CHAPTER 12

Management & Governance

Introduction

Managing cloud environments requires robust governance and precise oversight to maintain performance, security, and scalability. AWS services are designed for management and governance. From real-time monitoring with Amazon CloudWatch to infrastructure orchestration through AWS CloudFormation, this chapter covers foundational tools that equip organizations with control and operational efficiency within AWS.

Structure

This chapter will cover the following topics:

* Amazon CloudWatch
* Amazon Managed Grafana
* Amazon Managed Service for Prometheus
* AWS Chatbot
* ChatOps for AWS
* AWS CloudFormation
* AWS CloudTrail
* AWS command line interface
* AWS compute optimizer
* AWS config
* AWS control tower
* AWS distro for OpenTelemetry
* AWS launch wizard
* Easily size, configure, and deploy third party applications on
* AWS license manager
* AWS managed services
* AWS management console mobile application
* AWS OpsWorks
* AWS organizations
* AWS personal health dashboard
* AWS proton
* AWS resilience hub
* AWS service catalog
* AWS service management connector
* AWS systems manager
* AWS trusted advisor
* AWS well-architected tool

Amazon CloudWatch overview

Amazon CloudWatch is a central monitoring service within AWS that enables real-time collection, analysis, and response to operational data. It helps businesses gain insights into their AWS infrastructure, improve performance, and address issues proactively.

Core functionality

CloudWatch provides a platform to track AWS resources, applications, and services. Users can collect metrics, access logs, and set up alarms, gaining real-time visibility into resource utilization and operational health.

Key features and Use Cases

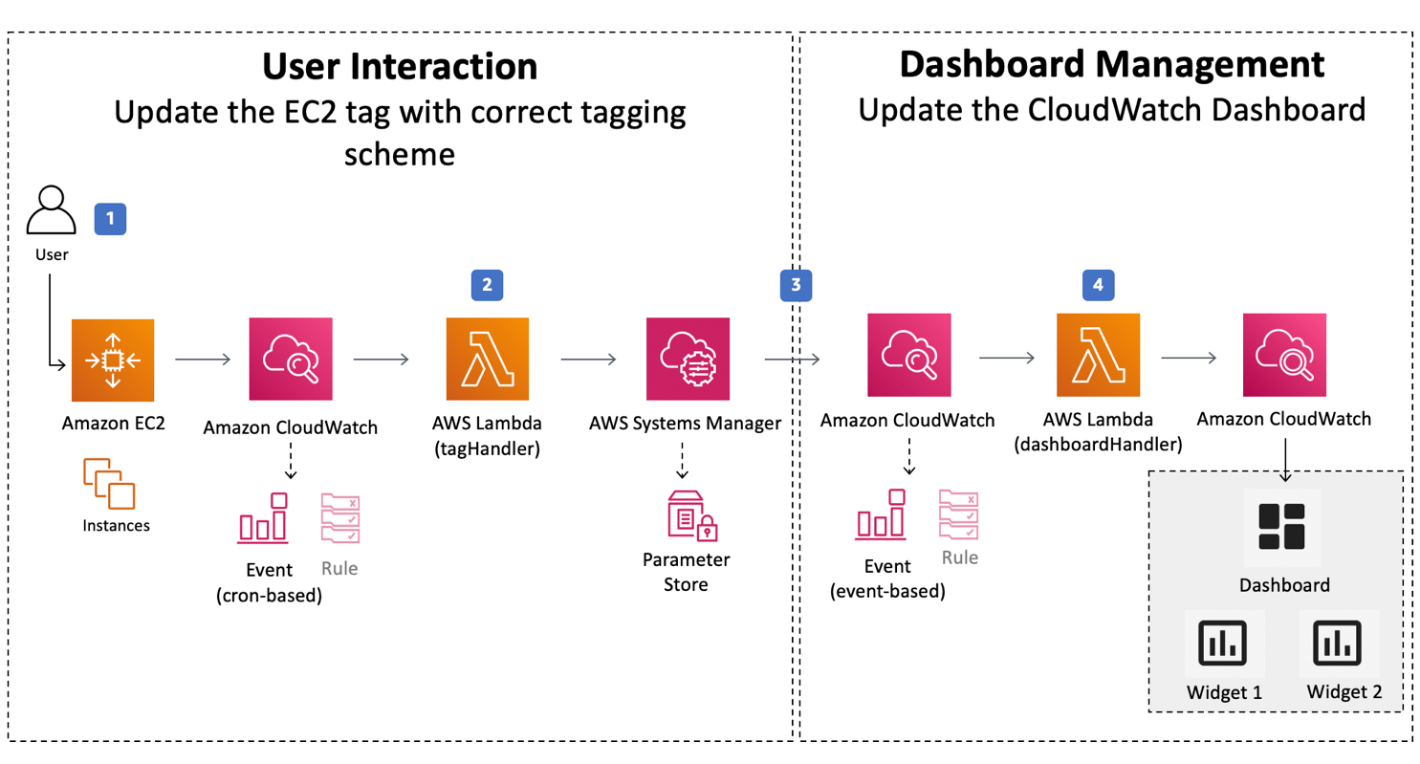
CloudWatch monitors key AWS components, including EC2 instances, Auto Scaling groups, and Elastic Load Balancers (ELBs). It helps businesses optimize performance, manage costs, and enhance operational efficiency.

Deployment strategies

To maximize CloudWatch's potential, businesses should use structured deployment strategies. This includes setting up custom dashboards, configuring alarms, utilizing CloudWatch Logs for troubleshooting, and optimizing overall monitoring and operations.

The accompanying diagram illustrates the flow of metrics and logs from AWS resources to Amazon CloudWatch, highlighting its role in real-time monitoring and alerting.

The diagram below illustrates how metrics and logs flow from various AWS resources to Amazon CloudWatch, emphasizing its role in real-time monitoring, alerting, and visualization.



**Figure 12.1:** Architecture of a Typical Monitoring Setup Using Amazon CloudWatch

Amazon Managed Grafana

Amazon Managed Grafana offers a centralized platform for visualizing operational data, making it easier for organizations to manage and analyze cloud resources. This fully managed service streamlines deployment, scales effortlessly, and reduces operational overhead. IT teams use it to create interactive dashboards that deliver actionable insights.

Security and compliance

Security is a top priority with Amazon Managed Grafana. It includes features like encryption, access controls, and AWS **Identity and Access Management** (**IAM**) integration. Organizations can deploy Managed Grafana confidently while adhering to compliance standards.

The illustration below shows the seamless integration of Amazon Managed Grafana with various AWS data sources, emphasizing its role in creating customizable dashboards for data visualization and analytics.

