Operational Excellence Pillar

AWS Well-Architected Framework

Operational Excellence Pillar: AWS Well-Architected Framework

Table of Contents

Operational Excellence Pillar - AWS Well-Architected Framework

Publication date: **October 20, 2022** (*Document revisions* (p. 90))

The focus of this paper is the operational excellence pillar of the AWS Well-Architected Framework. It provides guidance to help you apply best practices in the design, delivery, and maintenance of AWS workloads.

Introduction

The AWS Well-Architected Framework helps you understand the benefits and risks of decisions you make while building workloads on AWS. By using the Framework you will learn operational and architectural best practices for designing and operating reliable, secure, efficient, and cost-effective workloads in the cloud. It provides a way to consistently measure your operations and architectures against best practices and identify areas for improvement. We believe that having Well-Architected workloads that are designed with operations in mind greatly increases the likelihood of business success.

The framework is based on six pillars:

- Operational Excellence
- Security
- Reliability
- Performance Efficiency
- Cost Optimization
- Sustainability

This paper focuses on the operational excellence pillar and how to apply it as the foundation of your well-architected solutions. Operational excellence is challenging to achieve in environments where operations is perceived as a function isolated and distinct from the lines of business and development teams that it supports. By adopting the practices in this paper you can build architectures that provide insight to their status, are enabled for effective and efficient operation and event response, and can continue to improve and support your business goals.

This paper is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, and operations team members. After reading this paper, you will understand AWS best practices and the strategies to use when designing cloud architectures for operational excellence. This paper does not provide implementation details or architectural patterns. However, it does include references to appropriate resources for this information.

Operational excellence

The operational excellence pillar includes how your organization supports your business objectives, your ability to run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value.

Topics

Design principles

There are five design principles for operational excellence in the cloud:

- **Perform operations as code:** In the cloud, you can apply the same engineering discipline that you use for application code to your entire environment. You can define your entire workload (applications, infrastructure, etc.) as code and update it with code. You can script your operations procedures and automate their execution by triggering them in response to events. By performing operations as code, you limit human error and enable consistent responses to events.
- **Make frequent, small, reversible changes:** Design workloads to allow components to be updated regularly to increase the flow of beneficial changes into your workload. Make changes in small increments that can be reversed if they fail to aid in the identification and resolution of issues introduced to your environment (without affecting customers when possible).
- **Refine operations procedures frequently:** As you use operations procedures, look for opportunities to improve them. As you evolve your workload, evolve your procedures appropriately. Set up regular game days to review and validate that all procedures are effective and that teams are familiar with them.
- **Anticipate failure:** Perform “pre-mortem” exercises to identify potential sources of failure so that they can be removed or mitigated. Test your failure scenarios and validate your understanding of their impact. Test your response procedures to ensure they are effective and that teams are familiar with their execution. Set up regular game days to test workload and team responses to simulated events.
- **Learn from all operational failures:** Drive improvement through lessons learned from all operational events and failures. *Share what is learned* across teams and through the entire organization.

Definition

There are four best practice areas for operational excellence in the cloud:

- Organization
- Prepare
- Operate
- Evolve

Your organization’s leadership defines business objectives. Your organization must understand requirements and priorities and use these to organize and conduct work to support the achievement of business outcomes. Your workload must emit the information necessary to support it. Implementing

services to enable integration, deployment, and delivery of your workload will enable an increased flow of beneficial changes into production by automating repetitive processes.

There may be risks inherent in the operation of your workload. You must understand those risks and make an informed decision to enter production. Your teams must be able to support your workload. Business and operational metrics derived from desired business outcomes will enable you to understand the health of your workload, your operations activities, and respond to incidents. Your priorities will change as your business needs and business environment changes. Use these as a feedback loop to continually drive improvement for your organization and the operation of your workload.

Organization

You need to understand your organization's priorities, your organizational structure, and how your organization supports your team members, so that they can support your business outcomes.

To enable operational excellence, you must understand the following:

Topics

Organization priorities

Your teams need to have a shared understanding of your entire workload, their role in it, and shared business goals to set the priorities that will enable business success. Well-defined priorities will maximize the benefits of your efforts. Review your priorities regularly so that they can be updated as your organization's needs change.

Best practices

OPS01-BP01 Evaluate external customer needs

Involve key stakeholders, including business, development, and operations teams, to determine where to focus efforts on external customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve your desired business outcomes.

Common anti-patterns:

- You have decided not to have customer support outside of core business hours, but you haven't reviewed historical support request data. You do not know whether this will have an impact on your customers.
- You are developing a new feature but have not engaged your customers to find out if it is desired, if desired in what form, and without experimentation to validate the need and method of delivery.

Benefits of establishing this best practice: Customers whose needs are satisfied are much more likely to remain customers. Evaluating and understanding external customer needs will inform how you prioritize your efforts to deliver business value.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Understand business needs: Business success is enabled by shared goals and understanding across stakeholders, including business, development, and operations teams.
  - Review business goals, needs, and priorities of external customers: Engage key stakeholders, including business, development, and operations teams, to discuss goals, needs, and priorities of external customers. This ensures that you have a thorough understanding of the operational support that is required to achieve business and customer outcomes.
  - Establish shared understanding: Establish shared understanding of the business functions of the workload, the roles of each of the teams in operating the workload, and how these factors support your shared business goals across internal and external customers.

Resources

Related documents:

- AWS Well-Architected Framework Concepts – Feedback loop

OPS01-BP02 Evaluate internal customer needs

Involve key stakeholders, including business, development, and operations teams, when determining where to focus efforts on internal customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve business outcomes.

Use your established priorities to focus your improvement efforts where they will have the greatest impact (for example, developing team skills, improving workload performance, reducing costs, automating runbooks, or enhancing monitoring). Update your priorities as needs change.

Common anti-patterns:

- You have decided to change IP address allocations for your product teams, without consulting them, to make managing your network easier. You do not know the impact this will have on your product teams.
- You are implementing a new development tool but have not engaged your internal customers to find out if it is needed or if it is compatible with their existing practices.
- You are implementing a new monitoring system but have not contacted your internal customers to find out if they have monitoring or reporting needs that should be considered.

Benefits of establishing this best practice: Evaluating and understanding internal customer needs will inform how you prioritize your efforts to deliver business value.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Understand business needs: Business success is enabled by shared goals and understanding across stakeholders including business, development, and operations teams.
  - Review business goals, needs, and priorities of internal customers: Engage key stakeholders, including business, development, and operations teams, to discuss goals, needs, and priorities of internal customers. This ensures that you have a thorough understanding of the operational support that is required to achieve business and customer outcomes.

- Establish shared understanding: Establish shared understanding of the business functions of the workload, the roles of each of the teams in operating the workload, and how these factors support shared business goals across internal and external customers.

Resources

Related documents:

- AWS Well-Architected Framework Concepts – Feedback loop

OPS01-BP03 Evaluate governance requirements

Ensure that you are aware of guidelines or obligations defined by your organization that may mandate or emphasize specific focus. Evaluate internal factors, such as organization policy, standards, and requirements. Validate that you have mechanisms to identify changes to governance. If no governance requirements are identified, ensure that you have applied due diligence to this determination.

Common anti-patterns:

- You are being audited and are asked to provide proof of compliance with internal governance. You have no idea if you are compliant because you have never evaluated what your compliance requirements are.
- You have suffered a compromise resulting in financial loss. You discover that the insurance that would have covered the financial loss was contingent on your implementation of specific security controls that are not in place and required by your governance.
- Your administrative account has been compromised resulting in the defacement of your company web site and damaged to customer trust. Your internal governance requires the use of Multifactor Authentication (MFA) to secure administrative accounts. You did not secure your administrative account with MFA and subject to disciplinary action.

Benefits of establishing this best practice: Evaluating and understanding the governance requirements that your organization applies to your workload will inform how you prioritize your efforts to deliver business value.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Understand governance requirements: Evaluate internal governance factors, such as program or organizational policy, program policies, issue or system specific policies, standards, procedures, baselines, and guidelines. Validate that you have mechanisms to identify changes to governance. If no governance requirements are identified, ensure that you have applied due diligence to this determination.

Resources

Related documents:

- AWS Cloud Compliance

OPS01-BP04 Evaluate compliance requirements

Evaluate external factors, such as regulatory compliance requirements and industry standards, to ensure that you are aware of guidelines or obligations that might mandate or emphasize specific focus. If no compliance requirements are identified, ensure that you apply due diligence to this determination.

Common anti-patterns:

- You are being audited and are asked to provide proof of compliance with industry regulations. You have no idea if you are compliant because you have never evaluated what your compliance requirements are.
- Your administrative account has been compromised resulting in the download of customer data and damaged to customer trust. Your industry best practices require the use of MFA to secure administrative accounts. You did not secure your administrative account with MFA and subject to litigation by your customers.

Benefits of establishing this best practice: Evaluating and understanding the compliance requirements that apply to your workload will inform how you prioritize your efforts to deliver business value.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Understand compliance requirements: Evaluate external factors, such as regulatory compliance requirements and industry standards, to ensure that you are aware of guidelines or obligations that might mandate or emphasize specific focus. If no compliance requirements are identified, ensure that due diligence was applied to the determination.
  - Understand regulatory compliance requirements: Identify regulatory compliance requirements that you are legally obligated to satisfy. Use these requirements to focus your efforts. Examples include obligations from privacy and data protection acts.
    - AWS Compliance
    - AWS Compliance Programs
    - AWS Compliance Latest News
  - Understand industry standards and best practices: Identify industry standards and best practice requirements that apply to your workload, such as the Payment Card Industry Data Security Standard (PCI DSS). Use these requirements to focus your efforts.
    - AWS Compliance Programs
  - Understand internal compliance requirements: Identify compliance requirements and best practices that are established by your organization. Use these requirements to focus your efforts. Examples include information security policies and data classification standards.

Resources

Related documents:

- AWS Cloud Compliance
- AWS Compliance
- AWS Compliance Latest News
- AWS Compliance Programs

OPS01-BP05 Evaluate threat landscape

Evaluate threats to the business (for example, competition, business risk and liabilities, operational risks, and information security threats) and maintain current information in a risk registry. Include the impact of risks when determining where to focus efforts.

The Well-Architected Framework emphasizes learning, measuring, and improving. It provides a consistent approach for you to evaluate architectures, and implement designs that will scale over time. AWS provides the AWS Well-Architected Tool to help you review your approach prior to development, the state of your workloads prior to production, and the state of your workloads in production. You can compare them to the latest AWS architectural best practices, monitor the overall status of your workloads, and gain insight to potential risks.

AWS customers are eligible for a guided Well-Architected Review of their mission-critical workloads to measure their architectures against AWS best practices. Enterprise Support customers are eligible for an Operations Review, designed to help them to identify gaps in their approach to operating in the cloud.

The cross-team engagement of these reviews helps to establish common understanding of your workloads and how team roles contribute to success. The needs identified through the review can help shape your priorities.

AWS Trusted Advisor is a tool that provides access to a core set of checks that recommend optimizations that may help shape your priorities. Business and Enterprise Support customers receive access to additional checks focusing on security, reliability, performance, and cost-optimization that can further help shape their priorities.

Common anti-patterns:

- You are using an old version of a software library in your product. You are unaware of security updates to the library for issues that may have unintended impact on your workload.
- Your competitor just released a version of their product that addresses many of your customers' complaints about your product. You have not prioritized addressing any of these known issues.
- Regulators have been pursuing companies like yours that are not compliant with legal regulatory compliance requirements. You have not prioritized addressing any of your outstanding compliance requirements.

Benefits of establishing this best practice: Identifying and understanding the threats to your organization and workload enables your determination of which threats to address, their priority, and the resources necessary to do so.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Evaluate threat landscape: Evaluate threats to the business (for example, competition, business risk and liabilities, operational risks, and information security threats), so that you can include their impact when determining where to focus efforts.
  - AWS Latest Security Bulletins
  - AWS Trusted Advisor
  - Maintain a threat model: Establish and maintain a threat model identifying potential threats, planned and in place mitigations, and their priority. Review the probability of threats manifesting as incidents, the cost to recover from those incidents and the expected harm caused, and the cost to prevent those incidents. Revise priorities as the contents of the threat model change.

Resources

Related documents:

- AWS Cloud Compliance
- AWS Latest Security Bulletins
- AWS Trusted Advisor

OPS01-BP06 Evaluate tradeoffs

Evaluate the impact of tradeoffs between competing interests or alternative approaches, to help make informed decisions when determining where to focus efforts or choosing a course of action. For example, accelerating speed to market for new features may be emphasized over cost optimization, or you may choose a relational database for non-relational data to simplify the effort to migrate a system, rather than migrating to a database optimized for your data type and updating your application.

AWS can help you educate your teams about AWS and its services to increase their understanding of how their choices can have an impact on your workload. You should use the resources provided by AWS Support (AWS Knowledge Center, AWS Discussion Forums, and AWS Support Center) and AWS Documentation to educate your teams. Reach out to AWS Support through AWS Support Center for help with your AWS questions.

AWS also shares best practices and patterns that we have learned through the operation of AWS in The Amazon Builders' Library. A wide variety of other useful information is available through the AWS Blog and The Official AWS Podcast.

Common anti-patterns:

- You are using a relational database to manage time series and non-relational data. There are database options that are optimized to support the data types you are using but you are unaware of the benefits because you have not evaluated the tradeoffs between solutions.
- Your investors request that you demonstrate compliance with Payment Card Industry Data Security Standards (PCI DSS). You do not consider the tradeoffs between satisfying their request and continuing with your current development efforts. Instead you proceed with your development efforts without demonstrating compliance. Your investors stop their support of your company over concerns about the security of your platform and their investments.

Benefits of establishing this best practice: Understanding the implications and consequences of your choices enables you to prioritize your options.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Evaluate tradeoffs: Evaluate the impact of tradeoffs between competing interests, to help make informed decisions when determining where to focus efforts. For example, accelerating speed to market for new features might be emphasized over cost optimization.
- AWS can help you educate your teams about AWS and its services to increase their understanding of how their choices can have an impact on your workload. You should use the resources provided by AWS Support (AWS Knowledge Center, AWS Discussion Forums, and AWS Support Center) and AWS Documentation to educate your teams. Reach out to AWS Support through AWS Support Center for help with your AWS questions.
- AWS also shares best practices and patterns that we have learned through the operation of AWS in The Amazon Builders' Library. A wide variety of other useful information is available through the AWS Blog and The Official AWS Podcast.

Resources

Related documents:

- AWS Blog
- AWS Cloud Compliance
- AWS Discussion Forums
- AWS Documentation
- AWS Knowledge Center
- AWS Support
- AWS Support Center
- The Amazon Builders' Library
- The Official AWS Podcast

OPS01-BP07 Manage benefits and risks

Manage benefits and risks to make informed decisions when determining where to focus efforts. For example, it may be beneficial to deploy a workload with unresolved issues so that significant new features can be made available to customers. It may be possible to mitigate associated risks, or it may become unacceptable to allow a risk to remain, in which case you will take action to address the risk.

You might find that you want to emphasize a small subset of your priorities at some point in time. Use a balanced approach over the long term to ensure the development of needed capabilities and management of risk. Update your priorities as needs change

Common anti-patterns:

- You have decided to include a library that does everything you need that one of your developers found on the internet. You have not evaluated the risks of adopting this library from an unknown source and do not know if it contains vulnerabilities or malicious code.
- You have decided to develop and deploy a new feature instead of fixing an existing issue. You have not evaluated the risks of leaving the issue in place until the feature is deployed and do not know what the impact will be on your customers.
- You have decided to not deploy a feature frequently requested by customers because of unspecified concerns from your compliance team.

Benefits of establishing this best practice: Identifying the available benefits of your choices, and being aware of the risks to your organization, enables you to make informed decisions.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Manage benefits and risks: Balance the benefits of decisions against the risks involved.
  - Identify benefits: Identify benefits based on business goals, needs, and priorities. Examples include time-to-market, security, reliability, performance, and cost.
  - Identify risks: Identify risks based on business goals, needs, and priorities. Examples include time-to-market, security, reliability, performance, and cost.
  - Assess benefits against risks and make informed decisions: Determine the impact of benefits and risks based on goals, needs, and priorities of your key stakeholders, including business, development,

and operations. Evaluate the value of the benefit against the probability of the risk being realized and the cost of its impact. For example, emphasizing speed-to-market over reliability might provide competitive advantage. However, it may result in reduced uptime if there are reliability issues.

Operating model

Your teams must understand their part in achieving business outcomes. Teams need to understand their roles in the success of other teams, the role of other teams in their success, and have shared goals. Understanding responsibility, ownership, how decisions are made, and who has authority to make decisions will help focus efforts and maximize the benefits from your teams.

The needs of a team will be shaped by their industry, their organization, the makeup of the team, and the characteristics of their workload. It is unreasonable to expect a single operating model to be able to support all teams and their workloads.

The number of operating models present in an organization is likely to increase with the number of development teams. You may need to use a combination of operating models.

Adopting standards and consuming services can simplify operations and limit the support burden in your operating model. The benefit of development efforts on shared standards is magnified by the number of teams who have adopted the standard and who will adopt new features.

It's critical that mechanisms exist to request additions, changes, and exceptions to standards in support of the teams' activities. Without this option, standards become a constraint on innovation. Requests should be approved where viable and determined to be appropriate after an evaluation of benefits and risks.

