS Well-Architected
- AWS Whitepapers & Guides
- Getting Started with AWS
- Overview of Amazon Web Services

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Minor revision (p. 32)	Updated guide to align with the IAM best practices. For more information, see Security best practices in IAM.	February 9, 2023
Document updated (p. 32)	Major updates.	November 9, 2022
Initial publication (p. 32)	Whitepaper published.	August 1, 2016

Notices

AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.