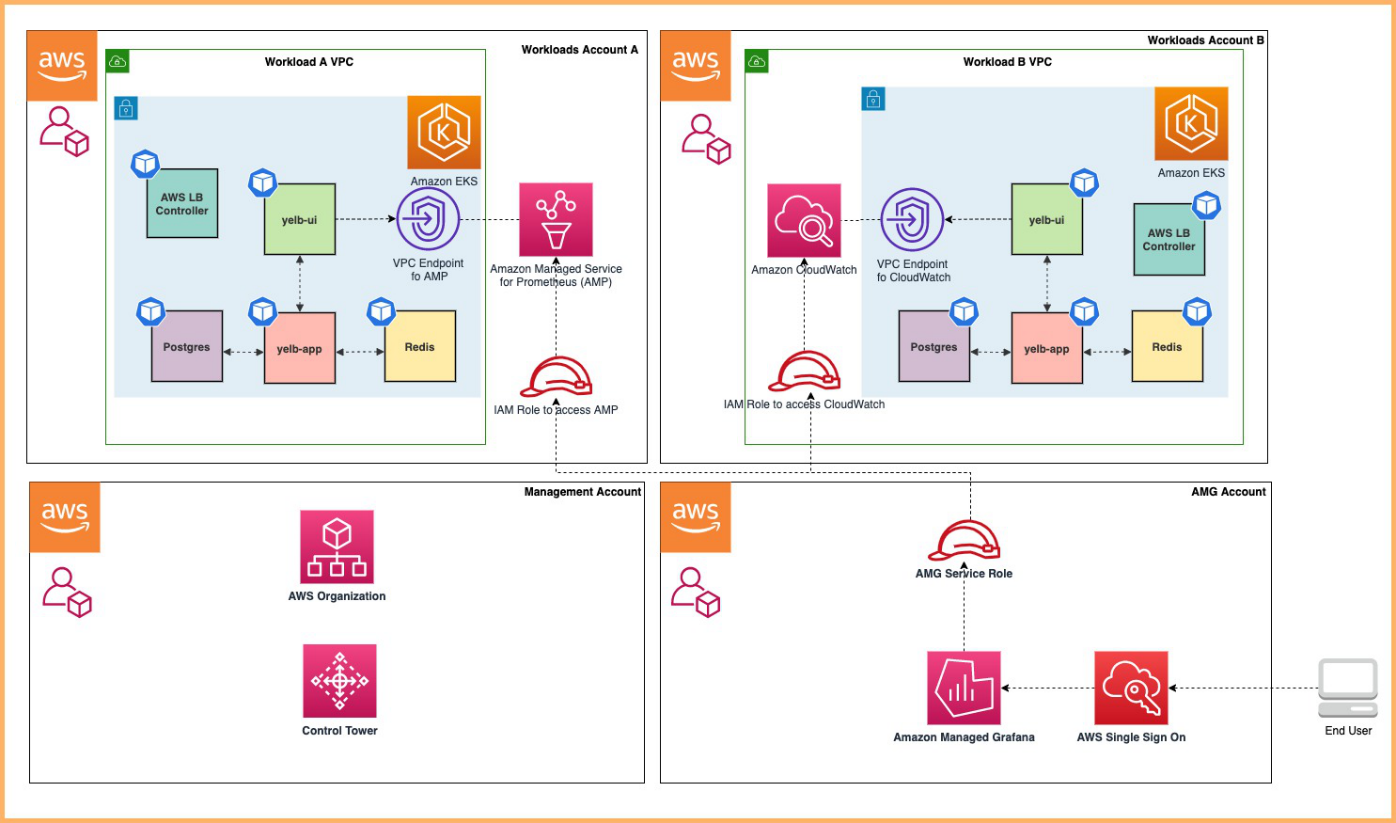


Figure 2 Typical Deployment Architecture of Amazon Managed Grafana

Deployment strategies

Organizations can improve Amazon Managed Grafana by implementing structured deployment strategies. Best practices include configuring data sources, designing dashboards, and enhancing performance to boost visibility and decision-making.

Amazon Managed Service for Prometheus

**Amazon Managed Service for Prometheus** (**AMP**) offers a scalable monitoring solution for cloud environments. As businesses adopt cloud technologies, efficient monitoring and alerting tools are essential for maintaining stability. AMP simplifies Prometheus deployment, providing real-time observability for AWS infrastructure.

Key capabilities and integration

AMP facilitates easy metric collection and analysis, integrating with AWS services like Amazon CloudWatch and Amazon Managed Grafana. It helps monitor system performance, ensuring high availability and optimal resource use.

Applications and best practices

AMP supports various use cases, including application performance monitoring, resource optimization, and compliance tracking. AWS documentation and whitepapers, such as *Enhance Observability with Amazon Managed Service for Prometheus*, offer best practices for maximizing AMP's capabilities.

The illustration below provides a visual overview of the key components and interactions within the AMP ecosystem, showcasing its seamless integration with AWS resources, applications, and the Prometheus toolkit for enhanced observability in the cloud.

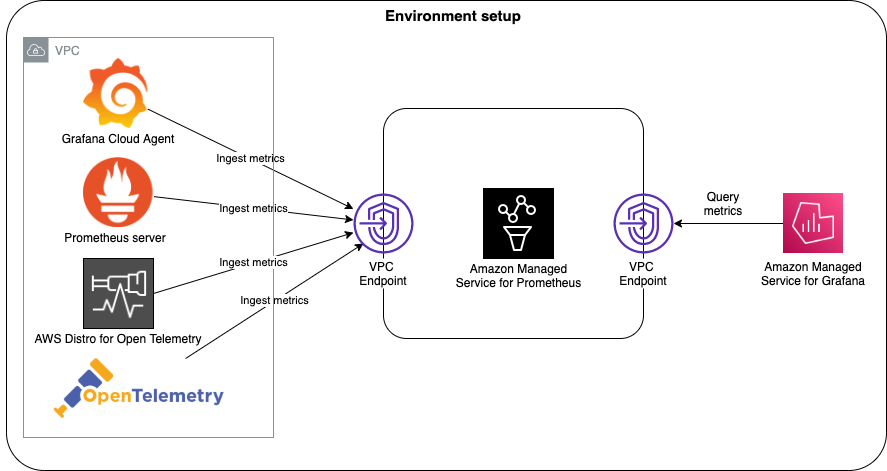


Figure 3 Amazon Managed Service for Prometheus (AMP) Ecosystem

Business benefits

AMP boosts operational efficiency by enabling quick issue detection and resolution. With real-time insights, IT teams can reduce risks, minimize downtime, and align monitoring with industry best practices.

AWS Chatbot

Effective communication is crucial for cloud governance. AWS Chatbot enhances operational awareness and response by integrating with collaboration platforms. It allows teams to receive real-time alerts, execute commands, and manage AWS resources directly in chat tools like Slack and Amazon Chime.

Use cases and best practices

Integrating AWS Chatbot with Amazon CloudWatch automates alert delivery and system notifications. For more strategies on optimizing AWS Chatbot, refer to the whitepaper *Effective Cloud Management with AWS Chatbot.*

Figure 4: AWS Chatbot Integration This illustration highlights AWS Chatbot's role in managing cloud alerts and executing AWS commands through chat-based interfaces.

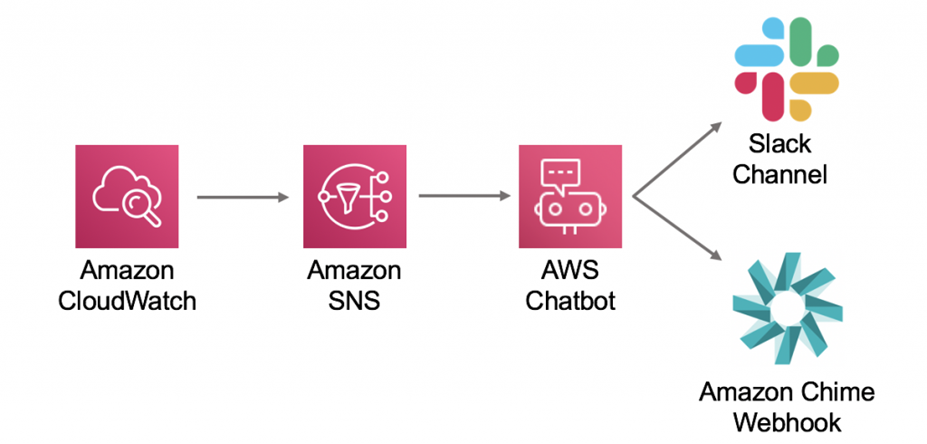


Figure 4 AWS Chatbot in Action

Business impact and future adoption

AWS Chatbot centralizes AWS notifications in workplace chat apps, improving team collaboration and operational efficiency. It enhances cloud management workflows by providing real-time alerts and enabling command execution.

AWS Services for cloud management

AWS provides various cloud management services, including Amazon CloudWatch for monitoring and AWS Chatbot for collaboration. These tools help organizations optimize performance, enhance security, and ensure regulatory compliance, giving them greater control over their AWS environments.

In summary, AWS Chatbot is essential for cloud management, acting as a communication hub that improves collaboration and responsiveness in AWS environments.

ChatOps for AWS

ChatOps integrates communication tools with AWS services, allowing teams to collaborate, execute commands, and receive alerts in real time.

Key features and integration

By using AWS Chatbot with tools like Slack and Amazon Chime, ChatOps enables automated responses and command execution. Incorporating AWS Lambda into these workflows automates routine tasks, increasing efficiency in managing AWS resources.

Business Impact

Adopting ChatOps simplifies communication and task execution, leading to faster response times and improved operational efficiency. This supports the goal of a more collaborative cloud management environment.

Applications

Businesses leveraging ChatOps report benefits such as quicker incident response, faster troubleshooting, reduced downtime, and better collaboration among teams.

In summary, ChatOps effectively merges communication and operations in AWS environments, streamlining workflows and enhancing cloud resource management. Embracing this approach leads to a more agile and responsive cloud management strategy.

AWS CloudFormation

AWS CloudFormation helps organizations manage infrastructure as code (IaC) by automating the provisioning and management of AWS resources. Users define infrastructure using declarative templates, which ensures consistency and scalability. Key concepts include stacks, templates, and change sets to automate deployments and reduce manual work.

Best practices

AWS documentation offers thorough guidance on CloudFormation implementation. The book *AWS CloudFormation: Getting Started Guide* by Mitch Tulloch provides template design strategies and advanced use cases, making it valuable for both beginners and experienced users. Exploring AWS documentation for real-world examples, reference architectures, and case studies enhances understanding and showcases effective CloudFormation usage.

Business advantages

CloudFormation streamlines deployment processes, maintains infrastructure consistency, and reduces configuration drift. By automating infrastructure provisioning, it aligns with business goals of efficiency, cost control, and agility. Adopting CloudFormation leads to improved resource governance, better cost management, and quicker time-to-market for applications.