A well-defined set of responsibilities will reduce the frequency of conflicting and redundant efforts. Business outcomes are easier to achieve when there is strong alignment and relationships between business, development, and operations teams.

Operating model 2 by 2 representations

These operating model 2 by 2 representations are illustrations to help you understand the relationships between teams in your environment. These diagrams focus on who does what and the relationships between teams, but we will also discuss governance and decision making in context of these examples.

Our teams may have responsibilities in multiple parts of multiple models depending on the workloads they support. You may wish to break out more specialized discipline areas than the high-level ones described. There is the potential for endless variation on these models as you separate or aggregate activities, or overlay teams and provide more specific detail.

You may identify that you have overlapping or unrecognized capabilities across teams that can provide additional advantage, or lead to efficiencies. You may also identify unsatisfied needs in your organization that you can plan to address.

When evaluating organizational change, examine the trade-offs between models, where your individual teams exist within the models (now and after the change), how your teams' relationship and responsibilities will change, and if the benefits merit the impact on your organization.

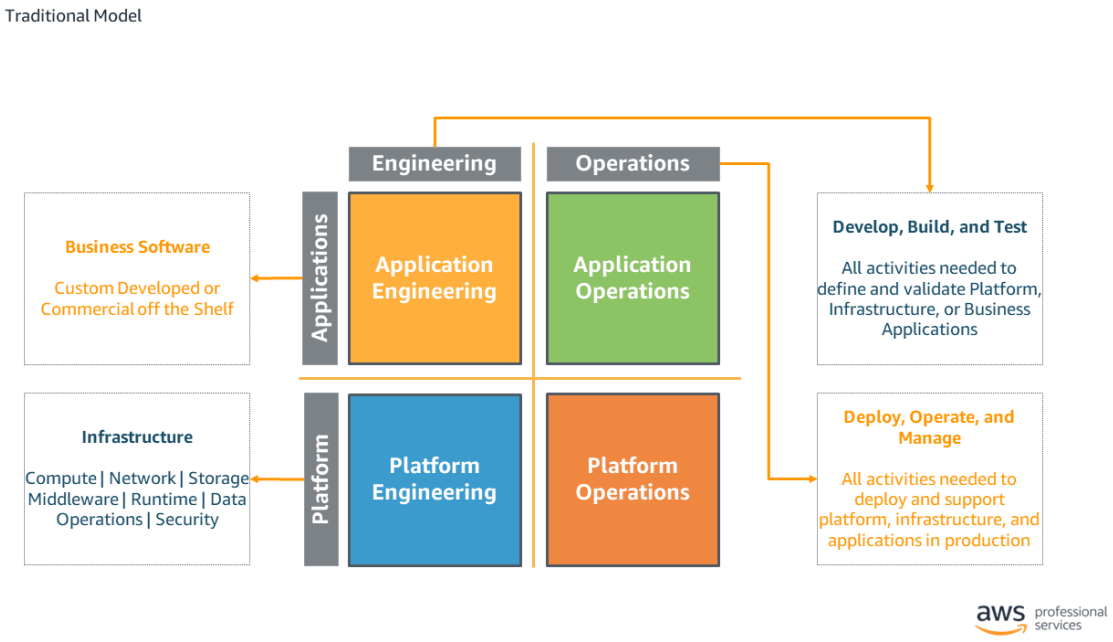
You can be successful using each of the following four operating models. Some models are more appropriate for specific use cases or at specific points in your development. Some of these models may provide advantages over the ones in use in your environment.

Topics

Fully separated operating model

In the following diagram, on the vertical axis we have applications and infrastructure. Applications refer to the workload serving a business outcome and can be custom developed or purchased software. Infrastructure refers to the physical and virtual infrastructure and other software that supports that workload.

On the horizontal axis, we have Engineering and Operations. Engineering refers to the development, building, and testing of applications and infrastructure. Operations is the deployment, update, and ongoing support of applications and infrastructure.



In many organizations, this "fully separated" model is present. The activities in each quadrant are performed by a separate team. Work is passed between teams through mechanisms such as work requests, work queues, tickets, or by using an IT service management (ITSM) system.

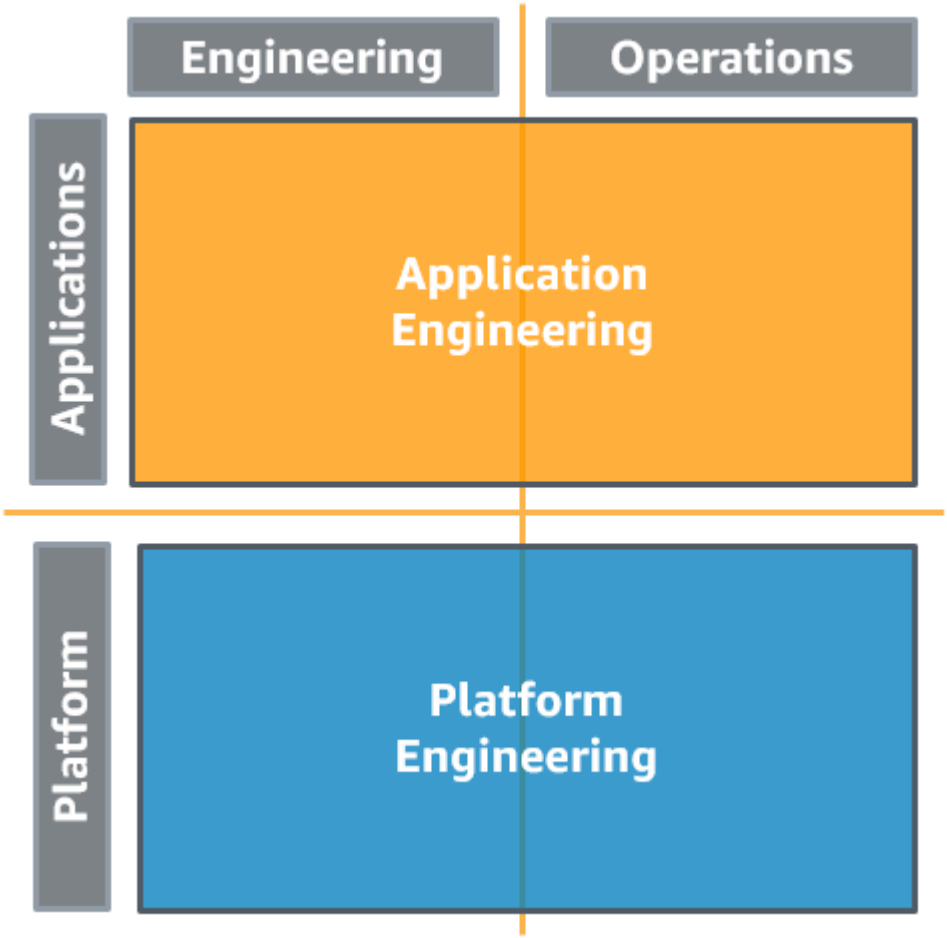
The transition of tasks to or between teams increases complexity, and creates bottlenecks and delays. Requests may be delayed until they are a priority. Defects identified late may require significant rework and may have to pass through the same teams and their functions once again. If there are incidents that require action by engineering teams, their responses are delayed by the hand off activity.

There is a higher risk of misalignment when business, development, and operations teams are organized around the activities or functions that are being performed. This can lead to teams focusing on their specific responsibilities instead of focusing on achieving business outcomes. Teams may be narrowly specialized, physically isolated, or logically isolated, hindering communication and collaboration.

Separated Application Engineering and Operations (AEO) and Infrastructure Engineering and Operations (IEO) with centralized governance

This “Separated AEO and IEO” model follows a “you build it you run it” methodology.

Your application engineers and developers perform both the engineering and the operation of their workloads. Similarly, your infrastructure engineers perform both the engineering and operation of the platforms they use to support application teams.



For this example, we are going to treat governance as centralized. Standards are distributed, provided, or shared to the application teams.

You should use tools or services that enable you to centrally govern your environments across accounts, such as AWS Organizations. Services like AWS Control Tower expand this management capability enabling you to define blueprints (supporting your operating models) for the setup of accounts, apply ongoing governance using AWS Organizations, and automate provisioning of new accounts.

“You build it you run it” does not mean that the application team is responsible for the full stack, tool chain, and platform.

The platform engineering team provides a standardized set of services (for example, development tools, monitoring tools, backup and recovery tools, and network) to the application team. The platform team

may also provide the application team access to approved cloud provider services, specific configurations of the same, or both.

Mechanisms that provide a self-service capability for deploying approved services and configurations, such as AWS Service Catalog, can help limit delays associated with fulfillment requests while enforcing governance.

The platform team enables full stack visibility so that application teams can differentiate between issues with their application components and the services and infrastructure components their applications consume. The platform team may also provide assistance configuring these services and guidance on how to improve the applications teams' operations.

As discussed previously, it's critical that mechanisms exist for the application team to request additions, changes, and exceptions to standards in support of teams' activities and innovation of their application.

The Separated AEO IEO model provides strong feedback loops to application teams. Day to day operations of a workload increases contact with customers either through direct interaction or indirectly through support and feature requests. This heightened visibility allows application teams to address issues more quickly. The deeper engagement and closer relationship provides insight to customer needs and enables more rapid innovation.

All of this is also true for the platform team supporting the application teams.

Adopted standards may be pre-approved for use, reducing the amount of review necessary to enter production. Consuming supported and tested standards provided by the platform team may reduce the frequency of issues with those services. Adoption of standards enables application teams to focus on differentiating their workloads.

Separated AEO and IEO with centralized governance and a service provider

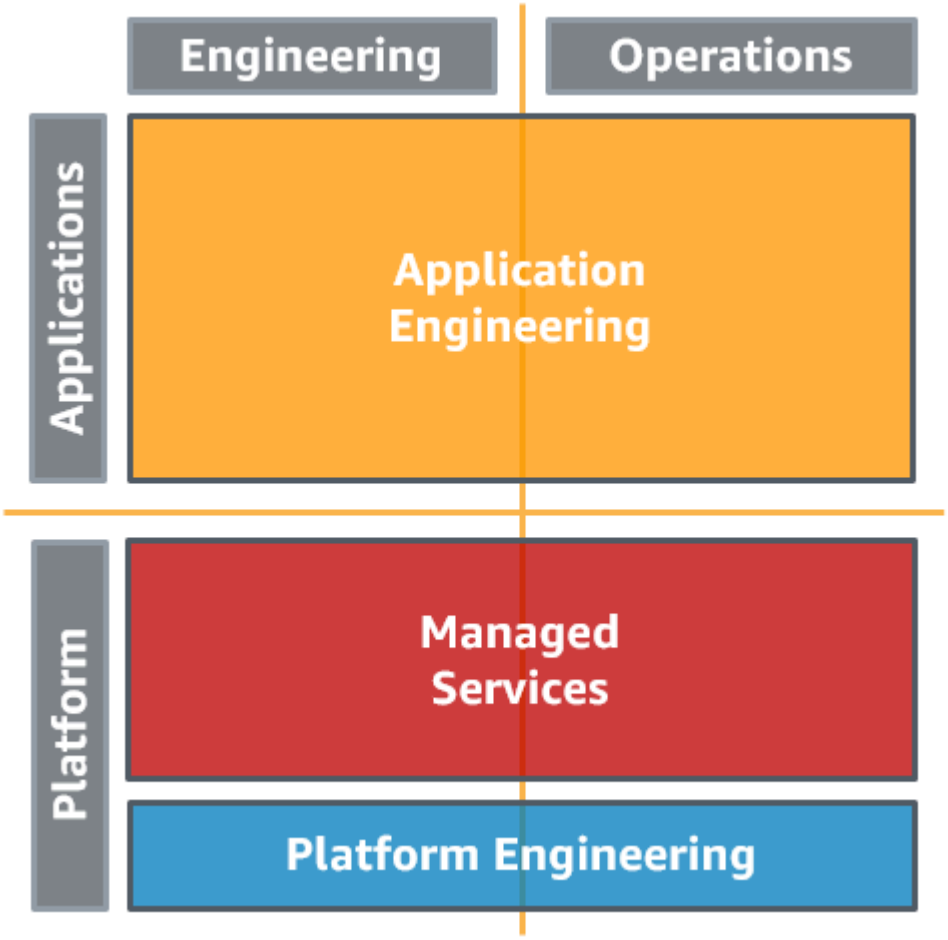
This "Separated AEO and IEO" model follows a "you build it you run it" methodology.

Your application engineers and developers perform both the engineering and the operation of their workloads.

Your organization may not have the existing skills, or team members, to support a dedicated platform engineering and operations team, or you may not want to make the investments of time and effort to do so.

Alternatively, you may wish to have a platform team that is focused on creating capabilities that will differentiate your business, but you want to offload the undifferentiated day to day operations to an outsourcer.

Managed Services providers such as AWS Managed Services, AWS Managed Services Partners, or Managed Services Providers in the AWS Partner Network, provide expertise implementing cloud environments, and support your security and compliance requirements and business goals.



For this variation, we are going to treat governance as centralized and managed by the platform team, with account creation and policies managed with AWS Organizations and AWS Control Tower.

This model does require you to modify your mechanisms to work with those of your service provider. It does not address the bottlenecks and delays created by transition of tasks between teams, including your service provider, or the potential rework related to the late identification of defects.

You gain the advantage of your providers' standards, best practices, processes, and expertise. You also gain the benefits of their ongoing development of their service offerings.

Adding Managed Services to your operating model can save you time and resources, and lets you keep your internal teams lean and focused on strategic outcomes that will differentiate your business, rather than developing new skills and capabilities.

Separated AEO and IEO with centralized governance and an internal service provider consulting partner

This "Separated AEO and IEO" model seeks to establish a "you build it you run it" methodology.

You want your application teams to perform the engineering and operations activities for their workloads, and to adopt a more DevOps like culture.

Your application teams may be in-progress migrating, adopting the cloud, or modernizing your workloads, and not have the existing skills to adequately support cloud and cloud operations. This lack of application team capabilities or familiarity may be barriers to your efforts.

To address this concern you establish a Cloud Center of Enablement team (CCoE) that provides a forum to ask questions, discuss needs, and identify solutions. Depending on the needs of your organization, the CCoE can be a dedicated team of experts or a virtual team with participants selected from across your organization. The CCoE enables cloud transformation for teams, establishes centralized cloud governance, and defines account and organization management standards. They also identify successful reference architectures and patterns for enterprise use.

We refer to CCoE as Cloud Center of Enablement, instead of the more common Cloud Center of Excellence, to place the emphasis on enabling the success of the supported teams and the achievement of business outcomes.

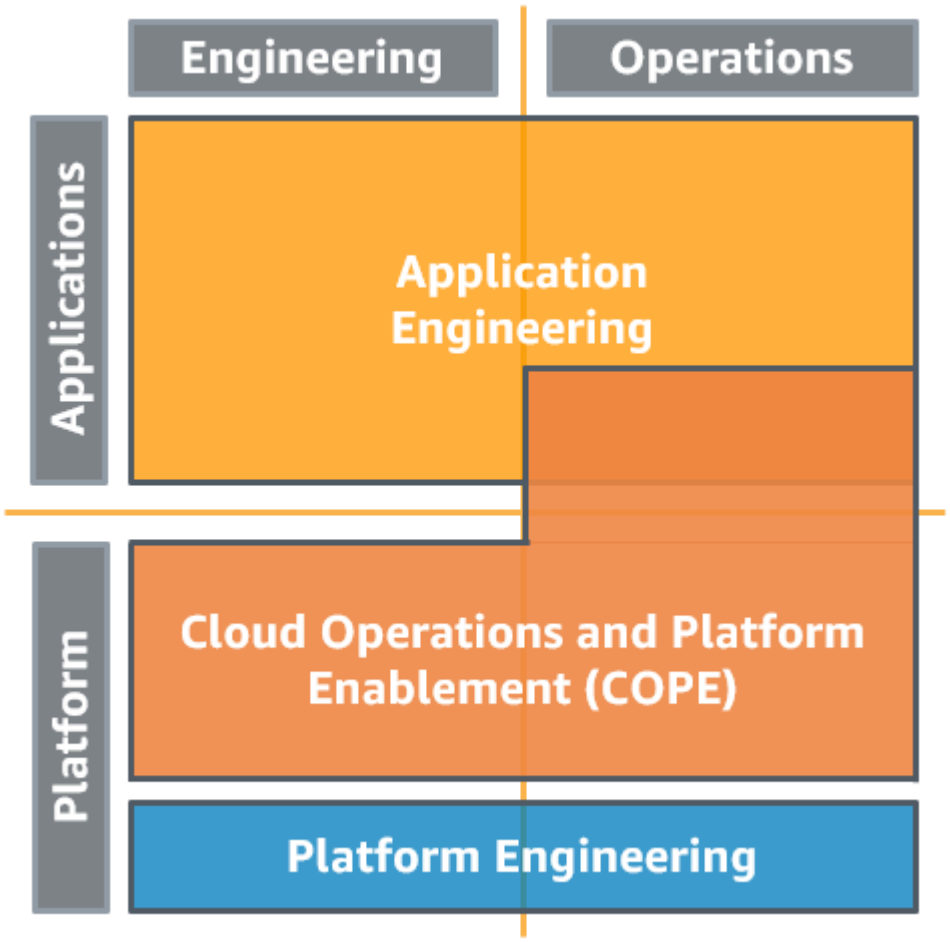
Your platform engineering team builds the core shared platform capabilities based on those standards for application teams to adopt. They codify the enterprise reference architectures and patterns that are provided to the application teams through a self-service mechanism. Using a service such as AWS Service Catalog the application teams can deploy approved reference architectures, patterns, services, and configurations, compliant by default with the centralized governance and security standards.