Use cases

AWS offers reference architectures and sample templates demonstrating CloudFormation's capabilities across industries. These resources support organizations in implementing scalable and secure cloud deployments.

In conclusion, AWS CloudFormation is crucial for managing infrastructure as code in AWS. It equips users with the knowledge to utilize CloudFormation effectively, highlighting its foundations, validation, examples, and business benefits.

AWS CloudTrail

AWS CloudTrail provides capabilities for tracking activity, monitoring security, and ensuring compliance in AWS environments. It logs and analyzes API calls, helping organizations detect anomalies, investigate security events, and maintain an audit trail of resource interactions.

Key features

CloudTrail records API activity and changes to AWS resources, storing logs in Amazon S3 for analysis. It supports multi-region logging, data event tracking, and log file validation to ensure accuracy. This detailed logging is essential for enhancing security and compliance.

Integration with AWS services

CloudTrail integrates seamlessly with other AWS services to improve governance. It pairs with Amazon CloudWatch for real-time alerts, AWS Security Hub for security insights, and AWS Lambda for automated responses to anomalies. This creates a comprehensive monitoring and response ecosystem.

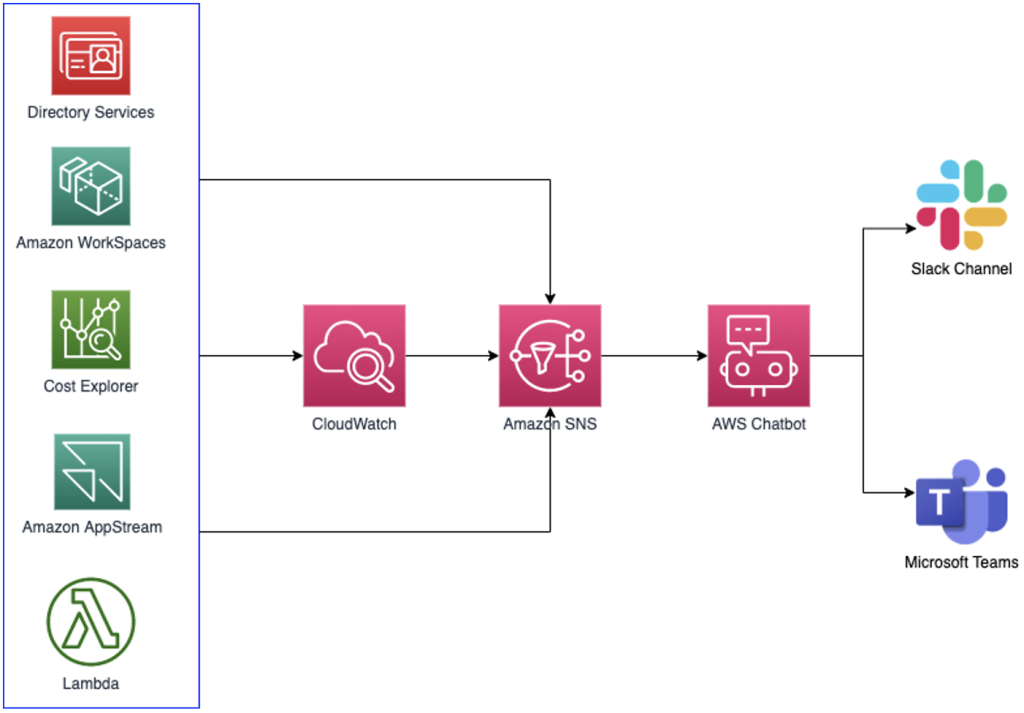
Deployment strategies

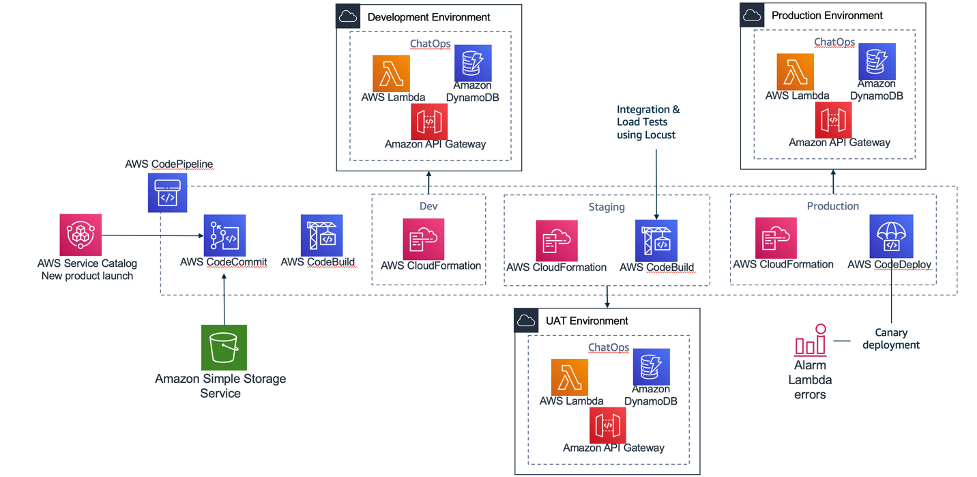
Organizations should configure CloudTrail for optimal security monitoring, compliance tracking, and forensic analysis. Best practices involve enabling multi-region trails, integrating with Amazon CloudWatch, and encrypting logs with AWS **Key Management Service** (**KMS**). Regular log reviews help identify unauthorized access attempts.

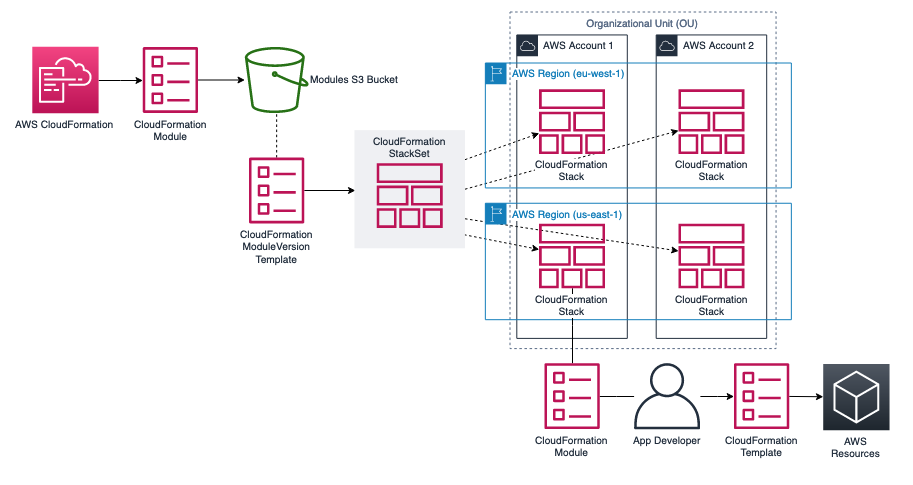
Use cases for AWS CloudTrail

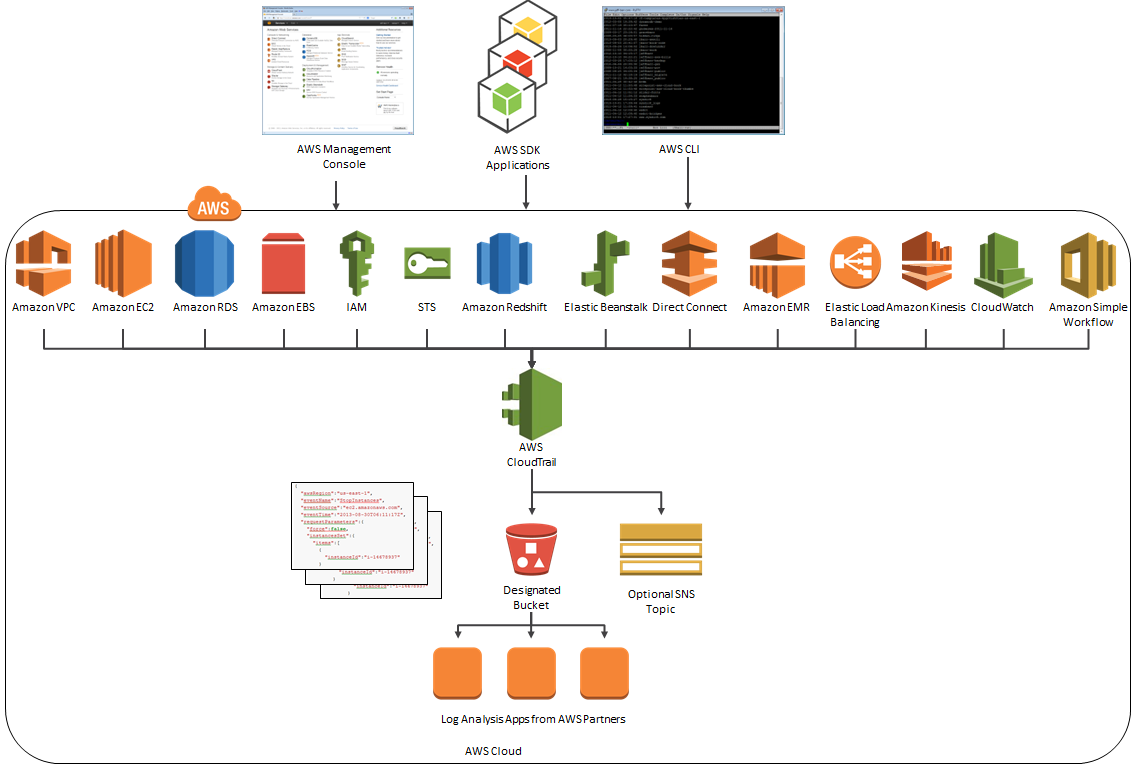
AWS CloudTrail is utilized for security incident response, regulatory compliance tracking, and operational troubleshooting. It enables organizations to monitor changes, ensure accountability, and maintain an auditable history of actions within AWS environments. The diagram illustrates how AWS CloudTrail captures events from AWS services and securely stores them in Amazon S3.

AWS CloudTrail is valuable for security analysis, resource change tracking, and compliance auditing. It tracks API calls, logs changes to resources, and delivers logs to Amazon S3. These logs are critical for demonstrating compliance with regulations. Additionally, CloudTrail helps organizations identify and respond to security incidents. It can be configured to meet specific security and compliance needs, fostering a robust governance framework.









AWS command line interface

The AWS **Command Line Interface** (**CLI**) serves as a unified tool for interacting with various AWS services from the command line. It provides administrators and developers with efficient access to AWS services, supports automated infrastructure management, and streamlines operational workflows. The CLI is designed to offer consistent experience across multiple AWS services, which reduces the learning curve for users.

Key features

AWS CLI offers a unified interface for command-line interactions with AWS services. It simplifies tasks, such as launching EC2 instances, configuring Amazon S3 buckets, and managing AWS Lambda functions. The CLI supports profile-based authentication, automation through scripting, and integration with AWS SDKs.

Integration with AWS services

The AWS CLI integrates with AWS CloudFormation, Amazon S3, and AWS Systems Manager, enabling effective management of infrastructure, automated deployments, and command execution across multiple environments. It also interacts with Amazon DynamoDB, AWS CloudFormation, and AWS IAM.

Deployment strategies

Organizations should configure the AWS CLI with credential profiles, automate repetitive tasks through shell scripts, and enforce security best practices with IAM policies. Using AWS CLI alongside infrastructure-as-code tools like AWS CloudFormation enhances automation and consistency.

Use cases

Organizations use the AWS CLI for infrastructure provisioning, automated backups, security audits, and continuous deployment pipelines. It enables engineers to streamline cloud management tasks, reducing manual efforts and enhancing operational efficiency.

AWS compute optimizer

AWS Compute Optimizer analyzes cloud workloads and offers recommendations for optimizing compute resources, improving cost efficiency and performance. Resource optimization is essential for operational efficiency and cost-effectiveness in cloud management. AWS Compute Optimizer provides important insights and recommendations for optimizing compute resources.

Key features

This machine learning-powered service evaluates EC2 instances, Auto Scaling groups, and AWS Lambda functions to recommend optimal configurations. Recommendations focus on reducing over-provisioning, improving resource utilization, and ensuring optimal performance-cost balance. By analyzing resource usage patterns, the Compute Optimizer identifies opportunities for improvement, allowing users to balance performance and cost.

Integration with AWS services

AWS Compute Optimizer integrates with Amazon CloudWatch for monitoring resource utilization and AWS Cost Explorer for cost analysis, helping organizations align performance optimization with financial efficiency.

Deployment strategies

Organizations should regularly review Compute Optimizer recommendations to adjust instance sizes, modify Auto Scaling policies, and optimize workloads. Integrating Compute Optimizer with Amazon CloudWatch enables teams to gain insights into resource performance, facilitating informed scaling decisions.

Use cases

Common use cases for AWS Compute Optimizer include optimizing EC2 workloads, reducing cloud spending through instance rightsizing, and improving application performance. The Compute Optimizer helps organizations enhance resource efficiency by aligning compute resources with actual demand, empowering informed decisions on resizing and changing instances.

AWS Config

AWS Config provides continuous oversight of resource configurations, ensuring compliance and security in cloud infrastructure. This fully managed service tracks changes, evaluates compliance, and automates corrective actions to maintain governance.

Key features

AWS Config continuously records configuration changes, offers an inventory of AWS resources, and enables automated compliance checks through AWS Config Rules, ensuring organizations adhere to best practices and regulations.

Integration with AWS services

AWS Config integrates with AWS CloudTrail for auditing, AWS Security Hub for centralized security management, and AWS Lambda for automated remediation of non-compliant resources. This integration facilitates effective governance by linking configuration monitoring with security controls.

Deployment strategies

Organizations should define custom AWS Config Rules, enable periodic compliance assessments, and integrate with AWS Systems Manager for operational control. Automating remediation with AWS Lambda ensures real-time policy enforcement.

Use cases

AWS Config is commonly used for security compliance audits, troubleshooting misconfigurations, and monitoring infrastructure drift. It helps organizations maintain control over resource configurations and quickly remediate non-compliance, enhancing security and governance.

AWS control tower

AWS Control Tower simplifies the management of multiple AWS accounts by providing a centralized service that automates account setup, enforces policies, and ensures compliance.

Key features

It offers an automated Landing Zone for multi-account setups, Service Control Policies (SCPs) for security guideline enforcement, and pre-configured guardrails for compliance management.

Integration with AWS services

AWS Control Tower integrates with AWS Organizations for central governance, AWS Single Sign-On for user authentication, and AWS Security Hub for security compliance monitoring.

Deployment strategies

Leverage the Account Factory to automate new account creation, customize SCPs to fit internal policies, and integrate tools like AWS Config for ongoing compliance validation, enhancing overall governance.

Use cases

AWS Control Tower helps organizations manage complex multi-account environments by ensuring centralized compliance, enforcing security, and streamlining account provisioning. It enables scaling operations while maintaining regulatory alignment and best security practices.

Business implications

Deploying AWS Control Tower improves governance, enhances security, and boosts operational efficiency. Its centralized management reduces complexity, allowing organizations to focus on innovation and growth.

AWS Distro for OpenTelemetry

**AWS Distro for OpenTelemetry** (**ADOT**) is essential for gaining visibility into distributed systems' performance. It standardizes observability data collection, providing real-time insights into application behavior and infrastructure performance.

ADOT is an open-source version of the OpenTelemetry project, designed for cloud-native environments. It generates, collects, and correlates traces and metrics, offering insights into application interactions.

Key features

ADOT supports application tracing and metric collection across various programming languages and integrates with OpenTelemetry SDKs. The OpenTelemetry Collector aggregates, processes, and forwards telemetry data.

Integration with AWS services

ADOT integrates with AWS X-Ray for distributed tracing, Amazon CloudWatch for metrics, and AWS Lambda for serverless a monitoring. This unified solution enhances performance diagnostics and troubleshooting, giving businesses comprehensive insights into their AWS workloads.

Deployment strategies

Organizations should use ADOT with existing monitoring frameworks to track distributed transactions and identify latency issues. Establishing clear observability goals and optimizing tracing configurations will enhance effectiveness.

Common use cases

Typical use cases for ADOT include application performance monitoring, real-time analytics for microservices, and improved DevOps workflows. By adopting OpenTelemetry, organizations can monitor request flows, detect anomalies, and proactively resolve issues to improve user experience.

Business implications

Implementing AWS Distro for OpenTelemetry enhances observability and decision-making. Detailed performance insights enable organizations to address issues proactively, optimize resource usage, and improve user experience.

ADOT standardizes observability practices across applications and services. Adopting open standards ensures compatibility with other tools, simplifying the management of distributed systems.

AWS launch wizard

AWS Launch Wizard simplifies and automates application deployment on AWS, ensuring adherence to best practices for security, performance, and cost efficiency. It streamlines the provisioning of infrastructure, making it essential for effective cloud operations.

Key features

* **Guided workflows\*\*:** Offers workflows for deploying Microsoft SQL Server, SAP, and other enterprise applications.
* **Automation\*\*:** Automates infrastructure sizing, configuration, and provisioning to reduce complexity and optimize deployments.

Integration with AWS services

Integrates with AWS CloudFormation for customization, AWS Systems Manager for operational management, and AWS IAM for role-based access control. This integration creates a comprehensive strategy for efficient application infrastructure management.