The platform engineering team also provides a standardized set of services (for example, development tools, monitoring tools, backup and recovery tools, and network) to the application teams.

Your organization has an “Internal MSP and Consulting Partner” that manages and supports the standardized services and provides assistance to application teams establishing their cloud presence based on the reference architectures and patterns. This “Cloud Operations and Platform Enablement (COPE)” team works with the applications teams to help them establish baseline operations with the application teams progressively taking more responsibility for their systems and resources over time. The COPE team drives continual improvement together with the CCoE and Platform Engineering teams, and acts as proponents for the application teams.

The application teams get assistance setting up environments, CICD pipelines, change management, observability and monitoring, and establishing incident and event management processes with the COPE team integrated with those of the enterprise as required. The COPE team participates with the application teams in the performance of these operations activities, phasing out the COPE team engagement over time as the application teams take ownership.

The application team gains the benefit of the skills of the COPE team and the lessons learned by the organization. They are protected by the guardrails established through centralized governance. The application team builds upon recognized successes and gain the benefit of continuing development of the organizational standards they have adopted. They gain greater insight to the operation of their workload through the process of establishing observability and monitoring, and are better able to understand the impact of changes they make to their workloads.



The COPE team retains the access necessary to support operations activities, provide an enterprise-operations view spanning application teams, and to provide critical incident management support. The COPE team retains responsibility for activities considered undifferentiated heavy lifting, which they satisfy through standard solutions supportable at scale. They also continue to manage well-understood programmatic and automated operations activities for the application teams so that they can focus on differentiating their applications.

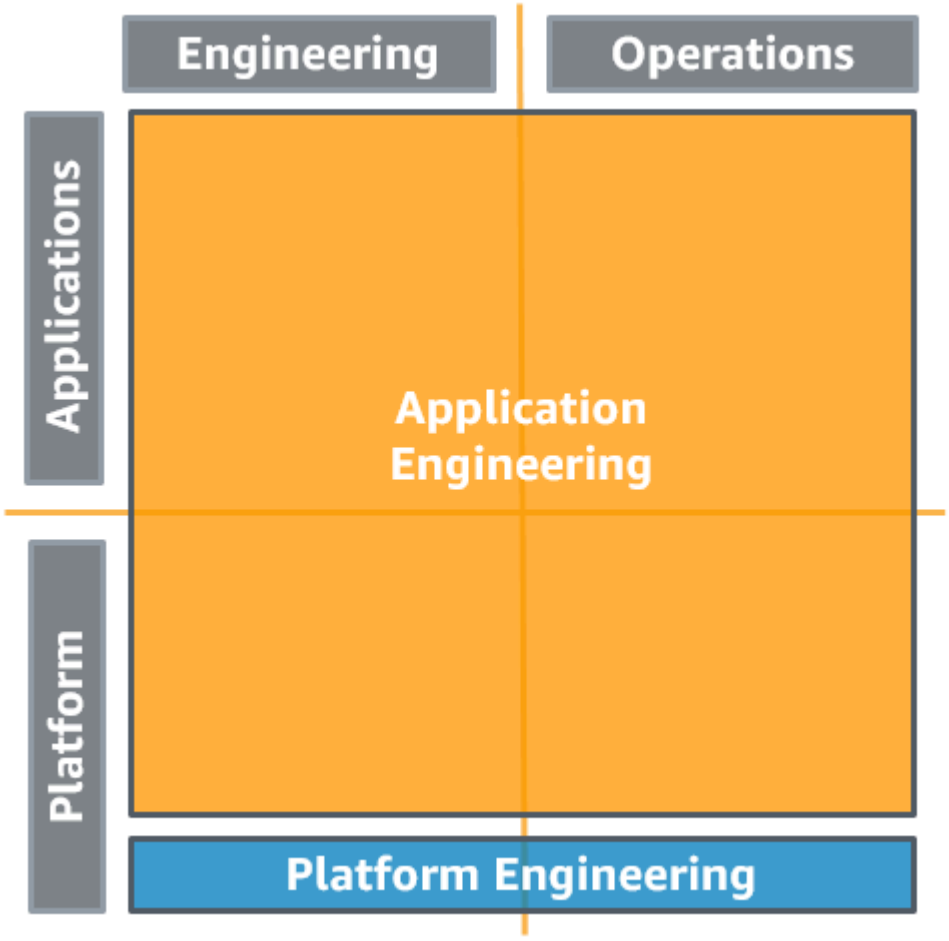
You gain the advantage of your organization's standards, best practices, processes, and expertise derived from the successes of your teams. You establish a mechanism to replicate these successful patterns for new teams adopting or modernizing on the cloud. This model places emphasis on the COPE team's ability to help application team get established, and transition knowledge and artifacts. It reduces the operational burdens of the application teams with the risk that application teams will fail to become largely independent. It establishes relationships between CCoE, COPE, and application teams creating a feedback loop to support further evolution and innovation.

Establishing your CCoE and COPE teams, while defining organization wide standards, can facilitate cloud adoption and support modernization efforts. By providing the additional supports of a COPE team acting as consultants and partners to your application teams you can remove barriers that slow application team adoption of beneficial cloud capabilities.

Separated AEO and IEO with decentralized governance

This “Separated AEO and IEO” model follows a “you build it you run it” methodology.

Your application engineers and developers perform both the engineering and the operation of their workloads. Similarly your infrastructure engineers perform both the engineering and operation of the platforms they use to support application teams.



For this example, we are going to treat governance as decentralized.

Standards are still distributed, provided, or shared to application teams by the platform team, but application teams are free to engineer and operate new platform capabilities in support of their workload.

In this model, there are fewer constraints on the application team, but that comes with a significant increase in responsibilities. Additional skills, and potentially team members, must be present to support the additional platform capabilities. The risk of significant rework is increased if skill sets are not adequate and defects are not recognized early.

You should enforce policies that are not specifically delegated to application teams. Use tools or services that enable you to centrally govern your environments across accounts, such as AWS Organizations. Services like AWS Control Tower expand this management capability enabling you to define blueprints (supporting your operating models) for the setup of accounts, apply ongoing governance using AWS Organizations, and automate provisioning of new accounts.

It's beneficial to have mechanisms for the application team to request additions and changes to standards. They may be able to contribute new standards that can provide benefit to other application teams. The platform teams may decide that providing direct support for these additional capabilities is an effective support for business outcomes.

This model limits constraints on innovation with significant skill and team member requirements. It addresses many of the bottlenecks and delays created by transition of tasks between teams while still promoting the development of effective relationships between teams and customers.

Relationships and ownership

Your operating model defines the relationships between teams and supports identifiable ownership and responsibility.

Best practices

OPS02-BP01 Resources have identified owners

Understand who has ownership of each application, workload, platform, and infrastructure component, what business value is provided by that component, and why that ownership exists. Understanding the business value of these individual components and how they support business outcomes informs the processes and procedures applied against them.

Benefits of establishing this best practice: Understanding ownership identifies whom can approve improvements, implement those improvements, or both.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Resources have identified owners: Define what ownership means for the resource use cases in your environment. Specify and record owners for resources including at a minimum name, contact information, organization, and team. Store resource ownership information with resources using metadata such as tags or resource groups. Use AWS Organizations to structure accounts and implement policies to ensure ownership and contact information are captured.
  - Define forms of ownership and how they are assigned: Ownership may have multiple definitions in your organization with different uses cases. You may wish to define a workload owner as the individual who owns the risk and liability for the operation of a workload, and whom ultimately has authority to make decisions about the workload. You may wish to define ownership in terms of financial or administrative responsibility where ownership rolls up to a parent organization. A developer may be the owner of their development environment and be responsible for incidents that its operation causes. Their product lead may own responsibility for the financial costs associated to the operation of their development environments.
  - Define who owns an organization, account, collection of resources, or individual components: Define and record ownership in an appropriately accessible location organized to support discovery. Update definitions and ownership details as they change.

- Capture ownership in the metadata for the resources: Capture resource ownership using metadata such as tags or resource groups, specifying ownership and contact information. Use AWS Organizations to structure accounts and ensure ownership and contact information are captured.

OPS02-BP02 Processes and procedures have identified owners

Understand who has ownership of the definition of individual processes and procedures, why those specific process and procedures are used, and why that ownership exists. Understanding the reasons that specific processes and procedures are used enables identification of improvement opportunities.

Benefits of establishing this best practice: Understanding ownership identifies who can approve improvements, implement those improvements, or both.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Process and procedures have identified owners responsible for their definition: Capture the processes and procedures used in your environment and the individual or team responsible for their definition.
  - Identify process and procedures: Identify the operations activities conducted in support of your workloads. Document these activities in a discoverable location.
  - Define who owns the definition of a process or procedure: Uniquely identify the individual or team responsible for the specification of an activity. They are responsible to ensure it can be successfully performed by an adequately skilled team member with the correct permissions, access, and tools. If there are issues with performing that activity, the team members performing it are responsible to provide the detailed feedback necessary for the activitiy to be improved.
  - Capture ownership in the metadata of the activity artifact: Procedures automated in services like AWS Systems Manager, through documents, and AWS Lambda, as functions, support capturing metadata information as tags. Capture resource ownership using tags or resource groups, specifying ownership and contact information. Use AWS Organizations to create tagging polices and ensure ownership and contact information are captured.

OPS02-BP03 Operations activities have identified owners responsible for their performance

Understand who has responsibility to perform specific activities on defined workloads and why that responsibility exists. Understanding who has responsibility to perform activities informs who will conduct the activity, validate the result, and provide feedback to the owner of the activity.

Benefits of establishing this best practice: Understanding who is responsible to perform an activity informs whom to notify when action is needed and who will perform the action, validate the result, and provide feedback to the owner of the activity.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Operations activities have identified owners responsible for their performance: Capture the responsibility for performing processes and procedures used in your environment
  - Identify process and procedures: Identify the operations activities conducted in support of your workloads. Document these activities in a discoverable location.
  - Define who is responsible to perform each activity: Identify the team responsible for an activity. Ensure they have the details of the activity, and the necessary skills and correct permissions,

access, and tools to perform the activity. They must understand the condition under which it is to be perform (for example, on an event or schedule). Make this information discoverable so that members of your organization can identify who they need to contact, team or individual, for specific needs.

OPS02-BP04 Team members know what they are responsible for

Understanding the responsibilities of your role and how you contribute to business outcomes informs the prioritization of your tasks and why your role is important. This enables team members to recognize needs and respond appropriately.

Benefits of establishing this best practice: Understanding your responsibilities informs the decisions you make, the actions you take, and your hand off activities to their proper owners.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Enure team members understand their roles and responsibilities: Identify team members roles and responsibilities and ensure they understand the expectations of their role. Make this information discoverable so that members of your organization can identify who they need to contact, team or individual, for specific needs.

OPS02-BP05 Mechanisms exist to identify responsibility and ownership

Where no individual or team is identified, there are defined escalation paths to someone with the authority to assign ownership or plan for that need to be addressed.

Benefits of establishing this best practice: Understanding who has responsbility or ownership allows you to reach out to the proper team or team member to make a request or transition a task. Having an identified person who has the authority to assign responsbility or ownership or plan to address needs reduces the risk of inaction and needs not being addressed.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Mechanisms exist to identify responsibility and ownership: Provide accessible mechanisms for members of your organization to discover and identify ownership and responsibility. These mechanisms will enable them to identify who to contact, team or individual, for specific needs.

OPS02-BP06 Mechanisms exist to request additions, changes, and exceptions

You are able to make requests to owners of processes, procedures, and resources. Make informed decisions to approve requests where viable and determined to be appropriate after an evaluation of benefits and risks.

Benefits of establishing this best practice: It's critical that mechanisms exist to request additions, changes, and exceptions in support of teams' activities. Without this option, current state become a constraint on innovation.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Mechanisms exist to request additions, changes, and exceptions: When standards are rigid innovation is constrained. Provide mechanisms for members of your organization to make requests to owners of processes, procedures, and resources in support of their business needs.

OPS02-BP07 Responsibilities between teams are predefined or negotiated

Have defined or negotiated agreements between teams describing how they work with and support each other (for example, response times, service level objectives, or service level agreements). Understanding the impact of the teams' work on business outcomes, and the outcomes of other teams and organizations, informs the prioritization of their tasks and enables them to respond appropriately.

When responsibility and ownership are undefined or unknown, you are at risk of both not addressing necessary activities in a timely fashion and of redundant and potentially conflicting efforts emerging to address those needs.

Benefits of establishing this best practice: Establishing the responsibilities between teams, the objectives, and the methods for communicating needs, eases the flow of requests and helps ensures the necessary information is provided. This reduces the delay introduced by transition tasks between teams and help support the achievement of business outcomes.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Responsibilities between teams are predefined or negotiated: Specifying the methods by which teams interact, and the information necessary for them to support each other, can help minimize the delay introduced as requests are iteratively reviewed and clarified. Having specific agreements that define expectations (for example, response time, or fulfillment time) enables teams to make effective plans and resource appropriately.

Organizational culture

Provide support for your team members so that they can be more effective in taking *action and supporting your business outcome.*

Best practices

OPS03-BP01 Executive Sponsorship

Senior leadership clearly sets expectations for the organization and evaluates success. Senior leadership is the sponsor, advocate, and driver for the adoption of best practices and evolution of the organization

Benefits of establishing this best practice: Engaged leadership, clearly communicated expectations, and shared goals ensures that team members know what is expected of them. Evaluating success enables identification of barriers to success so that they can be addressed through intervention by the sponsor advocate or their delegates.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Executive Sponsorship: Senior leadership clearly sets expectations for the organization and evaluates success. Senior leadership is the sponsor, advocate, and driver for the adoption of best practices and evolution of the organization
  - Set expectations: Define and publish goals for your organizations including how they will be measured.
  - Track achievement of goals: Measure the incremental achievement of goals regularly and share the results so that appropriate action can be taken if outcomes are at risk.
  - Provide the resources necessary to achieve your goals: Regularly review if resources are still appropriate, of if additional resources are needed based on: new information, changes to goals, responsibilities, or your business environment.
  - Advocate for your teams: Remain engaged with your teams so that you understand how they are doing and if there are external factors affecting them. When your teams are impacted by external factors, reevaluate goals and adjust targets as appropriate. Identify obstacles that are impeding your teams progress. Act on behalf of your teams to help address obstacles and remove unnecessary burdens.
  - Be a driver for adoption of best practices: Acknowledge best practices that have provide quantifiable benefits and recognize the creators and adopters. Encourage further adoption to magnify the benefits achieved.
  - Be a driver for evolution of for your teams: Create a culture of continual improvement. Encourage both personal and organizational growth and development. Provide long term targets to strive for that will require incremental achievement over time. Adjust this vision to compliment your needs, business goals, and business environment as they change.

OPS03-BP02 Team members are empowered to take action when outcomes are at risk

The workload owner has defined guidance and scope empowering team members to respond when outcomes are at risk. Escalation mechanisms are used to get direction when events are outside of the defined scope.

Benefits of establishing this best practice: By testing and validating changes early, you are able to address issues with minimized costs and limit the impact on your customers. By testing prior to deployment you minimize the introduction of errors.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Team members are empowered to take action when outcomes are at risk: Provide your team members the permissions, tools, and opportunity to practice the skills necessary to respond effectively.

- Give your team members opportunity to practice the skills necessary to respond: Provide alternative safe environments where processes and procedures can be tested and trained upon safely. Perform game days to allow team members to gain experience responding to real world incidents in simulated and safe environments.
- Define and acknowledge team members' authority to take action: Specifically define team members authority to take action by assigning permissions and access to the workloads and components they support. Acknowledge that they are empowered to take action when outcomes are at risk.

OPS03-BP03 Escalation is encouraged

Team members have mechanisms and are encouraged to escalate concerns to decision makers and stakeholders if they believe outcomes are at risk. Escalation should be performed early and often so that risks can be identified, and prevented from causing incidents.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Encourage early and frequent escalation: Organizationally acknowledge that escalation early and often is the best practice. Organizationally acknowledge and accept that escalations may prove to be unfounded, and that it is better to have the opportunity to prevent an incident then to miss that opportunity by not escalating.
  - Have a mechanism for escalation: Have documented procedures defining when and how escalation should occur. Document the series of people with increasing authority to take action or approve action and their contact information. Escalation should continue until the team member is satisfied that they have handed off the risk to a person able to address it, or they have contacted the person who owns the risk and liability for the operation of the workload. It is that person who ultimately owns all decisions with respect to their workload. Escalations should include the nature of the risk, the criticality of the workload, who is impacted, what the impact is, and the urgency, that is, when is the impact expected.
  - Protect employees who escalate: Have policy that protects team members from retribution if they escalate around a non-responsive decision maker or stakeholder. Have mechanisms in place to identify if this is occurring and respond appropriately.

OPS03-BP04 Communications are timely, clear, and actionable

Mechanisms exist and are used to provide timely notice to team members of known risks and planned events. Necessary context, details, and time (when possible) are provided to support determining if action is necessary, what action is required, and to take action in a timely manner. For example, providing notice of software vulnerabilities so that patching can be expedited, or providing notice of planned sales promotions so that a change freeze can be implemented to avoid the risk of service disruption.

Planned events can be recorded in a change calendar or maintenance schedule so that team members can identify what activities are pending.