Deployment strategies

Organizations should use AWS Launch Wizard for structured deployments. Custom deployment parameters can align with security policies, and monitoring solutions can further enhance outcomes.

Use cases

AWS Launch Wizard is ideal for deploying database-driven applications and SAP environments. It accelerates rollout while ensuring scalability, resilience, and compliance.

Business implications

Adopting AWS Launch Wizard results in significant time and resource savings. Automation minimizes deployment errors, speeds up time-to-market, and allows IT teams to focus on strategic initiatives.

AWS CloudTrail: Enhancing visibility and security

AWS CloudTrail provides essential logging and monitoring for AWS accounts, improving governance and compliance.

Overview

CloudTrail records AWS API calls, capturing key information like the caller's identity, timestamps, and request parameters. This functionality supports operational and risk auditing for AWS accounts.

Key features

CloudTrail offers a comprehensive view of account activity, recording changes to resources and identifying who made changes and when.

Integration with AWS services

Integrates with Amazon S3 and AWS CloudWatch for secure log storage and alerts based on specific events. This integration enhances monitoring and accountability within AWS environments.

Security and compliance

AWS CloudTrail enhances security and ensures compliance by detecting unusual activity, aiding troubleshooting, and meeting regulatory requirements. It records API calls, which helps trace actions in an AWS environment.

Business implications

Adopting AWS CloudTrail increases transparency, accountability, and security for organizations. CloudTrail logs are useful for forensic analysis, compliance reporting, and auditing, contributing to a more secure AWS environment aligned with industry’s best practices.

Further reading

For a deeper understanding of AWS CloudTrail, refer to the official AWS documentation. Internal references provide technical insights, while external sources like case studies offer real-world examples of security implementations using AWS CloudTrail.

AWS Managed Services: Streamlining operations

AWS AMS simplifies and streamlines cloud operations for users by offloading daily management tasks, allowing organizations to focus on innovation. AMS supports a wide range of AWS services, providing a holistic management solution.

Key features

AMS automates operational tasks such as change requests, monitoring, patch management, and backup services. It includes components like Control Tower and Landing Zones, enhancing understanding of its architecture.

Integration with AWS services

AMS integrates seamlessly with various AWS services, including AWS Organizations, AWS Config, and AWS CloudTrail, improving governance, security, and compliance.

Security and compliance

AMS prioritizes security and compliance through automated security practices, regular patching, and adherence to the AWS Well-Architected Framework.

Business implications

Implementing AWS Managed Services leads to operational efficiency, scalability, and improved governance. It shifts the focus from routine tasks to strategic initiatives, enhancing business agility and competitiveness.

Further reading

For a comprehensive understanding of AWS Managed Services, consult the official AWS documentation. Internal references offer technical insights, while external sources provide perspectives on successful AMS implementations.

AWS management console mobile application

The AWS Management Console Mobile Application allows users to manage their AWS resources from anywhere.

Overview and core functionalities

This mobile application extends the AWS Management Console experience to devices, enabling users to monitor resource health, respond to alarms, and access critical operational metrics.

Key features and interface\*\*

The AWS Management Console Mobile Application provides an intuitive interface for viewing and interacting with AWS resources. Key features include real-time status monitoring, easy resource navigation, and command execution, offering a clear understanding of the application's capabilities.

Security considerations

Security is crucial for mobile usage. This section covers the security measures in the AWS Management Console Mobile Application, such as multi-factor authentication (MFA), data encryption in transit, and best practices for securing AWS resource access on mobile devices.

Practical benefits and use cases

The mobile application offers significant benefits for users needing access to AWS resources on the go. Real-world use cases demonstrate how it improves operational efficiency, speeds up issue resolution, and aids quick decision-making.

Integration with AWS services

The application integrates seamlessly with various AWS services. Users can interact with services like Amazon EC2, Amazon S3, and AWS Lambda through the app, enhancing overall management experience.

Business implications

From a business perspective, the AWS Management Console Mobile Application boosts operational agility and responsiveness. It helps AWS users stay connected and manage resources effectively, even when away from their desks.

Further reading

For more insights into the AWS Management Console Mobile Application, readers are encouraged to explore the AWS documentation and user testimonials to gain a better understanding of its usability and benefits.

AWS OpsWorks: Streamlined application management

AWS OpsWorks simplifies the deployment and management of cloud applications. This section outlines the features, benefits, and practical aspects of using AWS OpsWorks.

Overview and core functionality

AWS OpsWorks automates code deployment, infrastructure configuration, and application lifecycle management. It allows users to define application architecture and deploy across multiple instances efficiently.

Key features and components

Gain insights into AWS OpsWorks' key features, including stacks, layers, and instances, which help in creating scalable and maintainable application architectures.

Deployment strategies and flexibility

Explore various deployment strategies supported by AWS OpsWorks, such as time-based and load-based deployments. OpsWorks also offers flexibility with custom Chef and Puppet recipes for application customization.

Integration with other AWS services

OpsWorks integrates smoothly with other AWS services, enhancing its functionality. This section explains how it connects with services like Amazon RDS, Elastic Load Balancing, and Auto Scaling.

Security considerations

Security is vital in application management. This section addresses AWS OpsWorks' security features, including IAM integration, encryption, and securing sensitive information.

Practical benefits and use cases

Explore the practical benefits and use cases of AWS OpsWorks, from automating tasks to managing complex applications, highlighting its efficiency and scalability.

Business implications

Using AWS OpsWorks leads to operational efficiency, cost savings, and reliability. This section discusses how it aligns with business objectives for consistent application deployment.

Further reading

For deeper insights into AWS OpsWorks, readers are encouraged to consult AWS resources and documentation.

AWS organizations: Streamlining Cloud Resource Management

In the realm of cloud management and governance, AWS Organizations appears as a cornerstone service, providing a comprehensive framework for structuring and managing the AWS environment. This section delves into the features, benefits, and strategic considerations of AWS Organizations for effective resource management in cloud infrastructures.

Overview and core functionality

AWS Organizations simplify the complexity of managing multiple AWS accounts within an organization[[1]](#footnote-1). This section offers an in-depth overview of how Organizations provide a hierarchical and scalable structure, enabling administrators to manage permissions, apply policies, and combine billing across accounts.

Key features and components

Readers will gain insights into the key features of AWS Organizations, including the **organizational units** (**OUs**), **service control policies** (**SCPs**), and the primary account. Understanding these components is crucial for designing an organizational structure that aligns with business needs and regulatory requirements.

Policy-based management

This section explores how AWS Organizations help policy-based management through SCPs. Readers will understand how SCPs help enforce security and compliance standards across the organization, ensuring consistent and controlled deployment of resources.

Consolidated billing and cost allocation

AWS Organizations simplifies billing and cost allocation by combining charges across all linked accounts. This section outlines the benefits of combined billing and how organizations can use this feature to gain insights into cost distribution and improve resource spending.

Strategic considerations for implementation

Strategic considerations are crucial for the successful implementation of AWS Organizations. This section offers practical guidance on structuring OUs, defining SCPs, and aligning the organizational hierarchy with business units, enhancing the agility and efficiency of resource management.

Integration with other AWS services

AWS Organizations seamlessly integrates with other AWS services, enhancing its capabilities. This section explores how Organizations can be integrated with AWS Identity and IAM, AWS **Single Sign-On** (**SSO**), and other services to create a unified and secure cloud environment.

Real-world deployment strategies

Readers will gain insights into real-world deployment strategies for AWS Organizations. Practical examples showcase how organizations can structure their accounts, apply policies, and use Organizations to efficiently manage resources, ensuring scalability and governance.

Business implications

From a business standpoint, AWS Organizations contribute to operational efficiency, cost management, and regulatory compliance. This section discusses how adopting Organizations aligns with business goals, fostering a streamlined and secure cloud environment.

Further reading

For readers seeking a deeper understanding of AWS Organizations, exploring AWS documentation[[2]](#footnote-2) is recommended. Internal references offer technical insights, while external references, such as case studies and best practice guides, offer practical perspectives on improving Organizations for specific use cases.

AWS Personal health dashboard: Proactive monitoring for operational excellence

In the landscape of cloud management and governance, the AWS **Personal Health Dashboard** (**PHD**) serves as a crucial tool, offering real-time insights into the operational status of AWS resources. This section navigates through the features, benefits, and practical applications of the AWS Personal Health Dashboard, offering readers a comprehensive understanding of its role in ensuring operational excellence.

Overview and core functionality

The AWS Personal Health Dashboard is a centralized service that offers a personalized view of the health of AWS resources[[3]](#footnote-3). This section provides an in-depth overview of how PHD aggregates information from multiple AWS services, presenting it in a combined and easily accessible format for users.

Key features and components

Readers will gain insights into the key features of the AWS Personal Health Dashboard, including the ability to view alerts, remediation guidance, and integration with AWS Support. Understanding these components is essential for leveraging PHD effectively in monitoring and keeping the health of AWS resources.

Proactive monitoring and alerting

This section explores how the AWS Personal Health Dashboard helps with proactive monitoring by providing real-time alerts about events that may affect AWS resources. Readers will understand how to interpret these alerts and take preemptive actions to ensure the continuity of operations.

Integration with AWS support

AWS PHD seamlessly integrates with AWS Support, enhancing its capabilities. This section discusses how users can use integration to access detailed support information, including status updates, upcoming maintenance events, and proactive recommendations.

Use cases and practical applications

Real-world use cases illustrate how organizations can receive help from the AWS Personal Health Dashboard. Examples include identifying and mitigating issues before they affect operations, minimizing downtime, and improving resource use for enhanced efficiency.

Customization and user preferences

PHD allows users to customize their dashboard based on their preferences and priorities. This section guides readers through the process of tailoring alerts, setting preferences, and configuring notifications, ensuring a personalized and efficient monitoring experience.

Strategic considerations for implementation

Strategic considerations are vital for the effective implementation of the AWS Personal Health Dashboard. This section offers practical guidance on aligning PHD with organizational goals, setting up custom alerts, and integrating them into existing monitoring workflows.

Business implications

From a business standpoint, the AWS Personal Health Dashboard contributes to operational resilience, minimizing the impact of potential issues on business operations. This section discusses how proactive monitoring with PHD aligns with business continuity and risk management strategies.

Further reading

For readers seeking a deeper understanding of AWS Personal Health Dashboard, exploring AWS documentation[[4]](#footnote-4) is recommended. Internal references offer technical insights, while external references, such as case studies and best practice guides, offer practical perspectives on maximizing the benefits of PHD.

AWS Proton: Streamlining container and serverless application deployment

In the dynamic realm of cloud management and governance, efficient application deployment is a cornerstone for operational success. AWS Proton appears as a comprehensive service, streamlining the deployment and management of containerized and serverless applications. This section provides a detailed exploration of AWS Proton, shedding light on its functionalities, benefits, and strategic implications for businesses.

Overview and core functionality

AWS Proton is introduced as a fully managed service designed to automate and streamline the deployment of containerized and serverless applications[[5]](#footnote-5). This section delves into the fundamental principles and core functionalities that underpin Proton's role in simplifying the application deployment lifecycle.

Key features and components

Readers gain insights into the key features and components of AWS Proton, including its ability to create and manage environments, define service templates, and help collaboration among development teams. Understanding these elements is crucial for harnessing Proton's capabilities effectively.

Service templates and environments

This section explores the concept of service templates in Proton, elucidating how they serve as blueprints for defining the infrastructure and configurations of applications. Additionally, the management of environments is discussed, emphasizing how Proton ensures consistency across different deployment stages.

Collaboration and governance

Proton is positioned as a collaborative tool for development teams. The section outlines how Proton helps collaboration by providing a centralized platform for defining, managing, and sharing service templates. Governance mechanisms within Proton to ensure adherence to organizational policies are also discussed.

Proton in action: Deployment workflows

Real-world examples and use cases prove how AWS Proton runs in practical scenarios. The section walks through deployment workflows, showcasing how Proton automates the provisioning of infrastructure, deployment of code, and management of application configurations.

Integration with AWS services

AWS Proton seamlessly integrates with other AWS services, enhancing its capabilities. This section explores how Proton integrates with AWS Identity and Access Management (IAM), AWS CloudFormation, and AWS CodePipeline, providing readers with a holistic understanding of its interoperability within the AWS ecosystem.

Best practices for proton implementation

Strategic considerations for implementing AWS Proton are discussed, offering readers guidance on improving Proton for their specific use cases. Best practices cover service template design, environment management, and using Proton for both containerized and serverless applications.

Business implications

From a business perspective, the deployment efficiency brought by AWS Proton translates into accelerated time-to-market for applications. This section explores how Proton aligns with business goals, fostering agility, reducing manual intervention, and minimizing deployment-related challenges.

Further reading

To delve deeper into AWS Proton's technical intricacies and best practices, readers are encouraged to explore AWS documentation[[6]](#footnote-6). Internal references provide insights into Proton's features, while external references, such as case studies and success stories, offer practical perspectives on its real-world implementation.

AWS Resilience hub: Strengthening cloud infrastructures

In the dynamic landscape of cloud management and governance, ensuring the resilience of cloud infrastructures is paramount. The AWS Resilience Hub appears as a critical component, providing organizations with tools and insights to enhance their ability to withstand disruptions. This section offers an in-depth exploration of the AWS Resilience Hub, its features, and its role in fortifying cloud environments.

Introduction to AWS resilience hub

The AWS Resilience Hub is introduced as a comprehensive service designed to help organizations manage and improve the resilience of their applications in the cloud[[7]](#footnote-7). This section emphasizes the significance of resilience in the context of cloud infrastructure and sets the stage for a detailed exploration of the tool's capabilities.

Key features and functionality

The section delves into the key features of the AWS Resilience Hub, highlighting its ability to aggregate and visualize resilience-related information across multiple AWS accounts and regions. This includes discussions on the hub's role in centralizing insights related to disaster recovery, fault tolerance, and overall system stability.

Integration with AWS organizations

Insights are provided on how the AWS Resilience Hub seamlessly integrates with AWS Organizations, allowing organizations to extend their resilience management across an entire organization. This integration enhances the tool's scalability and adaptability to diverse cloud architectures.

Resilience dashboard

A detailed exploration of the Resilience Dashboard is undertaken, elucidating how it offers a centralized view of an organization's resilience posture. This includes discussions on the dashboard's capabilities for checking the health of critical resources, finding vulnerabilities, and ensuring compliance with resilience best practices.

Automated insights and recommendations

The section explores how the AWS Resilience Hub employs automated insights and recommendations to help organizations in showing areas for improvement. Practical examples illustrate how these automated features contribute to proactive resilience management.

Real-time monitoring and alerting

Readers gain insights into how the AWS Resilience Hub helps real-time monitoring and alerting, ensuring that organizations can respond swiftly to potential disruptions. This includes discussions on the tool's capabilities for detecting anomalies and providing actionable alerts.

Business continuity and disaster recovery

The role of the AWS Resilience Hub in supporting business continuity and disaster recovery strategies is emphasized. Case studies and practical examples illustrate how organizations use the tool to ensure uninterrupted operations in the face of unforeseen events.

Business implications

From a business standpoint, the deployment of AWS Resilience Hub is positioned as a strategic investment in long-term operational resilience. This section explores how organizations can derive business value by ensuring continuous availability and reliability of critical applications.

Further reading

Readers are encouraged to explore the AWS Resilience Hub documentation for a more comprehensive understanding of the tool's technical intricacies[[8]](#footnote-8). Internal references offer detailed insights, while external case studies and success stories offer practical perspectives on resilience management.

AWS service catalog: Standardizing and managing IT services

In the landscape of cloud management and governance, the AWS Service Catalog plays a pivotal role in providing organizations with a standardized and efficient approach to managing IT services. This section delves into the intricate details of the AWS Service Catalog, exploring its features, benefits, and strategic implications for businesses.

Overview and core functionality

AWS Service Catalog is introduced as a service that allows organizations to create, govern, and manage a catalog of IT services. This section provides a foundational understanding of the core functionalities that underpin the role of Service Catalog in IT service management[[9]](#footnote-9).

Key features and components

Readers are guided through the key features and components of the AWS Service Catalog, including the concept of portfolios, products, and constraints. Understanding these elements is crucial for setting up a well-organized and governed catalog of IT services.

Portfolios and products

This section explores how the AWS Service Catalog allows organizations to group and manage related products through portfolios. It sheds light on the creation of portfolios to align with organizational structures and the inclusion of products within them.

Constraints and governance

AWS Service Catalog provides mechanisms for governance through the imposition of constraints. The section discusses how constraints can be defined to control the use of specific resources or configurations, ensuring compliance with organizational policies and standards.

Integration with AWS services

Service Catalog seamlessly integrates with other AWS services. This section explores how Service Catalog interacts with AWS IAM, AWS CloudFormation, and AWS Lambda, offering readers insights into its interoperability within the AWS ecosystem.

End-user experience

The end-user experience is a crucial aspect of AWS Service Catalog. This section details how end-users can browse and launch products from the catalog, emphasizing the self-service nature of Service Catalog and its role in empowering teams.

Lifecycle management

AWS Service Catalog helps the entire lifecycle of IT services. From the creation of products to versioning and updates, this section provides a comprehensive understanding of how Service Catalog ensures the smooth evolution of IT services over time.

Best practices for implementation

Strategic considerations for implementing AWS Service Catalog are discussed, providing readers with guidance on designing efficient catalogs, defining effective constraints, and improving the end-user experience.