On AWS, AWS Systems Manager Change Calendar can be used to record these details. It supports programmatic checks of calendar status to determine if the calendar is open or closed to activity at a particular point of time. Operations activities can be planned around specific *approved* windows of time that are reserved for potentially disruptive activities. AWS Systems Manager Maintenance Windows allows you to schedule activities against instances and other supported resources to automate the activities and make those activities discoverable.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Communications are timely, clear, and actionable: Mechanisms are in place to provide notification of risks or planned events in a clear and actionable way with enough notice to allow appropriate responses.
  - Document planned activities on a change calendar and provide notifications: Provide an accessible source of information where planned events can be discovered. Provide notifications of planned events from the same system.
  - Track events and activity that may have an impact on your workload: Monitoring vulnerability notifications and patch information to understand vulnerabilities in the wild and potential risks associated to your workload components. Provide notification to team members so that they can take action.

Resources

Related documents:

- AWS Systems Manager Change Calendar
- AWS Systems Manager Maintenance Windows

OPS03-BP05 Experimentation is encouraged

Experimentation accelerates learning and keeps team members interested and engaged. An undesired result is a successful experiment that has identified a path that will not lead to success. Team members are not punished for successful experiments with undesired results. Experimentation is required for innovation to happen and turn ideas into outcomes.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Experimentation is encouraged: Encourage experimentation to support learning and innovation.
  - Experiment with a variety of technologies: Encourage experimentation with technologies that may have applicability now or in the future to the achievement of your business outcomes. This knowledge may inform future innovation.
  - Experiment with a goal in mind: Encourage experimentation with specific goals for team members to reach for, or with technologies that may have applicability in the near future. This knowledge may inform your innovation.
  - Provide structured time to experiment: Dedicate specific times when team members can be free of their normal responsibilities, so that they can focus on their experiments.
  - Provide the resources to support experimentation: Fund the resources required to conduct experiments (for example, software, or cloud resources).
  - Acknowledge success: Recognize the value yielded by experimentation. Understand that experiments with undesired outcomes are successful and have identified a path that will not lead to success. Team members are not punished for undesired outcomes from experiments.

OPS03-BP06 Team members are enabled and encouraged to maintain and grow their skill sets

Teams must grow their skill sets to adopt new technologies, and to support changes in demand and responsibilities in support of your workloads. Growth of skills in new technologies is frequently a source of team member satisfaction and supports innovation. Support your team members' pursuit and maintenance of industry certifications that validate and acknowledge their growing skills. Cross train to promote knowledge transfer and reduce the risk of significant impact when you lose skilled and experienced team members with institutional knowledge. Provide dedicated structured time for learning.

AWS provides resources, including the AWS Getting Started Resource Center, AWS Blogs, AWS Online Tech Talks, AWS Events and Webinars, and the AWS Well-Architected Labs, that provide guidance, examples, and detailed walkthroughs to educate your teams.

AWS also shares best practices and patterns that we have learned through the operation of AWS in The Amazon Builders' Library and a wide variety of other useful educational material through the AWS Blog and The Official AWS Podcast.

You should take advantage of the education resources provided by AWS such as the Well-Architected labs, AWS Support (AWS Knowledge Center, AWS Discussion Forms, and AWS Support Center) and AWS Documentation to educate your teams. Reach out to AWS Support through AWS Support Center for help with your AWS questions.

AWS Training and Certification provides some free training through self-paced digital courses on AWS fundamentals. You can also register for instructor-led training to further support the development of your teams' AWS skills.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Team members are enabled and encouraged to maintain and grow their skill sets: To adopt new technologies, support innovation, and to support changes in demand and responsibilities in support of your workloads continuing education is necessary.
  - Provide resources for education: Provided dedicated structured time, access to training materials, lab resources, and support participation in conferences and professional organizations that provide opportunities for learning from both educators and peers. Provide junior team members' access to senior team members as mentors or allow them to shadow their work and be exposed to their methods and skills. Encourage learning about content not directly related to work in order to have a broader perspective.
  - Team education and cross-team engagement: Plan for the continuing education needs of your team members. Provide opportunities for team members to join other teams (temporarily or permanently) to share skills and best practices benefiting your entire organization
  - Support pursuit and maintenance of industry certifications: Support your team members acquiring and maintaining industry certifications that validate what they have learned, and acknowledge their accomplishments.

Resources

Related documents:

- AWS Getting Started Resource Center
- AWS Blogs
- AWS Cloud Compliance

- AWS Discussion Forms
- AWS Documentation
- AWS Online Tech Talks
- AWS Events and Webinars
- AWS Knowledge Center
- AWS Support
- AWS Training and Certification
- AWS Well-Architected Labs,
- The Amazon Builders' Library
- The Official AWS Podcast.

OPS03-BP07 Resource teams appropriately

Maintain team member capacity, and provide tools and resources to support your workload needs. Overtasking team members increases the risk of incidents resulting from human error. Investments in tools and resources (for example, providing automation for frequently performed activities) can scale the effectiveness of your team, enabling them to support additional activities.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Resource teams appropriately: Ensure you have an understanding of the success of your teams and the factors that contribute to their success or lack of success. Act to support teams with appropriate resources.
  - Understand team performance: Measure the achievement of operational outcomes and the development of assets by your teams. Track changes in output and error rate over time. Engage with teams to understand the work related challenges that impact them (for example, increasing responsibilities, changes in technology, loss of personnel, or increase in customers supported).
  - Understand impacts on team performance: Remain engaged with your teams so that you understand how they are doing and if there are external factors affecting them. When your teams are impacted by external factors, reevaluate goals and adjust targets as appropriate. Identify obstacles that are impeding your teams progress. Act on behalf of your teams to help address obstacles and remove unnecessary burdens.
  - Provide the resources necessary for teams to be successful: Regularly review if resources are still appropriate, of if additional resources are needed, and make appropriate adjustments to support teams.

OPS03-BP08 Diverse opinions are encouraged and sought within and across teams

Leverage cross-organizational diversity to seek multiple unique perspectives. Use this perspective to increase innovation, challenge your assumptions, and reduce the risk of confirmation bias. Grow inclusion, diversity, and accessibility within your teams to gain beneficial perspectives.

Organizational culture has a direct impact on team member job satisfaction and retention. Enable the engagement and capabilities of your team members to enable the success of your business.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Seek diverse opinions and perspectives: Encourage contributions from everyone. Give voice to under-represented groups. Rotate roles and responsibilities in meetings.
  - Expand roles and responsibilities: Provide opportunity for team members to take on roles that they might not otherwise. They will gain experience and perspective from the role, and from interactions with new team members with whom they might not otherwise interact. They will bring their experience and perspective to the new role and team members they interact with. As perspective increases, additional business opportunities may emerge, or new opportunities for improvement may be identified. Have members within a team take turns at common tasks that others typically perform to understand the demands and impact of performing them.
  - Provide a safe and welcoming environment: Have policy and controls that protect team members' mental and physical safety within your organization. Team members should be able to interact without fear of reprisal. When team members feel safe and welcome they are more likely to be engaged and productive. The more diverse your organization the better your understanding can be of the people you support including your customers. When your team members are comfortable, feel free to speak, and are confident they will be heard, they are more likely to share valuable insights (for example, marketing opportunities, accessibility needs, unserved market segments, unacknowledged risks in your environment).
  - Enable team members to participate fully: Provide the resources necessary for your employees to participate fully in all work related activities. Team members that face daily challenges have developed skills for working around them. These uniquely developed skills can provide significant benefit to your organization. Supporting team members with necessary accommodations will increase the benefits you can receive from their contributions.

Prepare

To prepare for operational excellence, you have to understand your workloads and their expected behaviors. You will then be able to design them to provide insight to their status and build the procedures to support them.

To prepare for operational excellence, you need to perform the following:

Topics

Design telemetry

Design your workload so that it provides the information necessary for you to understand its internal state (for example, metrics, logs, events, and traces) across all components in support of observability and investigating issues. Iterate to develop the telemetry necessary to monitor the health of your workload, identify when outcomes are at risk, and enable effective responses.

In AWS, you can emit and collect logs, metrics, and events from your applications and workloads components to enable you to understand their internal state and health. You can integrate distributed tracing to track requests as they travel through your workload. Use this data to understand how your application and underlying components interact and to analyze issues and performance.

When instrumenting your workload, capture a broad set of information to enable situational awareness (for example, changes in state, user activity, privilege access, utilization counters), knowing that you can use filters to select the most useful information over time.

Best practices

OPS04-BP01 Implement application telemetry

Application telemetry is the foundation for observability of your workload. Your application should emit telemetry that provides insight into the state of the application and the achievement of business outcomes. From troubleshooting to measuring the impact of a new feature, application telemetry informs the way you build, operate, and evolve your workload.

Application telemetry consists of metrics and logs. Metrics are diagnostic information, such as your pulse or temperature. Metrics are used collectively to describe the state of your application. Collecting metrics over time can be used to develop baselines and detect anomalies. Logs are messages that the application

sends about its internal state or events that occur. Error codes, transaction identifiers, and user actions are examples of events that are logged.

Desired Outcome:

- Your application emits metrics and logs that provide insight into its health and the achievement of business outcomes.
- Metrics and logs are stored centrally for all applications in the workload.

Common anti-patterns:

- Your application doesn't emit telemetry. You are forced to rely upon your customers to tell you when something is wrong.
- A customer has reported that your application is unresponsive. You have no telemetry and are unable to confirm that the issue exists or characterize the issue without using the application yourself to understand the current user experience.

Benefits of establishing this best practice:

- You can understand the health of your application, the user experience, and the achievement of business outcomes.
- You can react quickly to changes in your application health.
- You can develop application health trends.
- You can make informed decisions about improving your application.
- You can detect and resolve application issues faster.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Implementing application telemetry consists of three steps: identifying a location to store telemetry, identifying telemetry that describes the state of the application, and instrumenting the application to emit telemetry.

As an example, an ecommerce company has a microservices based architecture. As part of their architectural design process they identified application telemetry that would help them understand the state of each microservice. For example, the user cart service emitted telemetry about events like add to cart, abandon cart, and length of time it took to add an item to the cart. All microservices would log errors, warnings, and transaction information. Telemetry would be sent to Amazon CloudWatch for storage and analysis.

Implementation steps

The first step is to identify a central location for telemetry storage for the applications in your workload. If you don’t have an existing platform Amazon CloudWatch provides telemetry collection, dashboards, analysis, and event generation capabilities.

To identify what telemetry you need, start with the following questions:

- Is my application healthy?
- Is my application achieving business outcomes?

  Your application should emit logs and metrics that collectively answer these questions. If you can’t answer those questions with the existing application telemetry, work with business and engineering

stakeholders to create a list of telemetry that can. You can request expert technical advice from your AWS account team as you identify and develop new application telemetry.

Once the additional application telemetry has been identified, work with your engineering stakeholders to instrument your application. The AWS Distro for Open Telemetry provides APIs, libraries, and agents that collect application telemetry. This example demonstrates how to instrument a JavaScript application with custom metrics.

Customers that want to understand the observability services that AWS offers can work through the One Observability Workshop on their own or request support from their AWS account team to guide them. This workshop guides you through the observability solutions at AWS and provides hands-on examples of how they're used.

For a deeper dive into application telemetry, read the Instrumenting distributed systems for operational visibility article in the Amazon Builder's Library. It explains how Amazon instruments applications and can serve as a guide for developing your own instrumentation guidelines.

Level of effort for the implementation plan: Medium

Resources

Related best practices:

the section called "OPS04-BP02 Implement and configure workload telemetry" (p. 32) – Application telemetry is a component of workload telemetry. In order to understand the health of the overall workload you need to understand the health of individual applications that make up the workload.

the section called "OPS04-BP03 Implement user activity telemetry" (p. 33) – User activity telemetry is often a subset of application telemetry. User activity like add to cart events, click streams, or completed transactions provide insight into the user experience.

the section called "OPS04-BP04 Implement dependency telemetry" (p. 34) – Dependency checks are related to application telemetry and may be instrumented into your application. If your application relies on external dependencies like DNS or a database your application can emit metrics and logs on reachability, timeouts, and other events.

the section called "OPS04-BP05 Implement transaction traceability" (p. 34) – Tracing transactions across a workload requires each application to emit information about how they process shared events. The way individual applications handle these events is emitted through their application telemetry.

the section called "OPS08-BP02 Define workload metrics" (p. 60) – Workload metrics are the key health indicators for your workload. Key application metrics are a part of workload metrics.

Related documents:

- AWS Builders Library – Instrumenting Distributed Systems for Operational Visibility
- AWS Distro for OpenTelemetry
- AWS Well-Architected Operational Excellence Whitepaper – Design Telemetry
- Creating metrics from log events using filters
- Implementing Logging and Monitoring with Amazon CloudWatch
- Monitoring application health and performance with AWS Distro for OpenTelemetry
- New – How to better monitor your custom application metrics using Amazon CloudWatch Agent
- Observability at AWS
- Scenario – Publish metrics to CloudWatch
- Start Building – How to Monitor your Applications Effectively
- Using CloudWatch with an AWS SDK

Related videos:

- AWS re:Invent 2021 - Observability the open-source way
- Collect Metrics and Logs from Amazon EC2 instances with the CloudWatch Agent
- How to Easily Setup Application Monitoring for Your AWS Workloads - AWS Online Tech Talks
- Mastering Observability of Your Serverless Applications - AWS Online Tech Talks
- Open Source Observability with AWS - AWS Virtual Workshop

Related examples:

- AWS Logging & Monitoring Example Resources
- AWS Solution: Amazon CloudWatch Monitoring Framework
- AWS Solution: Centralized Logging
- One Observability Workshop

OPS04-BP02 Implement and configure workload telemetry

Design and configure your workload to emit information about its internal state and current status, for example, API call volume, HTTP status codes, and scaling events. Use this information to help determine when a response is required.

Use a service such as Amazon CloudWatch to aggregate logs and metrics from workload components (for example, API logs from AWS CloudTrail, AWS Lambda metrics, Amazon VPC Flow Logs, and other services).

Common anti-patterns:

- Your customers are complaining about poor performance. There are no recent changes to your application and so you suspect an issue with a workload component. You have no telemetry to analyze to determine what component or components are contributing to the poor performance.
- Your application is unreachable. You lack the telemetry to determine if it's a networking issue.

Benefits of establishing this best practice: Understanding what is going on inside your workload enables you to respond if necessary.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Implement log and metric telemetry: Instrument your workload to emit information about its internal state, status, and the achievement of business outcomes. Use this information to determine when a response is required.
  - Gaining better observability of your VMs with Amazon CloudWatch - AWS Online Tech Talks
  - How Amazon CloudWatch works
  - What is Amazon CloudWatch?
  - Using Amazon CloudWatch metrics
  - What is Amazon CloudWatch Logs?
    - Implement and configure workload telemetry: Design and configure your workload to emit information about its internal state and current status (for example, API call volume, HTTP status codes, and scaling events).

- Amazon CloudWatch metrics and dimensions reference
- AWS CloudTrail
- What Is AWS CloudTrail?
- VPC Flow Logs

Resources

Related documents:

- AWS CloudTrail
- Amazon CloudWatch Documentation
- Amazon CloudWatch metrics and dimensions reference
- How Amazon CloudWatch works
- Using Amazon CloudWatch metrics
- VPC Flow Logs
- What Is AWS CloudTrail?
- What is Amazon CloudWatch Logs?
- What is Amazon CloudWatch?

Related videos:

- Application Performance Management on AWS
- Gaining Better Observability of Your VMs with Amazon CloudWatch
- Gaining better observability of your VMs with Amazon CloudWatch - AWS Online Tech Talks

OPS04-BP03 Implement user activity telemetry

Instrument your application code to emit information about user activity, for example, click streams, or started, abandoned, and completed transactions. Use this information to help understand how the application is used, patterns of usage, and to determine when a response is required.

Common anti-patterns:

- Your developers have deployed a new feature without user telemetry, and utilization has increased. You cannot determine if the increased utilization is from use of the new feature, or is an issue introduced with the new code.
- Your developers have deployed a new feature without user telemetry. You cannot tell if your customers are using it without reaching out and asking them.

Benefits of establishing this best practice: Understand how your customers use your application to identify patterns of usage, unexpected behaviors, and to enable you to respond if necessary.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Implement user activity telemetry: Design your application code to emit information about user activity (for example, click streams, or started, abandoned, and completed transactions). Use this

information to help understand how the application is used, patterns of usage, and to determine when a response is required.

OPS04-BP04 Implement dependency telemetry

Design and configure your workload to emit information about the status (for example, reachability or response time) of resources it depends on. Examples of external dependencies can include, external databases, DNS, and network connectivity. Use this information to determine when a response is required.

Common anti-patterns:

- You are unable to determine if the reason your application is unreachable is a DNS issue without manually performing a check to see if your DNS provider is working.
- Your shopping cart application is unable to complete transactions. You are unable to determine if it's a problem with your credit card processing provider without contacting them to verify.

Benefits of establishing this best practice: Understanding the health of your dependencies enables you to respond if necessary.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Implement dependency telemetry: Design and configure your workload to emit information about the state and status of systems it depends on. Some examples include: external databases, DNS, network connectivity, and external credit card processing services.
  - Amazon CloudWatch Agent with AWS Systems Manager integration - unified metrics & log collection for Linux & Windows
  - Collect metrics and logs from Amazon EC2 instances and on-premises servers with the CloudWatch Agent

Resources

Related documents:

- Amazon CloudWatch Agent with AWS Systems Manager integration - unified metrics & log collection for Linux & Windows
- Collect metrics and logs from Amazon EC2 instances and on-premises servers with the CloudWatch Agent

Related examples:

- Well-Architected Labs – Dependency Monitoring

OPS04-BP05 Implement transaction traceability

Implement your application code and configure your workload components to emit information about the flow of transactions across the workload. Use this information to determine when a response is required and to assist you in identifying the factors contributing to an issue.