Business implications

From a business perspective, AWS Service Catalog contributes to cost efficiency, standardization, and compliance. This section explores how Service Catalog aligns with business goals, offering a structured approach to IT service management that fosters consistency and governance.

Further reading

For a more profound insight into AWS Service Catalog's technical intricacies and best practices, readers are encouraged to explore AWS documentation[[10]](#footnote-10). Internal references provide insights into Service Catalog's features, while external references, such as case studies, offer practical perspectives on its real-world implementation.

AWS service management connector: Bridging the gap between AWS and ServiceNow

In the realm of cloud management and governance, the AWS Service Management Connector appears as a key facilitator, seamlessly integrating AWS services with the popular **IT Service Management** (**ITSM**) platform, ServiceNow. This section delves into the comprehensive features, integration capabilities, and strategic advantages that AWS Service Management Connector brings to organizations.

Overview and integration

AWS Service Management Connector is introduced as a bridge that connects AWS services and ServiceNow. This section provides a high-level overview of the purpose and significance of this connector in the context of IT service management[[11]](#footnote-11).

Seamless AWS ServiceNow integration

Readers are guided through the seamless integration capabilities of AWS Service Management Connector with ServiceNow. This includes the bi-directional communication that allows AWS resources to be provisioned, watched, and managed directly from the ServiceNow platform.

Key features and functionality

This section explores the key features of AWS Service Management Connector, emphasizing its ability to streamline workflows, enhance visibility, and improve collaboration between AWS and ServiceNow users. It highlights features such as incident management, change management, and service catalog integration.

Use cases and benefits

Real-world use cases and benefits are presented to illustrate how organizations can use AWS Service Management Connector to improve their IT service management processes. Examples may include faster incident resolution, improved change management, and enhanced reporting capabilities.

Governance and compliance

AWS Service Management Connector plays a crucial role in ensuring governance and compliance by providing a standardized and controlled way to interact with AWS resources from ServiceNow. This section delves into the governance features that support regulatory compliance and security best practices.

Implementation best practices

Strategic considerations for implementing AWS Service Management Connector are discussed, providing readers with guidance on configuration, setup, and ongoing management. Best practices for keeping secure and efficient integration are emphasized.

Business implications

From a business standpoint, AWS Service Management Connector contributes to operational efficiency, collaboration, and a unified approach to IT service management. This section explores how the connector aligns with business goals, fostering an integrated and streamlined ITSM environment.

Further reading

For readers seeking a deeper technical understanding, AWS documentation[[12]](#footnote-12) is recommended. Internal references shed light on the technical intricacies of AWS Service Management Connector, while external case studies and success stories offer practical insights into its real-world impact.

AWS systems manager: Orchestrating operational insights and management

In the intricate landscape of cloud management, AWS Systems Manager stands out as a versatile and powerful solution. This section explores the comprehensive capabilities of AWS Systems Manager, elucidating its role in orchestrating operational insights and management across AWS environments.

Overview and core functionality

AWS Systems Manager is introduced as a unified interface for centralizing operational data and automating tasks. This section provides an overview of its core functionality, emphasizing its role in simplifying resource and application management at scale[[13]](#footnote-13).

Operational insights and visibility

Readers are guided through how AWS Systems Manager provides operational insights and visibility into their infrastructure. The service's ability to aggregate and visualize data, including patch compliance, instance inventory, and operational health, is highlighted.

Key features and use cases

This section delves into the key features of AWS Systems Manager, including capabilities such as Run Command, State Manager, and Automation. Real-world use cases are presented to showcase how organizations can use these features for tasks like patch management, configuration management, and automated responses to operational issues.

Automation and orchestration

A detailed exploration of the automation and orchestration capabilities of AWS Systems Manager is provided. This includes insights into how automation workflows can be created to streamline operational tasks, ensuring consistency and efficiency in managing resources.

Security and compliance

AWS Systems Manager plays a pivotal role in keeping security and compliance by providing a centralized platform for managing and enforcing security policies. This section outlines the security features of the service, including the integration with AWS **Identity and Access Management** (**IAM**) and compliance reporting.

Integration with other AWS services

The seamless integration of AWS Systems Manager with other AWS services is explored. Examples may include integration with Amazon CloudWatch for enhanced monitoring and with AWS Identity and Access Management for secure access control.

Real-world deployment strategies

Practical deployment strategies are offered, providing guidance on best practices for configuring AWS Systems Manager to meet specific organizational needs. Insights into creating automation documents, managing patch baselines, and improving operational tasks are discussed.

Business implications

From a business standpoint, AWS Systems Manager contributes to operational efficiency, cost optimization, and enhanced security. This section explores how the service aligns with broader business goals, enabling organizations to keep a proactive and well-managed AWS environment.

Further reading

To delve deeper into the technical intricacies of AWS Systems Manager, readers are encouraged to explore AWS documentation[[14]](#footnote-14). Internal references offer detailed insights, while external case studies and best practices offer a broader understanding of the service's applications.

AWS trusted advisor: Proactive guidance for optimizing AWS environments

In the intricate landscape of AWS management and governance, AWS Trusted Advisor appears as a proactive and indispensable tool for improving AWS environments. This section delves into the comprehensive capabilities of AWS Trusted Advisor, providing insights into its role as a guidance system for best practices, cost optimization, and enhanced security.

Overview and core functionality

AWS Trusted Advisor is introduced as a cloud service that offers best practices and recommendations to improve AWS environments across various pillars, including cost optimization, performance, security, and reliability[[15]](#footnote-15). The section provides an overview of its core functionality, emphasizing its role in delivering actionable insights for improving AWS resources.

Best practices and recommendations

Readers are guided through the key best practices and recommendations provided by AWS Trusted Advisor. This includes insights into cost-saving opportunities, performance improvement suggestions, security enhancements, and reliability optimizations.

Pillars of evaluation

The section explores the pillars across which AWS Trusted Advisor evaluates AWS environments. This includes a deep dive into each pillar, offering practical examples of the types of checks performed and the corresponding recommendations provided.

Automation and integration

AWS Trusted Advisor's role in automating the identification of issues and the provision of recommendations is highlighted. Integration possibilities with AWS CloudWatch and AWS IAM are discussed, showcasing how organizations can use automation for a proactive approach to AWS management.

Cost optimization strategies

A detailed examination of cost optimization strategies recommended by AWS Trusted Advisor is provided. This includes insights into areas such as the rightsizing of instances, using reserved instances, and finding idle resources to streamline costs.

Security and compliance

The proactive security checks performed by AWS Trusted Advisor are explored. Readers gain insights into how the service shows potential security vulnerabilities and compliance gaps, enabling organizations to strengthen their security posture.

Real-world implementation

Practical implementation strategies are offered, guiding how organizations can effectively use AWS Trusted Advisor to enhance their AWS environments. This includes creating custom notification settings, implementing automated checks, and integrating recommendations into existing workflows.

Business implications

From a business standpoint, AWS Trusted Advisor contributes to cost efficiency, enhanced performance, and fortified security. This section explores how organizations can align their AWS management strategies with the recommendations provided by Trusted Advisor to achieve the best outcomes.

Further reading

To delve deeper into the technical intricacies of AWS Trusted Advisor, readers are encouraged to explore AWS documentation[[16]](#footnote-16). Internal references offer detailed insights, while external case studies and best practices offer a broader understanding of the service's applications.

AWS well-architected tool: Blueprint for building robust architectures

In the realm of AWS management and governance, the AWS Well-Architected Tool stands as a guiding blueprint for organizations aiming to build robust, scalable, and efficient cloud architecture. This section provides an in-depth exploration of the tool's functionalities, its significance in the cloud landscape, and how it contributes to the well-architected framework.

Introduction to AWS well-architected tool

The AWS Well-Architected Tool is introduced as a service that enables organizations to review and improve their cloud architecture by following the best practices recommended by AWS[[17]](#footnote-17). This section emphasizes the tool's role in aligning architecture with the five pillars of the Well-Architected Framework: operational excellence, security, reliability, performance efficiency, and cost optimization.

Pillar-specific evaluation

The section delves into each pillar of the Well-Architected Framework, providing insights into the specific considerations and best practices assessed by the tool. This includes discussions on operational practices, security measures, reliability strategies, performance optimization, and cost-saving opportunities.

Aligning with best practices

The AWS Well-Architected Tool guides readers in aligning cloud architecture with industry best practices. This includes a detailed examination of how the tool finds areas for improvement and provides actionable recommendations to enhance the overall architecture.

Risk mitigation and security

The tool's role in risk identification and mitigation explored, with a specific focus on security considerations. This section gives practical examples of how the Well-Architected Tool helps organizations enhance their security posture by finding vulnerabilities and suggesting remediation steps.

Performance optimization

Insights into how the Well-Architected Tool evaluates performance efficiency are provided. This includes discussions on resource use, load balancing, and other considerations aimed at improving the overall performance of cloud architecture.

Real-time evaluation

The section discusses the real-time evaluation capabilities of the Well-Architected Tool. This includes insights into how organizations can use the tool iteratively throughout the lifecycle of their applications to ensure continuous alignment with best practices.

Actionable recommendations

A detailed examination of the actionable recommendations provided by the tool is offered. This includes strategies for implementing changes based on the tool's insights to drive improvements in cloud architecture.

Business implications

From a business perspective, the AWS Well-Architected Tool is positioned as a proactive measure to ensure that cloud architecture is not only functional but also improved for long-term success. This section explores how adherence to the Well-Architected Framework contributes to organizational resilience and agility.

Further reading

Readers are encouraged to explore AWS documentation for the Well-Architected Tool[[18]](#footnote-18). Internal references offer detailed insights, while external case studies and success stories offer a broader understanding of how organizations have received help from adopting the Well-Architected Framework.

Conclusion

In conclusion, Chapter 11, *AWS Cloud Computing Master Class* has provided an extensive overview of various AWS management and governance services essential for ensuring the reliability, security, and efficiency of cloud infrastructures. Throughout the chapter, we explored a plethora of AWS tools and services designed to watch, govern, and improve cloud resources effectively[[19]](#footnote-19).

From the foundational services like Amazon CloudWatch and AWS CloudTrail to the advanced solutions such as AWS Systems Manager and AWS Trusted Advisor, organizations have at their disposal a comprehensive toolkit to manage their AWS environments with precision and

confidence. These services empower organizations to gain insights, enforce compliance, automate workflows, and enhance operational resilience in the cloud[[20]](#footnote-20).

As organizations continue to embrace cloud computing, the importance of effective management and governance cannot be overstated. The scalability and flexibility of the cloud brings forth new challenges in terms of resource management, security, and compliance. However, with the right combination of AWS management and governance services, organizations can navigate these challenges seamlessly while unlocking the full potential of the cloud.

To illustrate the practical application of AWS management and governance services, let us consider a scenario:

**Scenario: Implementing governance framework for a financial services firm**

Imagine a large financial services firm that has recently migrated its operations to AWS cloud infrastructure to use its scalability and agility. However, the firm faces stringent regulatory requirements and internal policies on data privacy, security, and compliance.

In response to these challenges, the firm decided to implement a comprehensive governance framework using AWS management services. Here is how they utilize various AWS services[[21]](#footnote-21):

* **AWS organizations:** The firm leverages AWS Organizations to centrally manage and govern multiple AWS accounts across different business units and departments. They implement organizational units to enforce policies and controls specific to each business unit while ensuring centralized billing and cost management.
* **AWS config:** With AWS Config, the firm continuously assesses the compliance of its AWS resources against predefined rules and policies. They configure custom rules to watch access controls, encryption settings, and resource configurations, ensuring adherence to industry standards and regulatory requirements.
* **AWS CloudFormation:** To keep consistency and enforce best practices across their AWS environments, the firm uses AWS CloudFormation to automate the deployment and configuration of infrastructure resources. They define infrastructure as code (IaC) templates to provision standardized environments, reducing manual errors, and ensuring reproducibility.
* **AWS CloudWatch:** For real-time monitoring and operational visibility, the firm relies on Amazon CloudWatch to watch performance metrics, detect anomalies, and trigger automated responses to events. They create custom dashboards and alarms to check **key performance indicators** (**KPIs**) and ensure the availability and performance of critical applications.
* **AWS systems manager:** To streamline operational tasks and ensure security compliance, the firm uses AWS Systems Manager to automate routine administrative tasks, such as patch management, software inventory, and configuration management. They use Systems Manager Automation to orchestrate workflows and remediate security vulnerabilities in real time.
* **AWS service catalog:** The firm uses AWS Service Catalog to govern the provisioning and lifecycle management of approved AWS services and resources. They create portfolios holding pre-approved service templates, allowing teams to provision resources while keeping compliance with organizational policies and standards.
* **AWS trusted advisor:** As part of their ongoing optimization efforts, the firm regularly consults AWS Trusted Advisor to find cost optimization opportunities, enhance security posture, and improve operational performance. They act on the recommendations provided by Trusted Advisor to improve resource use, reduce costs, and enhance overall efficiency.

By implementing this comprehensive governance framework, the financial services firm successfully addresses regulatory requirements, mitigates security risks, and enhances operational efficiency in its AWS cloud environment. Through the strategic use of AWS management and governance services, the firm achieves greater visibility, control, and resilience in the cloud.

In conclusion, mastering AWS management and governance is essential for organizations looking to harness the full potential of cloud computing while keeping security, compliance, and operational excellence. By using the diverse array of AWS management services discussed in this chapter, organizations can build resilient, secure, and cost-effective cloud infrastructures that drive innovation and business growth in the digital age.

1. Amazon Web Services, Inc. (n.d.a). AWS Organizations. Retrieved from https://aws.amazon.com/organizations/ [↑](#footnote-ref-1)
2. Amazon Web Services, Inc. (n.d.b). AWS Organizations Documentation. Retrieved from https://docs.aws.amazon.com/organizations/ [↑](#footnote-ref-2)
3. Amazon Web Services, Inc. (n.d.a). AWS Personal Health Dashboard. Retrieved from https://aws.amazon.com/personal-health-dashboard/ [↑](#footnote-ref-3)
4. Amazon Web Services, Inc. (n.d.b). AWS Personal Health Dashboard Documentation. Retrieved from https://docs.aws.amazon.com/personal-health/latest/userguide/ [↑](#footnote-ref-4)
5. Amazon Web Services, Inc. (n.d.a). AWS Proton. Retrieved from https://aws.amazon.com/proton/ [↑](#footnote-ref-5)
6. Amazon Web Services, Inc. (n.d.b). AWS Proton Documentation. Retrieved from https://docs.aws.amazon.com/proton/ [↑](#footnote-ref-6)
7. Amazon Web Services, Inc. (n.d.a). AWS Resilience Hub. Retrieved from https://aws.amazon.com/architecture/resilience-hub/ [↑](#footnote-ref-7)
8. Amazon Web Services, Inc. (n.d.b). AWS Resilience Hub Documentation. Retrieved from https://docs.aws.amazon.com/wellarchitected/latest/userguide/intro.html [↑](#footnote-ref-8)
9. Amazon Web Services, Inc. (n.d.a). AWS Service Catalog. Retrieved from https://aws.amazon.com/servicecatalog/ [↑](#footnote-ref-9)
10. Amazon Web Services, Inc. (n.d.b). AWS Service Catalog Documentation. Retrieved from https://docs.aws.amazon.com/servicecatalog/ [↑](#footnote-ref-10)
11. Amazon Web Services, Inc. (n.d.a). AWS Service Management Connector. Retrieved from https://aws.amazon.com/servicenow/ [↑](#footnote-ref-11)
12. Amazon Web Services, Inc. (n.d.b). AWS Service Management Connector Documentation. Retrieved from https://docs.aws.amazon.com/servicenow/ [↑](#footnote-ref-12)
13. Amazon Web Services, Inc. (n.d.a). AWS Systems Manager. Retrieved from https://aws.amazon.com/systems-manager/ [↑](#footnote-ref-13)
14. Amazon Web Services, Inc. (n.d.b). AWS Systems Manager Documentation. Retrieved from https://docs.aws.amazon.com/systems-manager/ [↑](#footnote-ref-14)
15. Amazon Web Services, Inc. (n.d.a). AWS Trusted Advisor. Retrieved from https://aws.amazon.com/trusted-advisor/ [↑](#footnote-ref-15)
16. Amazon Web Services, Inc. (n.d.b). AWS Trusted Advisor Documentation. Retrieved from https://docs.aws.amazon.com/awstrustedadvisor/latest/ug/welcome.html [↑](#footnote-ref-16)
17. Amazon Web Services, Inc. (n.d.a). AWS Well-Architected Tool. Retrieved from https://aws.amazon.com/architecture/well-architected/ [↑](#footnote-ref-17)
18. Amazon Web Services, Inc. (n.d.b). AWS Well-Architected Tool Documentation. Retrieved from https://docs.aws.amazon.com/wellarchitected/latest/userguide/intro.html [↑](#footnote-ref-18)
19. Jones, A. (2019). Mastering AWS Management & Governance. Boston: Pearson. [↑](#footnote-ref-19)
20. Smith, J. (2021). Cloud Governance Best Practices. New York: Wiley. [↑](#footnote-ref-20)
21. Amazon Web Services, Inc. (n.d.). AWS Documentation. Retrieved from https://docs.aws.amazon.com/index.html [↑](#footnote-ref-21)