On AWS, you can use distributed tracing services, such as AWS X-Ray, to collect and record traces as transactions travel through your workload, generate maps to see how transactions flow across your

workload and services, gain insight to the relationships between components, and identify and analyze issues in real time.

Common anti-patterns:

- You have implemented a serverless microservices architecture spanning multiple accounts. Your customers are experiencing intermittent performance issues. You are unable to discover which function or component is responsible because you lack the traces that would allow you to pinpoint where in the application the performance issue exists and what is causing the issue.
- You are trying to determine where the performance bottlenecks are in your workload so that they can be addressed in your development efforts. You are unable to see the relationship between your application components, and the services they interact with, to determine where the bottlenecks are because you lack the traces that would allow you to drill down into the specific services and paths impacting application performance.

Benefits of establishing this best practice: Understanding the flow of transactions across your workload allows you to understand the expected behavior of your workload transactions, and variations from expected behavior across your workload, enabling you to respond if necessary.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Implement transaction traceability: Design your application and workload to emit information about the flow of transactions across system components, such as transaction stage, active component, and time to complete activity. Use this information to determine what is in progress, what is complete, and what the results of completed activities are. This helps you determine when a response is required. For example, longer than expected transaction response times within a component can indicate issues with that component.
  - AWS X-Ray
  - What is AWS X-Ray?

Resources

Related documents:

- AWS X-Ray
- What is AWS X-Ray?

Design for operations

Adopt approaches that improve the flow of changes into production and that enable refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues deployed, and enable rapid identification and remediation of issues introduced through deployment activities.

In AWS, you can view your entire workload (applications, infrastructure, policy, governance, and operations) as code. It can all be defined in and updated using code. This means you can apply the same engineering discipline that you use for application code to every element of your stack.

Best practices

- OPS05-BP01 Use version control (p. 36)
- OPS05-BP02 Test and validate changes (p. 37)

OPS05-BP01 Use version control

Use version control to enable tracking of changes and releases.

Many AWS services offer version control capabilities. Use a revision or source control system such as AWS CodeCommit to manage code and other artifacts, such as version-controlled AWS CloudFormation templates of your infrastructure.

Common anti-patterns:

- You have been developing and storing your code on your workstation. You have had an unrecoverable storage failure on the workstation your code is lost.
- After overwriting the existing code with your changes, you restart your application and it is no longer operable. You are unable to revert to the change.
- You have a write lock on a report file that someone else needs to edit. They contact you asking that you stop work on it so that they can complete their tasks.
- Your research team has been working on a detailed analysis that will shape your future work. Someone has accidentally saved their shopping list over the final report. You are unable to revert the change and will have to recreate the report.

Benefits of establishing this best practice: By using version control capabilities you can easily revert to known good states, previous versions, and limit the risk of assets being lost.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Use version control: Maintain assets in version controlled repositories. Doing so supports tracking changes, deploying new versions, detecting changes to existing versions, and reverting to prior versions (for example, rolling back to a known good state in the event of a failure). Integrate the version control capabilities of your configuration management systems into your procedures.
  - Introduction to AWS CodeCommit
  - What is AWS CodeCommit?

Resources

Related documents:

- What is AWS CodeCommit?

Related videos:

- Introduction to AWS CodeCommit

OPS05-BP02 Test and validate changes

Test and validate changes to help limit and detect errors. Automate testing to reduce errors caused by manual processes, and reduce the level of effort to test.

Many AWS services offer version control capabilities. Use a revision or source control system such as AWS CodeCommit to manage code and other artifacts, such as version-controlled AWS CloudFormation templates of your infrastructure.

Common anti-patterns:

- You deploy your new code to production and customers start calling because your application is no longer working.
- You apply new security groups to enhance your perimeter security. It works with unintended consequences; Your users are unable to access your applications.
- You modify a method invoked by your new function. Another function was also dependant on that method and no longer works. The issue is not detected and enters production. The other function is not invoked for some time and finally fails in production without any correlation to the cause.

Benefits of establishing this best practice: By testing and validating changes early, you are able to address issues with minimized costs and limit the impact on your customers. By testing prior to deployment you minimize the introduction of errors.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Test and validate changes: Changes should be tested and the results validated at all lifecycle stages (for example, development, test, and production). Use testing results to confirm new features and mitigate the risk and impact of failed deployments. Automate testing and validation to ensure consistency of review, to reduce errors caused by manual processes, and reduce the level of effort.
  - What is AWS CodeBuild?
  - Local build support for AWS CodeBuild

Resources

Related documents:

- AWS Developer Tools
- Local build support for AWS CodeBuild
- What is AWS CodeBuild?

OPS05-BP03 Use configuration management systems

Use configuration management systems to make and track configuration changes. These systems reduce errors caused by manual processes and reduce the level of effort to deploy changes.

Static configuration management sets values when initializing a resource that are expected to remain consistent throughout the resource's lifetime. Some examples include setting the configuration for a

web or application server on an instance, or defining the configuration of an AWS service within the AWS Management Console or through the AWS CLI.

Dynamic configuration management sets values at initialization that can or are expected to change during the lifetime of a resource. For example, you could set a feature toggle to enable functionality in your code via a configuration change, or change the level of log detail during an incident to capture more data and then change back following the incident eliminating the now unnecessary logs and their associated expense.

If you have dynamic configurations in your applications running on instances, containers, serverless functions, or devices, you can use AWS AppConfig to manage and deploy them across your environments.

On AWS, you can use AWS Config to continuously monitor your AWS resource configurations across accounts and Regions. It enables you to track their configuration history, understand how a configuration change would affect other resources, and audit them against expected or desired configurations using AWS Config Rules and AWS Config Conformance Packs.

On AWS, you can build continuous integration/continuous deployment (CI/CD) pipelines using services such as AWS Developer Tools (for example, AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeDeploy, and AWS CodeStar).

Have a change calendar and track when significant business or operational activities or events are planned that may be impacted by implementation of change. Adjust activities to manage risk around those plans. AWS Systems Manager Change Calendar provides a mechanism to document blocks of time as open or closed to changes and why, and share that information with other AWS accounts. AWS Systems Manager Automation scripts can be configured to adhere to the change calendar state.

AWS Systems Manager Maintenance Windows can be used to schedule the performance of AWS SSM Run Command or Automation scripts, AWS Lambda invocations, or AWS Step Functions activities at specified times. Mark these activities in your change calendar so that they can be included in your evaluation.

Common anti-patterns:

- You manually update the web server configuration across your fleet and a number of servers become unresponsive due to update errors.
- You manually update your application server fleet over the course of many hours. The inconsistency in configuration during the change causes unexpected behaviors.
- Someone has updated your security groups and your web servers are no longer accessible. Without knowledge of what was changed you spend significant time investigating the issue extending your time to recovery.

Benefits of establishing this best practice: Adopting configuration management systems reduces the level of effort to make and track changes, and the frequency of errors caused by manual procedures.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Use configuration management systems: Use configuration management systems to track and implement changes, to reduce errors caused by manual processes, and reduce the level of effort.
  - Infrastructure configuration management
  - AWS Config
  - What is AWS Config?
  - Introduction to AWS CloudFormation
  - What is AWS CloudFormation?

- AWS OpsWorks
- What is AWS OpsWorks?
- Introduction to AWS Elastic Beanstalk
- What is AWS Elastic Beanstalk?

Resources

Related documents:

- AWS AppConfig
- AWS Developer Tools
- AWS OpsWorks
- AWS Systems Manager Change Calendar
- AWS Systems Manager Maintenance Windows
- Infrastructure configuration management
- What is AWS CloudFormation?
- What is AWS Config?
- What is AWS Elastic Beanstalk?
- What is AWS OpsWorks?

Related videos:

- Introduction to AWS CloudFormation
- Introduction to AWS Elastic Beanstalk

OPS05-BP04 Use build and deployment management systems

Use build and deployment management systems. These systems reduce errors caused by manual processes and reduce the level of effort to deploy changes.

In AWS, you can build continuous integration/continuous deployment (CI/CD) pipelines using services such as AWS Developer Tools (for example, AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeDeploy, and AWS CodeStar).

Common anti-patterns:

- After compiling your code on your development system you, copy the executable onto your production systems and it fails to start. The local log files indicates that it has failed due to missing dependencies.
- You successfully build your application with new features in your development environment and provide the code to Quality Assurance (QA). It fails QA because it is missing static assets.
- On Friday, after much effort, you successfully built your application manually in your development environment including your newly coded features. On Monday, you are unable to repeat the steps that allowed you to successfully build your application.
- You perform the tests you have created for your new release. Then you spend the next week setting up a test environment and performing all the existing integration tests followed by the performance tests. The new code has an unacceptable performance impact and must be redeveloped and then retested.

Benefits of establishing this best practice: By providing mechanisms to manage build and deployment activities you reduce the level of effort to perform repetitive tasks, free your team members to focus on their high value creative tasks, and limit the introduction of error from manual procedures.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Use build and deployment management systems: Use build and deployment management systems to track and implement change, to reduce errors caused by manual processes, and reduce the level of effort. Fully automate the integration and deployment pipeline from code check-in through build, testing, deployment, and validation. This reduces lead time, enables increased frequency of change, and reduces the level of effort.
  - What is AWS CodeBuild?
  - Continuous integration best practices for software development
  - Slalom: CI/CD for serverless applications on AWS
  - Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services
  - What is AWS CodeDeploy?

Resources

Related documents:

- AWS Developer Tools
- What is AWS CodeBuild?
- What is AWS CodeDeploy?

Related videos:

- Continuous integration best practices for software development
- Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services
- Slalom: CI/CD for serverless applications on AWS

OPS05-BP05 Perform patch management

Perform patch management to gain features, address issues, and remain compliant with governance. Automate patch management to reduce errors caused by manual processes, and reduce the level of effort to patch.

Patch and vulnerability management are part of your benefit and risk management activities. It is preferable to have immutable infrastructures and deploy workloads in verified known good states. Where that is not viable, patching in place is the remaining option.

Updating machine images, container images, or Lambda custom runtimes and additional libraries to remove vulnerabilities are part of patch management. You should manage updates to Amazon Machine Images (AMIs) for Linux or Windows Server images using EC2 Image Builder. You can use Amazon Elastic Container Registry with your existing pipeline to manage Amazon ECS images and manage Amazon EKS images. AWS Lambda includes version management features.

Patching should not be performed on production systems without first testing in a safe environment. Patches should only be applied if they support an operational or business outcome. On AWS, you can use AWS Systems Manager Patch Manager to automate the process of patching managed systems and schedule the activity using AWS Systems Manager Maintenance Windows.

Common anti-patterns:

- You are given a mandate to apply all new security patches within two hours resulting in multiple outages due to application incompatibility with patches.
- An unpatched library results in unintended consequences as unknown parties use vulnerabilities within it to access your workload.
- You patch the developer environments automatically without notifying the developers. You receive multiple complaints from the developers that their environment cease to operate as expected.
- You have not patched the commercial off-the-self software on a persistent instance. When you have an issue with the software and contact the vendor, they notify you that version is not supported and you will have to patch to a specific level to receive any assistance.
- A recently released patch for the encryption software you used has significant performance improvements. Your unpatched system has performance issues that remain in place as a result of not patching.

Benefits of establishing this best practice: By establishing a patch management process, including your criteria for patching and methodology for distribution across your environments, you will be able to realize their benefits and control their impact. This will enable the adoption of desired features and capabilities, the removal of issues, and sustained compliance with governance. Implement patch management systems and automation to reduce the level of effort to deploy patches and limit errors caused by manual processes.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Patch management: Patch systems to remediate issues, to gain desired features or capabilities, and to remain compliant with governance policy and vendor support requirements. In immutable systems, deploy with the appropriate patch set to achieve the desired result. Automate the patch management mechanism to reduce the elapsed time to patch, to reduce errors caused by manual processes, and reduce the level of effort to patch.
  - AWS Systems Manager Patch Manager

Resources

Related documents:

- AWS Developer Tools
- AWS Systems Manager Patch Manager

Related videos:

- CI/CD for Serverless Applications on AWS
- Design with Ops in Mind

Related examples:

- Well-Architected Labs – Inventory and Patch Management

OPS05-BP06 Share design standards

Share best practices across teams to increase awareness and maximize the benefits of development efforts.

On AWS, application, compute, infrastructure, and operations can be defined and managed using code methodologies. This allows for easy release, sharing, and adoption.

Many AWS services and resources are designed to be shared across accounts, enabling you to share created assets and learnings across your teams. For example, you can share CodeCommit repositories, Lambda functions, Amazon S3 buckets, and AMIs to specific accounts.

When you publish new resources or updates, use Amazon SNS to provide cross account notifications. Subscribers can use Lambda to get new versions.

If shared standards are enforced in your organization, it's critical that mechanisms exist to request additions, changes, and exceptions to standards in support of teams' activities. Without this option, standards become a constraint on innovation.

Common anti-patterns:

- You have created your own user authentication mechanism, as have each of the other development teams in your organization. Your users have to maintain a separate set of credentials for each part of the system they want to access.
- You have created your own user authentication mechanism, as have each of the other development teams in your organization. Your organization is given a new compliance requirement that must be met. Every individual development team must now invest the resources to implement the new requirement.
- You have created your own screen layout, as have each of the other development teams in your organization. Your users are complaining about the difficulty of navigating the inconsistent interfaces.

Benefits of establishing this best practice: Use shared standards to support the adoption of best practices and to maximizes the benefits of development efforts where standards satisfy requirements for multiple applications or organizations.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Share design standards: Share existing best practices, design standards, checklists, operating procedures, and guidance and governance requirements across teams to reduce complexity and maximize the benefits from development efforts. Ensure that procedures exist to request changes, additions, and exceptions to design standards to support continual improvement and innovation. Ensure that teams are aware of published content so that they can take advantage of content, and limit rework and wasted effort.
  - Delegating access to your AWS environment
  - Share an AWS CodeCommit repository
  - Easy authorization of AWS Lambda functions
  - Sharing an AMI with specific AWS accounts
  - Speed template sharing with an AWS CloudFormation designer URL
  - Using AWS Lambda with Amazon SNS

Resources

Related documents:

- Easy authorization of AWS Lambda functions
- Share an AWS CodeCommit repository
- Sharing an AMI with specific AWS accounts

- Speed template sharing with an AWS CloudFormation designer URL
- Using AWS Lambda with Amazon SNS

Related videos:

- Delegating access to your AWS environment

OPS05-BP07 Implement practices to improve code quality

Implement practices to improve code quality and minimize defects. Some examples include test-driven development, code reviews, and standards adoption.

On AWS, you can integrate services such as Amazon CodeGuru with your pipeline to automatically identify potential code and security issues using program analysis and machine learning. CodeGuru provides recommendations on how to implement the AWS best practices to address these issues.

Common anti-patterns:

- To be able to test your feature sooner, you have decided to not integrate your standard input sanitization library. After testing, you commit your code without remembering to complete incorporation of the library.
- You have minimal experience with the dataset you are processing and are unaware that there are a series of edge cases that can exist in your dataset. Those edge cases are not compatible with the code that you have implemented.

Benefits of establishing this best practice: By adopting practices to improve code quality, you can help minimize issues introduced to production.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Implement practices to improve code quality: Implement practices to improve code quality to minimize defects and the risk of their being deployed. For example, test-driven development, pair programming, code reviews, and standards adoption.
  - Amazon CodeGuru

Resources

Related documents:

- Amazon CodeGuru

OPS05-BP08 Use multiple environments

Use multiple environments to experiment, develop, and test your workload. Use increasing levels of controls as environments approach production to gain confidence your workload will operate as intended when deployed.

Common anti-patterns:

- You are performing development in a shared development environment and another developer overwrites your code changes.
- The restrictive security controls on your shared development environment are preventing you from experimenting with new services and features.
- You perform load testing on your production systems and cause an outage for your users.
- A critical error resulting in data loss has occurred in production. In your production environment, you attempt to recreate the conditions that lead to the data loss so that you can identify how it happened and prevent it from happening again. To prevent further data loss during testing, you are forced to make the application unavailable to your users.
- You are operating a mutli-tenant service and are unable to support a customer request for a dedicated environment.
- You may not always test, but when you do it's in production.
- You believe that the simplicity of a single environment overrides the scope of impact of changes within the environment.

Benefits of establishing this best practice: By deploying multiple environments you can support multiple simultaneous development, testing, and production environments without creating conflicts between developers or user communities.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Use multiple environments: Provide developers sandbox environments with minimized controls to enable experimentation. Provide individual development environments to enable work in parallel, increasing development agility. Implement more rigorous controls in the environments approaching production to allow developers to innovate. Use infrastructure as code and configuration management systems to deploy environments that are configured consistent with the controls present in production to ensure systems operate as expected when deployed. When environments are not in use, turn them off to avoid costs associated with idle resources (for example, development systems on evenings and weekends). Deploy production equivalent environments when load testing to enable valid results.
  - What is AWS CloudFormation?
  - How do I stop and start Amazon EC2 instances at regular intervals using AWS Lambda?

Resources

Related documents:

- How do I stop and start Amazon EC2 instances at regular intervals using AWS Lambda?
- What is AWS CloudFormation?

OPS05-BP09 Make frequent, small, reversible changes

Frequent, small, and reversible changes reduce the scope and impact of a change. This eases troubleshooting, enables faster remediation, and provides the option to roll back a change.

Common anti-patterns:

- You deploy a new version of your application quarterly.
- You frequently make changes to your database schema.

- You perform manual in-place updates, overwriting existing installations and configurations.

Benefits of establishing this best practice: You recognize benefits from development efforts faster by deploying small changes frequently. When the changes are small, it is much easier to identify if they have unintended consequences. When the changes are reversible, there is less risk to implementing the change as recovery is simplified.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Make frequent, small, reversible changes: Frequent, small, and reversible changes reduce the scope and impact of a change. This eases troubleshooting, enables faster remediation, and provides the option to roll back a change. It also increases the rate at which you can deliver value to the business.

OPS05-BP10 Fully automate integration and deployment

Automate build, deployment, and testing of the workload. This reduces errors caused by manual processes and reduces the effort to deploy changes.

Apply metadata using Resource Tags and AWS Resource Groups following a consistent tagging strategy to enable identification of your resources. Tag your resources for organization, cost accounting, access controls, and targeting the execution of automated operations activities.

Common anti-patterns:

- On Friday you, finish authoring the new code for your feature branch. On Monday, after running your code quality test scripts and each of your unit tests scripts, you will check in your code for the next scheduled release.
- You are assigned to code a fix for a critical issue impacting a large number of customers in production. After testing the fix, you commit your code and email change management to request approval to deploy it to production.

Benefits of establishing this best practice: By implementing automated build and deployment management systems, you reduce errors caused by manual processes and reduce the effort to deploy changes enabling your team members to focus on delivering business value.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Use build and deployment management systems: Use build and deployment management systems to track and implement change, to reduce errors caused by manual processes, and reduce the level of effort. Fully automate the integration and deployment pipeline from code check-in through build, testing, deployment, and validation. This reduces lead time, enables increased frequency of change, and reduces the level of effort.
  - What is AWS CodeBuild?
  - Continuous integration best practices for software development
  - Slalom: CI/CD for serverless applications on AWS
  - Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services
  - What is AWS CodeDeploy?

Resources

Related documents:

- What is AWS CodeBuild?
- What is AWS CodeDeploy?

Related videos:

- Continuous integration best practices for software development
- Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services
- Slalom: CI/CD for serverless applications on AWS

Mitigate deployment risks

Adopt approaches that provide fast feedback on quality and enable rapid recovery from changes that do not have desired outcomes. Using these practices mitigates the impact of issues introduced through the deployment of changes.

The design of your workload should include how it will be deployed, updated, and operated. You will want to implement engineering practices that align with defect reduction and quick and safe fixes.

Best practices

- OPS06-BP01 Plan for unsuccessful changes (p. 46)
- OPS06-BP02 Test and validate changes (p. 47)
- OPS06-BP03 Use deployment management systems (p. 47)
- OPS06-BP04 Test using limited deployments (p. 48)
- OPS06-BP05 Deploy using parallel environments (p. 49)
- OPS06-BP06 Deploy frequent, small, reversible changes (p. 50)
- OPS06-BP07 Fully automate integration and deployment (p. 51)
- OPS06-BP08 Automate testing and rollback (p. 52)

OPS06-BP01 Plan for unsuccessful changes

Plan to revert to a known good state, or remediate in the production environment if a change does not have the desired outcome. This preparation reduces recovery time through faster responses.

Common anti-patterns:

- You performed a deployment and your application has become unstable but there appear to be active users on the system. You have to decide whether to roll back the change and impact the active users or wait to roll back the change knowing the users may be impacted regardless.
- After making a routine change, your new environments are accessible but one of your subnets has become unreachable. You have to decide whether to roll back everything or try to fix the inaccessible subnet. While you are making that determination, the subnet remains unreachable.

Benefits of establishing this best practice: Having a plan in place reduces the mean time to recover (MTTR) from unsuccessful changes, reducing the impact to your end users.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Plan for unsuccessful changes: Plan to revert to a known good state (that is, roll back the change), or remediate in the production environment (that is, roll forward the change) if a change does not have the desired outcome. When you identify changes that you cannot roll back if unsuccessful, apply due diligence prior to committing the change.

OPS06-BP02 Test and validate changes

Test changes and validate the results at all lifecycle stages to confirm new features and minimize the risk and impact of failed deployments.

On AWS, you can create temporary parallel environments to lower the risk, effort, and cost of experimentation and testing. Automate the deployment of these environments using AWS CloudFormation to ensure consistent implementations of your temporary environments.

Common anti-patterns:

- You deploy a cool new feature to your application. It doesn't work. You don't know.
- You update your certificates. You accidentally install the certificates to the wrong components. You don't know.

Benefits of establishing this best practice: By testing and validating changes following deployment you are able to identify issues early providing an opportunity to mitigate the impact on your customers.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Test and validate changes: Test changes and validate the results at all lifecycle stages (for example, development, test, and production), to confirm new features and minimize the risk and impact of failed deployments.
  - AWS Cloud9
  - What is AWS Cloud9?
  - How to test and debug AWS CodeDeploy locally before you ship your code

Resources

Related documents:

- AWS Cloud9
- AWS Developer Tools
- How to test and debug AWS CodeDeploy locally before you ship your code
- What is AWS Cloud9?

OPS06-BP03 Use deployment management systems

Use deployment management systems to track and implement change. This reduces errors caused by manual processes and reduces the effort to deploy changes.

In AWS, you can build Continuous Integration/Continuous Deployment (CI/CD) pipelines using services such as AWS Developer Tools (for example, AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeDeploy, and AWS CodeStar).

Common anti-patterns:

- You manually deploy updates to the application servers across your fleet and a number of servers become unresponsive due to update errors.
- You manually deploy to your application server fleet over the course of many hours. The inconsistency in versions during the change causes unexpected behaviors.

Benefits of establishing this best practice: Adopting deployment management systems reduces the level of effort to deploy changes, and the frequency of errors caused by manual procedures.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Use deployment management systems: Use deployment management systems to track and implement change. This will reduce errors caused by manual processes, and reduce the level of effort to deploy changes. Automate the integration and deployment pipeline from code check-in through testing, deployment, and validation. This reduces lead time, enables increased frequency of change, and further reduces the level of effort.
  - Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services
  - What is AWS CodeDeploy?
  - What is AWS Elastic Beanstalk?
  - What is Amazon API Gateway?

Resources

Related documents:

- AWS CodeDeploy User Guide
- AWS Developer Tools
- Try a Sample Blue/Green Deployment in AWS CodeDeploy
- What is AWS CodeDeploy?
- What is AWS Elastic Beanstalk?
- What is Amazon API Gateway?

Related videos:

- Deep Dive on Advanced Continuous Delivery Techniques Using AWS
- Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services

OPS06-BP04 Test using limited deployments

Test with limited deployments alongside existing systems to confirm desired outcomes prior to full scale deployment. For example, use deployment canary testing or one-box deployments.

Common anti-patterns:

- You deploy an unsuccessful change to all of production all at once. You don't know.

Benefits of establishing this best practice: By testing and validating changes following limited deployment you are able to identify issues early with minimal impact on your customers providing an opportunity to further mitigate the impact on your customers.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Test using limited deployments: Test with limited deployments alongside existing systems to confirm desired outcomes prior to full scale deployment. For example, use deployment canary testing or one-box deployments.
  - AWS CodeDeploy User Guide
  - Blue/Green deployments with AWS Elastic Beanstalk
  - Set up an API Gateway canary release deployment
  - Try a Sample Blue/Green Deployment in AWS CodeDeploy
  - Working with deployment configurations in AWS CodeDeploy

Resources

Related documents:

- AWS CodeDeploy User Guide
- Blue/Green deployments with AWS Elastic Beanstalk
- Set up an API Gateway canary release deployment
- Try a Sample Blue/Green Deployment in AWS CodeDeploy
- Working with deployment configurations in AWS CodeDeploy

OPS06-BP05 Deploy using parallel environments

Implement changes onto parallel environments, and then transition over to the new environment. Maintain the prior environment until there is confirmation of successful deployment. Doing so minimizes recovery time by enabling rollback to the previous environment.

Common anti-patterns:

- You perform a mutable deployment by modifying your existing systems. After discovering that the change was unsuccessful, you are forced to modify the systems again to restore the old version extending your time to recovery.
- During a maintenance window, you decommission the old environment and then start building your new environment. Many hours into the procedure, you discover unrecoverable issues with the deployment. While extremely tired, you are forced to find the previous deployment procedures and start rebuilding the old environment.

Benefits of establishing this best practice: By using parallel environments, you can pre-deploy the new environment and transition over to them when desired. If the new environment is not successful, you can recover quickly by transitioning back to your original environment.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Deploy using parallel environments: Implement changes onto parallel environments, and transition or cut over to the new environment. Maintain the prior environment until there is confirmation of successful deployment. This minimizes recovery time by enabling rollback to the previous environment. For example, use immutable infrastructures with blue/green deployments.
  - Working with deployment configurations in AWS CodeDeploy
  - Blue/Green deployments with AWS Elastic Beanstalk
  - Set up an API Gateway canary release deployment
  - Try a Sample Blue/Green Deployment in AWS CodeDeploy

Resources

Related documents:

- AWS CodeDeploy User Guide
- Blue/Green deployments with AWS Elastic Beanstalk
- Set up an API Gateway canary release deployment
- Try a Sample Blue/Green Deployment in AWS CodeDeploy
- Working with deployment configurations in AWS CodeDeploy

Related videos:

- Deep Dive on Advanced Continuous Delivery Techniques Using AWS

OPS06-BP06 Deploy frequent, small, reversible changes

Use frequent, small, and reversible changes to reduce the scope of a change. This results in easier troubleshooting and faster remediation with the option to roll back a change.

Common anti-patterns:

- You deploy a new version of your application quarterly.
- You frequently make changes to your database schema.
- You perform manual in-place updates, overwriting existing installations and configurations.

Benefits of establishing this best practice: You recognize benefits from development efforts faster by deploying small changes frequently. When the changes are small it is much easier to identify if they have unintended consequences. When the changes are reversible there is less risk to implementing the change as recovery is simplified.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Deploy frequent, small, reversible changes: Use frequent, small, and reversible changes to reduce the scope of a change. This results in easier troubleshooting and faster remediation with the option to roll back a change.

OPS06-BP07 Fully automate integration and deployment

Automate build, deployment, and testing of the workload. This reduces errors cause by manual processes and reduces the effort to deploy changes.

Apply metadata using Resource Tags and AWS Resource Groups following a consistent tagging strategy to enable identification of your resources. Tag your resources for organization, cost accounting, access controls, and targeting the execution of automated operations activities.

Common anti-patterns:

- On Friday, you finish authoring the new code for your feature branch. On Monday, after running your code quality test scripts and each of your unit tests scripts, you will check in your code for the next scheduled release.
- You are assigned to code a fix for a critical issue impacting a large number of customers in production. After testing the fix, you commit your code and email change management to request approval to deploy it to production.

Benefits of establishing this best practice: By implementing automated build and deployment management systems you reduce errors caused by manual processes and reduce the effort to deploy changes enabling your team members to focus on delivering business value.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Use build and deployment management systems: Use build and deployment management systems to track and implement change, to reduce errors caused by manual processes, and reduce the level of effort. Fully automate the integration and deployment pipeline from code check-in through build, testing, deployment, and validation. This reduces lead time, enables increased frequency of change, and reduces the level of effort.
  - What is AWS CodeBuild?
  - Continuous integration best practices for software development
  - Slalom: CI/CD for serverless applications on AWS
  - Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services
  - What is AWS CodeDeploy?
  - Deep Dive on Advanced Continuous Delivery Techniques Using AWS

Resources

Related documents:

- Try a Sample Blue/Green Deployment in AWS CodeDeploy
- What is AWS CodeBuild?
- What is AWS CodeDeploy?

Related videos:

- Continuous integration best practices for software development
- Deep Dive on Advanced Continuous Delivery Techniques Using AWS

- Introduction to AWS CodeDeploy - automated software deployment with Amazon Web Services
- Slalom: CI/CD for serverless applications on AWS

OPS06-BP08 Automate testing and rollback

Automate testing of deployed environments to confirm desired outcomes. Automate rollback to a previous known good state when outcomes are not achieved to minimize recovery time and reduce errors caused by manual processes.

Common anti-patterns:

- You deploy changes to your workload. After your see that the change is complete, you start post deployment testing. After you see that they are complete, you realize that your workload is inoperable and customers are disconnected. You then begin rolling back to the previous version. After an extended time to detect the issue, the time to recover is extended by your manual redeployment.

Benefits of establishing this best practice: By testing and validating changes following deployment, you are able to identify issues immediately. By automatically rolling back to the previous version, the impact on your customers is minimized.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Automate testing and rollback: Automate testing of deployed environments to confirm desired outcomes. Automate rollback to a previous known good state when outcomes are not achieved to minimize recovery time and reduce errors caused by manual processes. For example, perform detailed synthetic user transactions following deployment, verify the results, and roll back on failure.
  - Redeploy and roll back a deployment with AWS CodeDeploy

Resources

Related documents:

- Redeploy and roll back a deployment with AWS CodeDeploy

Operational readiness and change management

Evaluate the operational readiness of your workload, processes, procedures, and personnel to understand the operational risks related to your workload. Manage the flow of change into your environments.

You should use a consistent process (including manual or automated checklists) to know when you are ready to go live with your workload or a change. This will also enable you to find any areas that you need to make plans to address. You will have runbooks that document your routine activities and playbooks that guide your processes for issue resolution. Use a mechanism to manage changes that supports the delivery of business value and help mitigate risks associated to change.

Best practices

- OPS07-BP01 Ensure personnel capability (p. 53)
- OPS07-BP02 Ensure consistent review of operational readiness (p. 54)

- OPS07-BP03 Use runbooks to perform procedures (p. 55)
- OPS07-BP04 Use playbooks to investigate issues (p. 56)
- OPS07-BP05 Make informed decisions to deploy systems and changes (p. 58)

OPS07-BP01 Ensure personnel capability

Have a mechanism to validate that you have the appropriate number of trained personnel to provide support for operational needs. Train personnel and adjust personnel capacity as necessary to maintain effective support.

You will need to have enough team members to cover all activities (including on-call). Ensure that your teams have the necessary skills to be successful with training on your workload, your operations tools, and AWS.

AWS provides resources, including the AWS Getting Started Resource Center, AWS Blogs, AWS Online Tech Talks, AWS Events and Webinars, and the AWS Well-Architected Labs, that provide guidance, examples, and detailed walkthroughs to educate your teams. Additionally, AWS Training and Certification provides some free training through self-paced digital courses on AWS fundamentals. You can also register for instructor-led training to further support the development of your teams' AWS skills.

Common anti-patterns:

- Deploying a workload without team members skilled to support the platform and services in use.
- Deploying a workload without team members available during intended hours of support.
- Deploying a workload without sufficient team members to support it if there are team members on leave or out sick.
- Deploying additional workloads without reviewing the additional impact on team members support it and other workloads.

Benefits of establishing this best practice: Having skilled team members enables effective support of your workload.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Personnel capability: Validate that there are sufficient trained personnel to effectively support the workload.
  - Team size: Ensure that you have enough team members to cover operational activities, including on-call duties.
  - Team skill: Ensure that your team members have sufficient training on AWS, your workload, and your operations tools to perform their duties.
    - AWS Events and Webinars
    - Welcome to AWS Training and Certification
  - Review capabilities: Review team size and skill as operating conditions and workloads change, to ensure there is sufficient capability to maintain operational excellence. Make adjustments to ensure that team size and skill match the operational requirements for the workloads that the team supports.

Resources

Related documents:

- AWS Blogs
- AWS Events and Webinars
- AWS Getting Started Resource Center
- AWS Online Tech Talks
- Welcome to AWS Training and Certification

Related examples:

- Well-Architected Labs

OPS07-BP02 Ensure consistent review of operational readiness

Ensure you have a consistent review of your readiness to operate a workload. Reviews must include, at a minimum, the operational readiness of the teams and the workload, and security requirements. Implement review activities in code and trigger automated review in response to events where appropriate, to ensure consistency, speed of delivery, and reduce errors caused by manual processes.

You should automate workload configuration testing by making baselines using AWS Config and checking your configurations using AWS Config Rules. You can evaluate security requirements and compliance using the services and features of AWS Security Hub. These services will aid in determining if your workloads are aligned with best practices and standards.

Common anti-patterns:

- Planning to deploy a workload without understanding if team members have the capability to support it.
- Planning to deploy a workload without understanding the procedures necessary to manage it.
- Planning to deploy a workload without knowing the processes to diagnose issues or respond to incidents.
- Planning to deploy a workload without understanding the security risks present in the workload.
- Planning to deploy a workload without understanding the if the workload complies with your governance and standards.

Benefits of establishing this best practice: Having skilled team members enables effective support of your workload.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Ensure consistent review of operational readiness: Ensure you have a consistent review of your readiness to operate a workload. Review must include at a minimum the operational readiness of the teams and the workload, and security considerations. Review elements can be hard requirements or you can make a risk-based decision to operate a workload that does not satisfy all requirements. Review elements can be specific to a workload, architecture, or can be implementation dependent. Implement reviews as code and trigger reviews in response to events where appropriate, to ensure consistency, speed of execution, and reduce errors caused by manual processes.
  - AWS Well-Architected Guidance: Operational Readiness Reviews (ORR) Whitepaper
  - AWS Systems Manager
  - AWS Config Rules dynamic compliance checking for cloud resources

- How to audit your AWS resources for security compliance by using custom AWS Config Rules
- How to track configuration changes to CloudFormation stacks using AWS Config
- Amazon Inspector update assessment reporting, proxy support, and more
- Create checklists: Ensure you have a consistent review of your readiness to operate a workload. Create operational readiness checklists and validate them against your business, development, operations, and governance requirements. Ensure they address: governance, best practices, configuration standards, restoration procedures, monitoring, maintenance procedures, IT operations procedures, and staffing.
- Use checklists: Make checklists accessible to developers so that they can develop to the appropriate standards. Evaluate checklists when moving between lifecycle stages and environments so that you can identify issues early, when the level of effort to remediate issues is lower. Use the results of checklists to make informed decisions about benefits and risks when considering promoting changes between environments.
- Implement checklists as code and trigger running the checklist in response to events: Implement checklists as code and automatically start running the checklist in response to events where possible, to enhance speed, ensure consistency, and reduce errors caused by manual processes. Integrate automated checklists into deployment pipelines.
  - AWS Config
  - What is AWS Config?
  - AWS Config: evaluating resources with Rules

Resources

Related documents:

- AWS Config
- AWS Config Rules dynamic compliance checking for cloud resources
- AWS Config: evaluating resources with Rules
- AWS Systems Manager
- AWS Well-Architected Guidance: Operational Readiness Reviews (ORR) Whitepaper
- Amazon Inspector update assessment reporting, proxy support, and more
- How to audit your AWS resources for security compliance by using custom AWS Config Rules
- How to track configuration changes to CloudFormation stacks using AWS Config
- What is AWS Config?

OPS07-BP03 Use runbooks to perform procedures

Runbooks are documented procedures to achieve specific outcomes. Enable consistent and prompt responses to well-understood events by documenting procedures in runbooks. Implement runbooks as code and trigger the runbooks in response to events where appropriate, to ensure consistency, speed responses, and reduce errors caused by manual processes.

Common anti-patterns:

- Planning to deploy a workload without knowing the procedures necessary to manage it.

Benefits of establishing this best practice: Capturing runbooks ensures that procedures can be consistently followed. Codifying your runbooks limits the introduction of errors from manual activity. Automating runbooks shortens the time to respond to an event by eliminating the requirement for team member intervention.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Use runbooks to perform standard procedures: Runbooks are documented procedures to achieve specific outcomes. Enable consistent and prompt responses to well understood events by documenting procedures in runbooks. Runbooks must contain the minimum information for an adequately skilled person to achieve the desired outcome. For example, required permissions, required tools, constraints on performing the procedure (for example, specific maintenance windows), and execution steps.
  - Implement runbooks as code: Perform your operations as code by implementing your runbooks as code to ensure consistency and reduce errors caused by manual processes
    - AWS Systems Manager Run Command
    - AWS Systems Manager – Working with Runbooks
    - AWS Systems Manager Automation
    - What is AWS Lambda?
  - Trigger runbooks in response to events: Trigger the execution of runbook code in response to observed events when appropriate. This increases the speed of the response and reduces the level of effort to respond.
    - What is Amazon CloudWatch Events?
    - Creating a CloudWatch Events rule that triggers on an event
    - Creating a CloudWatch Events rule that triggers on an AWS API call using AWS CloudTrail
    - CloudWatch Events event examples from supported services
    - Using Amazon CloudWatch Alarms

Resources

Related documents:

- AWS Systems Manager Automation
- AWS Systems Manager Run Command
- AWS Systems Manager – Working with Runbooks
- CloudWatch Events event examples from supported services
- Creating a CloudWatch Events rule that triggers on an AWS API call using AWS CloudTrail
- Creating a CloudWatch Events rule that triggers on an event
- Using Amazon CloudWatch Alarms
- What is AWS Lambda?
- What is Amazon CloudWatch Events?

Related examples:

- Automating operations with Playbooks and Runbooks

OPS07-BP04 Use playbooks to investigate issues

Enable consistent and prompt responses to issues that are not well understood by documenting the investigation process in playbooks. Playbooks are the predefined steps performed to identify the factors contributing to a failure scenario. The results from any process step are used to determine the next steps to take until the issue is identified or escalated.

AWS allows you to treat your operations as code, scripting your runbook and playbook activities to reduce the risk of human error. You can use Resource Tags or Resource Groups with your scripts to selectively execute based on criteria you have defined (for example, environment, owner, role, or version).

You can use scripted procedures to enable automation by starting the scripts in response to events. By treating both your operations and workloads as code, you can also script and automate the evaluation of your environments.

You should script procedures on your instances using AWS Systems Manager (SSM) Run Command, use AWS Systems Manager Automation to script actions and create workflows on instances and other resources, or use AWS Lambda serverless compute functions to script responses to events across AWS service APIs and your own custom interfaces. You can also use AWS Step Functions to coordinate multiple AWS services scripted into serverless workflows. Automate your responses by initiating these scripts using CloudWatch Events and route desired events to additional operations support systems using Amazon EventBridge.

You should test your procedures, failure scenarios, and the success of your responses (for example, by holding game days and testing prior to going live) to identify areas you need to plan to address.

On AWS, you can create temporary parallel environments to lower the risk, effort, and cost of experimentation and testing. Automate the deployment of these environments using AWS CloudFormation to ensure consistent implementations of your temporary environments. Perform failure injection testing in safe environments where there will be acceptable or no customer impact, and develop or revise appropriate responses.

Common anti-patterns:

- Planning to deploy a workload without knowing the processes to diagnose issues or respond to incidents.

Benefits of establishing this best practice: Creating playbooks ensures that processes can be consistently followed. Codifying your playbooks limits the introduction of errors from manual activity. Automating playbooks shortens the time to respond to an event by eliminating the requirement for team member intervention or providing them additional information when their intervention begins.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Use playbooks to identify issues: Playbooks are documented processes to investigate issues. Enable consistent and prompt responses to failure scenarios by documenting processes in playbooks. Playbooks must contain the information and guidance necessary for an adequately skilled person to gather applicable information, identify potential sources of failure, isolate faults, and determine contributing factors (that is, perform root cause analysis).
  - Implement playbooks as code: Perform your operations as code by scripting your playbooks to ensure consistency and limit reduce errors caused by manual processes. Playbooks can be composed of multiple scripts representing the different steps that might be necessary to identify the contributing factors to an issue. Runbook activities can be triggered or performed as part of playbook activities, or may prompt for execution of a playbook in response to identified events.
    - Automate your operational playbooks with AWS Systems Manager
    - AWS Systems Manager Run Command
    - AWS Systems Manager Automation
    - What is AWS Lambda?
    - What is Amazon CloudWatch Events?
    - Using Amazon CloudWatch Alarms

Resources

Related documents:

- Amazon EventBridge
- AWS Step Functions
- AWS Systems Manager Automation
- AWS Systems Manager Run Command
- Automate your operational playbooks with AWS Systems Manager
- Using Amazon CloudWatch Alarms
- What is AWS Lambda?
- What is Amazon CloudWatch Events?

Related examples:

- Automating operations with Playbooks and Runbooks

OPS07-BP05 Make informed decisions to deploy systems and changes

Evaluate the capabilities of the team to support the workload and the workload's compliance with governance. Evaluate these against the benefits of deployment when determining whether to transition a system or change into production. Understand the benefits and risks to make informed decisions.

A pre-mortem is an exercise where a team simulates a failure to develop mitigation strategies. Use pre-mortems to anticipate failure and create procedures where appropriate. When you make changes to the checklists you use to evaluate your workloads, plan what you will do with live systems that no longer comply.

Common anti-patterns:

- Deciding to deploy a workload without understanding the security risks present in the workload.
- Deciding to deploy a workload without understanding if it complies with your governance and standards.
- Deciding to deploy a workload without understanding if your team can support it.
- Deciding to deploy a workload without understanding how it benefits the organization.

Benefits of establishing this best practice: Having skilled team members enables effective support of your workload.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Make informed decisions to deploy workloads and changes: Evaluate the capabilities of the team to support the workload and the workload's compliance with governance. Evaluate these against the benefits of deployment when determining whether to transition a system or change into production. Understand the benefits and risks, and make informed decisions.

Operate

Success is the achievement of business outcomes as measured by the metrics you define. By understanding the health of your workload and operations, you can identify when organizational and business outcomes may become at risk, or are at risk, and respond appropriately.

To be successful, you must be able to:

Topics

Understanding workload health

Define, capture, and analyze workload metrics to gain visibility to workload events so that you can take appropriate action.

Your team should be able to understand the health of your workload easily. You will want to use metrics based on workload outcomes to gain useful insights. You should use these metrics to implement dashboards with business and technical viewpoints that will help team members make informed decisions.

AWS makes it easy to bring together and analyze your workload logs so that you can generate metrics, understand the health of your workload, and gain insight from operations over time.

Best practices

OPS08-BP01 Identify key performance indicators

Identify key performance indicators (KPIs) based on desired business outcomes (for example, order rate, customer retention rate, and profit versus operating expense) and customer outcomes (for example, customer satisfaction). Evaluate KPIs to determine workload success.

Common anti-patterns:

- You are asked by business leadership how successful a workload has been serving business needs but have no frame of reference to determine success.
- You are unable to determine if the commercial off-the-shelf application you operate for your organization is cost-effective.

Benefits of establishing this best practice: By identifying key performance indicators you enable achieving business outcomes as the test of the health and success of your workload.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Identify key performance indicators: Identify key performance indicators (KPIs) based on desired business and customer outcomes. Evaluate KPIs to determine workload success.

OPS08-BP02 Define workload metrics

Define workload metrics to measure the achievement of KPIs (for example, abandoned shopping carts, orders placed, cost, price, and allocated workload expense). Define workload metrics to measure the health of the workload (for example, interface response time, error rate, requests made, requests completed, and utilization). Evaluate metrics to determine if the workload is achieving desired outcomes, and to understand the health of the workload.

You should send log data to a service such as CloudWatch Logs, and generate metrics from observations of necessary log content.

CloudWatch has specialized features such as Amazon CloudWatch Insights for .NET and SQL Server and Container Insights that can assist you by identifying and setting up key metrics, logs, and alarms across your specifically supported application resources and technology stack.

Common anti-patterns:

- You have defined standard metrics, not associated to any KPIs or tailored to any workload.
- You have errors in your metrics calculations that will yield invalid results.
- You don't have any metrics defined for your workload.
- You only measure for availability.

Benefits of establishing this best practice: By defining and evaluating workload metrics you can determine the health of your workload and measure the achievement of business outcomes.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Define workload metrics: Define workload metrics to measure the achievement of KPIs. Define workload metrics to measure the health of the workload and its individual components. Evaluate metrics to determine if the workload is achieving desired outcomes, and to understand the health of the workload.
  - Publish custom metrics
  - Searching and filtering log data
  - Amazon CloudWatch metrics and dimensions reference

Resources

Related documents:

- Amazon CloudWatch metrics and dimensions reference
- Publish custom metrics
- Searching and filtering log data

OPS08-BP03 Collect and analyze workload metrics

Perform regular proactive reviews of metrics to identify trends and determine where appropriate responses are needed.

You should aggregate log data from your application, workload components, services, and API calls to a service such as CloudWatch Logs. Generate metrics from observations of necessary log content to enable insight into the performance of operations activities.

On AWS, you can analyze workload metrics and identify operational issues using the machine learning capabilities of Amazon DevOps Guru. AWS DevOps Guru provides notification of operational issues with targeted and proactive recommendations to resolve issues and maintain application health.

In the AWS Shared Responsibility Model, portions of monitoring are delivered to you through the AWS Health Dashboard. This dashboard provides alerts and remediation guidance when AWS is experiencing events that might affect you. Customers with Business and Enterprise Support subscriptions also get access to the AWS Health API, enabling integration to their event management systems.

On AWS, you can export your log data to Amazon S3 or send logs directly to Amazon S3 for long-term storage. Using AWS Glue, you can discover and prepare your log data in Amazon S3 for analytics, storing associated metadata in the AWSAWS Glue Data Catalog. Amazon Athena, through its native integration with AWS Glue, can then be used to analyze your log data, querying it using standard SQL. Using a business intelligence tool like Amazon QuickSight you can visualize, explore, and analyze your data.

An alternative solution would be to use the Amazon OpenSearch Service and OpenSearch Dashboards to collect, analyze, and display logs on AWS across multiple accounts and AWS Regions.

Common anti-patterns:

- You are asked by the network design team for current network bandwidth utilization rates. You provide the current metrics, network utilization is at 35%. They reduce circuit capacity as a cost savings measure causing widespread connectivity issues as your point-in-time measurement did not reflect the trend in utilization rates.
- Your router has failed. It has been logging non-critical memory errors with greater and greater frequency up until its complete failure. You did not detect this trend and as a result did not replace the faulty memory before the router caused a service interruption.

Benefits of establishing this best practice: By collecting and analyzing your workload metrics you gain understanding of the health of your workload and can gain insight to trends that may have an impact on your workload or the achievement of your business outcomes.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Collect and analyze workload metrics: Perform regular proactive reviews of metrics to identify trends and determine where appropriate responses are needed.
  - Using Amazon CloudWatch metrics
  - Amazon CloudWatch metrics and dimensions reference
  - Collect metrics and logs from Amazon EC2 instances and on-premises servers with the CloudWatch Agent

Resources

Related documents:

- Amazon Athena
- Amazon CloudWatch metrics and dimensions reference
- Amazon DevOps Guru
- AWS Glue
- AWSAWS Glue Data Catalog
- Amazon OpenSearch Service
- AWS Health Dashboard
- Amazon QuickSight
- Collect metrics and logs from Amazon EC2 instances and on-premises servers with the CloudWatch Agent
- Using Amazon CloudWatch metrics

OPS08-BP04 Establish workload metrics baselines

Establish baselines for metrics to provide expected values as the basis for comparison and identification of under- and over-performing components. Identify thresholds for improvement, investigation, and intervention.

Common anti-patterns:

- A server is running at 95% CPU utilization you are asked if that is good or bad. CPU utilization on that server has not been baselined so you have no idea if that is good or bad.

Benefits of establishing this best practice: By defining baseline metric values you are able to evaluate current metric values, and metric trends, to determine if action is required.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Establish baselines for workload metrics: Establish baselines for workload metrics to provide expected values as the basis for comparison.
  - Creating Amazon CloudWatch Alarms

Resources

Related documents:

- Creating Amazon CloudWatch Alarms

OPS08-BP05 Learn expected patterns of activity for workload

Establish patterns of workload activity to identify anomalous behavior so that you can respond appropriately if required.

CloudWatch through the CloudWatch Anomaly Detection feature applies statistical and machine learning algorithms to generate a range of expected values that represent normal metric behavior.

Amazon DevOps Guru can be used to identify anomalous behavior through event correlation, log analysis, and applying machine learning to analyze your workload telemetry. When unexpected

behaviors are detected, it provides the related metrics and events with recommendations to address the behavior.

Common anti-patterns:

- You are reviewing network utilization logs and see that network utilization increased between 11:30am and 1:30pm and then again at 4:30pm through 6:00pm. You are unaware if this should be considered normal or not.
- Your web servers reboot every night at 3:00am. You are unaware if this is an expected behavior.

Benefits of establishing this best practice: By learning patterns of behavior you can recognize unexpected behavior and take action if necessary.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Learn expected patterns of activity for workload: Establish patterns of workload activity to determine when behavior is outside of the expected values so that you can respond appropriately if required.

Resources

Related documents:

- Amazon DevOps Guru
- CloudWatch Anomaly Detection

OPS08-BP06 Alert when workload outcomes are at risk

Raise an alert when workload outcomes are at risk so that you can respond appropriately if necessary.

Ideally, you have previously identified a metric threshold that you are able to alarm upon or an event that you can use to trigger an automated response.

On AWS, you can use Amazon CloudWatch Synthetics to create canary scripts to monitor your endpoints and APIs by performing the same actions as your customers. The telemetry generated and the insight gained can enable you to identify issues before your customers are impacted.

You can also use CloudWatch Logs Insights to interactively search and analyze your log data using a purpose-built query language. CloudWatch Logs Insights automatically discovers fields in logs from AWS services, and custom log events in JSON. It scales with your log volume and query complexity and gives you answers in seconds, helping you to search for the contributing factors of an incident.

Common anti-patterns:

- You have no network connectivity. No one is aware. No one is trying to identify why or taking action to restore connectivity.
- Following a patch, your persistent instances have become unavailable, disrupting users. Your users have opened support cases. No one has been notified. No one is taking action.

Benefits of establishing this best practice: By identifying that business outcomes are at risk and alerting for action to be taken you have the opportunity to prevent or mitigate the impact of an incident.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Alert when workload outcomes are at risk: Raise an alert when workload outcomes are at risk so that you can respond appropriately if required.
  - What is Amazon CloudWatch Events?
  - Creating Amazon CloudWatch Alarms
  - Invoking Lambda functions using Amazon SNS notifications

Resources

Related documents:

- Amazon CloudWatch Synthetics
- CloudWatch Logs Insights
- Creating Amazon CloudWatch Alarms
- Invoking Lambda functions using Amazon SNS notifications
- What is Amazon CloudWatch Events?

OPS08-BP07 Alert when workload anomalies are detected

Raise an alert when workload anomalies are detected so that you can respond appropriately if necessary.

Your analysis of your workload metrics over time may establish patterns of behavior that you can quantify sufficiently to define an event or raise an alarm in response.

Once trained, the CloudWatch Anomaly Detection feature can be used to alarm on detected anomalies or can provide overlaid expected values onto a graph of metric data for ongoing comparison.

Common anti-patterns:

- Your retail website sales have increased suddenly and dramatically. No one is aware. No one is trying to identify what lead to this surge. No one is taking action to ensure quality customer experiences under the additional load.
- Following the application of a patch, your persistent servers are rebooting frequently, disrupting users. Your servers typically reboot up to three times but not more. No one is aware. No one is trying to identify why this is happening.

Benefits of establishing this best practice: By understanding patterns of workload behavior, you can identify unexpected behavior and take action if necessary.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Alert when workload anomalies are detected: Raise an alert when workload anomalies are detected so that you can respond appropriately if required.
  - What is Amazon CloudWatch Events?
  - Creating Amazon CloudWatch Alarms

- Invoking Lambda functions using Amazon SNS notifications

Resources

Related documents:

- Creating Amazon CloudWatch Alarms
- CloudWatch Anomaly Detection
- Invoking Lambda functions using Amazon SNS notifications
- What is Amazon CloudWatch Events?

OPS08-BP08 Validate the achievement of outcomes and the effectiveness of KPIs and metrics

Create a business-level view of your workload operations to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.

AWS also has support for third-party log analysis systems and business intelligence tools through the AWS service APIs and SDKs (for example, Grafana, Kibana, and Logstash).

Common anti-patterns:

- Page response time has never been considered a contributor to customer satisfaction. You have never established a metric or threshold for page response time. Your customers are complaining about slowness.
- You have not been achieving your minimum response time goals. In an effort to improve response time, you have scaled up your application servers. You are now exceeding response time goals by a significant margin and also have significant unused capacity you are paying for.

Benefits of establishing this best practice: By reviewing and revising KPIs and metrics, you understand how your workload supports the achievement of your business outcomes and can identify where improvement is needed to reach business goals.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Validate the achievement of outcomes and the effectiveness of KPIs and metrics: Create a business level view of your workload operations to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.
  - Using Amazon CloudWatch dashboards
  - What is log analytics?

Resources

Related documents:

- Using Amazon CloudWatch dashboards

- What is log analytics?

Understanding operational health

Define, capture, and analyze operations metrics to gain visibility to workload events so that you can take appropriate action.

Your team should be able to understand the health of your operations easily. You will want to use metrics based on operations outcomes to gain useful insights. You should use these metrics to implement dashboards with business and technical viewpoints that will help team members make informed decisions.

AWS makes it easier to bring together and analyze your operations logs so that you can generate metrics, know the status of your operations, and gain insight from operations over time.

OPS09-BP01 Identify key performance indicators

Identify key performance indicators (KPIs) based on desired business outcomes (for example, new features delivered) and customer outcomes (for example, customer support cases). Evaluate KPIs to determine operations success.

Common anti-patterns:

- You are asked by business leadership how successful operations is at accomplishing business goals but have no frame of reference to determine success.
- You are unable to determine if your maintenance windows have an impact on business outcomes.

Benefits of establishing this best practice: By identifying key performance indicators you enable achieving business outcomes as the test of the health and success of your operations.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Identify key performance indicators: Identify key performance indicators (KPIs) based on desired business and customer outcomes. Evaluate KPIs to determine operations success.

OPS09-BP02 Define operations metrics

Define operations metrics to measure the achievement of KPIs (for example, successful deployments, and failed deployments). Define operations metrics to measure the health of operations activities (for example, mean time to detect an incident (MTTD), and mean time to recovery (MTTR) from an incident). Evaluate metrics to determine if operations are achieving desired outcomes, and to understand the health of your operations activities.

Common anti-patterns:

- Your operations metrics are based on what the team thinks is reasonable.
- You have errors in your metrics calculations that will yield incorrect results.
- You don't have any metrics defined for your operations activities.

Benefits of establishing this best practice: By defining and evaluating operations metrics you can determine the health of your operations activities and measure the achievement of business outcomes.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Define operations metrics: Define operations metrics to measure the achievement of KPIs. Define operations metrics to measure the health of operations and its activities. Evaluate metrics to determine if operations are achieving desired outcomes, and to understand the health of the operations.
  - Publish custom metrics
  - Searching and filtering log data
  - Amazon CloudWatch metrics and dimensions reference

Resources

Related documents:

- AWS Answers: Centralized Logging
- Amazon CloudWatch metrics and dimensions reference
- Detect and React to Changes in Pipeline State with Amazon CloudWatch Events
- Publish custom metrics
- Searching and filtering log data

Related videos:

- Build a Monitoring Plan

OPS09-BP03 Collect and analyze operations metrics

Perform regular, proactive reviews of metrics to identify trends and determine where appropriate responses are needed.

You should aggregate log data from the execution of your operations activities and operations API calls, into a service such as CloudWatch Logs. Generate metrics from observations of necessary log content to gain insight into the performance of operations activities.

On AWS, you can export your log data to Amazon S3 or send logs directly to Amazon S3 for long-term storage. Using AWS Glue, you can discover and prepare your log data in Amazon S3 for analytics, storing associated metadata in the AWSAWS Glue Data Catalog. Amazon Athena, through its native integration with AWS Glue, can then be used to analyze your log data, querying it using standard SQL. Using a business intelligence tool like Amazon QuickSight you can visualize, explore, and analyze your data.

Common anti-patterns:

- Consistent delivery of new features is considered a key performance indicator. You have no method to measure how frequently deployments occur.
- You log deployments, rolled back deployments, patches, and rolled back patches to track you operations activities, but no one reviews the metrics.
- You have a recovery time objective to restore a lost database within fifteen minutes that was defined when the system was deployed and had no users. You now have ten thousand users and have been operating for two years. A recent restore took over two hours. This was not recorded and no one is aware.

Benefits of establishing this best practice: By collecting and analyzing your operations metrics, you gain understanding of the health of your operations and can gain insight to trends that have may an impact on your operations or the achievement of your business outcomes.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Collect and analyze operations metrics: Perform regular proactive reviews of metrics to identify trends and determine where appropriate responses are needed.
  - Using Amazon CloudWatch metrics
  - Amazon CloudWatch metrics and dimensions reference
  - Collect metrics and logs from Amazon EC2 instances and on-premises servers with the CloudWatch Agent

Resources

Related documents:

- Amazon Athena
- Amazon CloudWatch metrics and dimensions reference
- Amazon QuickSight
- AWS Glue
- AWSAWS Glue Data Catalog
- Collect metrics and logs from Amazon EC2 instances and on-premises servers with the CloudWatch Agent
- Using Amazon CloudWatch metrics

OPS09-BP04 Establish operations metrics baselines

Establish baselines for metrics to provide expected values as the basis for comparison and identification of under and over performing operations activities.

Common anti-patterns:

- You have been asked what the expected time to deploy is. You have not measured how long it takes to deploy and can not determine expected times.
- You have been asked what how long it takes to recover from an issue with the application servers. You have no information about time to recovery from first customer contact. You have no information about time to recovery from first identification of an issue through monitoring.
- You have been asked how many support personnel are required over the weekend. You have no idea how many support cases are typical over a weekend and can not provide an estimate.
- You have a recovery time objective to restore lost databases within fifteen minutes that was defined when the system was deployed and had no users. You now have ten thousand users and have been operating for two years. You have no information on how the time to restore has changed for your database.

Benefits of establishing this best practice: By defining baseline metric values you are able to evaluate current metric values, and metric trends, to determine if action is required.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Learn expected patterns of activity for operations: Establish patterns of operations activity to determine when behavior is outside of the expected values so that you can respond appropriately if required.

OPS09-BP05 Learn the expected patterns of activity for operations

Establish patterns of operations activities to identify anomalous activity so that you can respond appropriately if necessary.

Common anti-patterns:

- Your deployment failure rate has increased substantially recently. You address each of the failures independently. You do not realize that the failures correspond to deployments by a new employee who is unfamiliar with the deployment management system.

Benefits of establishing this best practice: By learning patterns of behavior, you can recognize unexpected behavior and take action if necessary.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Learn expected patterns of activity for operations: Establish patterns of operations activity to determine when behavior is outside of the expected values so that you can respond appropriately if required.

OPS09-BP06 Alert when operations outcomes are at risk

Raise an alert when operations outcomes are at risk so that you can respond appropriately if necessary.

Ideally, you have previously identified a metric that you are able to alarm upon or an event that you can use to trigger an automated response.

You can also use CloudWatch Logs Insights to interactively search and analyze your log data using a purpose-built query language. CloudWatch Logs Insights automatically discovers fields in logs from AWS services, and custom log events in JSON. It scales with your log volume and query complexity and gives you answers in seconds helping you to search for the contributing factors of an incident.

Common anti-patterns:

- You are addressing an incident that has caused significant customer impact. Your customers are creating support cases that are not being responded to within the expected time frame. No one is taking action to address this challenge because no one has been alerted to the issue.
- It is 11:00 pm. You are supposed to deploy by midnight to support a customer event but one of the developers is correcting issues with their code. They estimate that they will have the fix within two hours. You are unlikely to deploy in time. You wait to hear from the developer.
- You are unable to restore the backup. You begin working on your resume.

Benefits of establishing this best practice: By identifying that business outcomes are at risk and alerting for action to be taken, you have the opportunity to avert or mitigate the impact of an incident.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Alert when operations outcomes are at risk: Raise an alert when operations outcomes are at risk so that you can respond appropriately if required.
  - What is Amazon CloudWatch Events?
  - Creating Amazon CloudWatch alarms
  - Invoking Lambda functions using Amazon SNS notifications

Resources

Related documents:

- AWS Answers: Centralized Logging
- CloudWatch Logs Insights
- Creating Amazon CloudWatch alarms
- Detect and React to Changes in Pipeline State with Amazon CloudWatch Events
- Invoking Lambda functions using Amazon SNS notifications

OPS09-BP07 Alert when operations anomalies are detected

Raise an alert when operations anomalies are detected so that you can respond appropriately if necessary.

Your analysis of your operations metrics over time may established patterns of behavior that you can quantify sufficiently to define an event or raise an alarm in response.

Once trained, the CloudWatch Anomaly Detection feature can be used to alarm on detected anomalies or can provide overlaid expected values onto a graph of metric data for ongoing comparison.

Amazon DevOps Guru can be used to identify anomalous behavior through event correlation, log analysis, and applying machine learning to analyze your workload telemetry. The insights gained are presented with the relevant data and recommendations.

Common anti-patterns:

- You are applying a patch to your fleet of instances. You tested the patch successfully in the test environment. The patch is failing for a large percentage of instances in your fleet. You do nothing.
- You note that there are deployments starting Friday end of day. Your organization has predefined maintenance windows on Tuesdays and Thursdays. You do nothing.

Benefits of establishing this best practice: By understanding patterns of operations behavior you can identify unexpected behavior and take action if necessary.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Alert when operations anomalies are detected: Raise an alert when operations anomalies are detected so that you can respond appropriately if required.
  - What is Amazon CloudWatch Events?
  - Creating Amazon CloudWatch alarms
  - Invoking Lambda functions using Amazon SNS notifications

Resources

Related documents:

- Amazon DevOps Guru
- CloudWatch Anomaly Detection
- Creating Amazon CloudWatch alarms
- Detect and React to Changes in Pipeline State with Amazon CloudWatch Events
- Invoking Lambda functions using Amazon SNS notifications
- What is Amazon CloudWatch Events?

OPS09-BP08 Validate the achievement of outcomes and the effectiveness of KPIs and metrics

Create a business-level view of your operations activities to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.

AWS also has support for third-party log analysis systems and business intelligence tools through the AWS service APIs and SDKs (for example, Grafana, Kibana, and Logstash).

Common anti-patterns:

- The frequency of your deployments has increased with the growth in number of development teams. Your defined expected number of deployments is once per week. You have been regularly deploying daily. When their is an issue with your deployment system, and deployments are not possible, it goes undetected for days.
- When your business previously provided support only during core business hours from Monday to Friday. You established a next business day response time goal for incidents. You have recently started offering 24x7 support coverage with a two hour response time goal. Your overnight staff are overwhelmed and customers are unhappy. There is no indication that there are issues with incident response times because you are reporting against a next business day target.

Benefits of establishing this best practice: By reviewing and revising KPIs and metrics, you understand how your workload supports the achievement of your business outcomes and can identify where improvement is needed to reach business goals.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Validate the achievement of outcomes and the effectiveness of KPIs and metrics: Create a business level view of your operations activities to help you determine if you are satisfying needs and to

identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.

- Using Amazon CloudWatch dashboards
- What is log analytics?

Resources

Related documents:

- Using Amazon CloudWatch dashboards
- What is log analytics?

Responding to events

You should anticipate operational events, both planned (for example, sales promotions, deployments, and failure tests) and unplanned (for example, surges in utilization and component failures). You should use your existing runbooks and playbooks to deliver consistent results when you respond to alerts. Defined alerts should be owned by a role or a team that is accountable for the response and escalations. You will also want to know the business impact of your system components and use this to target efforts when needed. You should perform a root cause analysis (RCA) after events, and then prevent recurrence of failures or document workarounds.

AWS simplifies your event response by providing tools supporting all aspects of your workload and operations as code. These tools allow you to script responses to operations events and trigger their execution in response to monitoring data.

In AWS, you can improve recovery time by replacing failed components with known good versions, rather than trying to repair them. You can then carry out analysis on the failed resource out of band.

Best practices

- OPS10-BP01 Use processes for event, incident, and problem management (p. 72)
- OPS10-BP02 Have a process per alert (p. 73)
- OPS10-BP03 Prioritize operational events based on business impact (p. 74)
- OPS10-BP04 Define escalation paths (p. 74)
- OPS10-BP05 Enable push notifications (p. 75)
- OPS10-BP06 Communicate status through dashboards (p. 76)
- OPS10-BP07 Automate responses to events (p. 77)

OPS10-BP01 Use processes for event, incident, and problem management

Have processes to address observed events, events that require intervention (incidents), and events that require intervention and either recur or cannot currently be resolved (problems). Use these processes to mitigate the impact of these events on the business and your customers by ensuring timely and appropriate responses.

On AWS, you can use AWS Systems Manager OpsCenter as a central location to view, investigate, and resolve operational issues related to any AWS resource. It aggregates operational issues and provides contextually relevant data to assist in incident response.

Common anti-patterns:

- A customer calls you to complain that your application isn't working. After hanging up you decide to reboot the server and then go back to what you were working on.
- You receive a trouble ticket that one of the applications you support isn't working. After a conversation with your peers you decide to try and find documentation on how it works. You locate some design documentation and start trying to figure out what might be wrong. At the end of your working day, you stop working on the issue and head home for the weekend.
- You receive a call notifying you that the retail site is offline. You share this information with your colleagues and everyone stops what they are doing to work on the issue independently.

Benefits of establishing this best practice: Having predefined processes for event, incident, and problem management enables consistent and timely responses to mitigate the impact on the business and your customers.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Use processes for event, incident, and problem management: Have processes to address observed events, events that require intervention (incidents), and events that require intervention and either recur or cannot currently be resolved (problems). Use these processes to mitigate the impact of these events on the business and your customers by ensuring timely and appropriate responses.

Resources

Related documents:

- AWS Systems Manager OpsCenter

OPS10-BP02 Have a process per alert

Have a well-defined response (runbook or playbook), with a specifically identified owner, for any event for which you raise an alert. This ensures effective and prompt responses to operations events and prevents actionable events from being obscured by less valuable notifications.

Common anti-patterns:

- Your monitoring system presents you a stream of approved connections along with other messages. The volume of messages is so large that you miss periodic error messages that require your intervention.
- You receive an alert that the website is down. There is no defined process for when this happens. You are forced to take an ad hoc approach to diagnose and resolve the issue. Developing this process as you go extends the time to recovery.

Benefits of establishing this best practice: By alerting only when action is required, you prevent low value alerts from concealing high value alerts. By having a process for every actionable alert, you enable a consistent and prompt response to events in your environment.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Process per alert: Any event for which you raise an alert should have a well-defined response (runbook or playbook) with a specifically identified owner (for example, individual, team, or role) accountable for successful completion. Performance of the response may be automated or conducted by another

team but the owner is accountable for ensuring the process delivers the expected outcomes. By having these processes, you ensure effective and prompt responses to operations events and you can prevent actionable events from being obscured by less valuable notifications. For example, automatic scaling might be applied to scale a web front end, but the operations team might be accountable to ensure that the automatic scaling rules and limits are appropriate for workload needs.

Resources

Related documents:

- Amazon CloudWatch Features
- What is Amazon CloudWatch Events?

Related videos:

- Build a Monitoring Plan

OPS10-BP03 Prioritize operational events based on business impact

Ensure that when multiple events require intervention, those that are most significant to the business are addressed first. Impacts can include loss of life or injury, financial loss, or damage to reputation or trust.

Common anti-patterns:

- You receive a support request to add a printer configuration for a user. While working on the issue, you receive a support request stating that your retail site is down. After completing the printer configuration for your user, you start work on the website issue.
- You get notified that both your retail website and your payroll system are down. You don't know which one should get priority.

Benefits of establishing this best practice: Prioritizing responses to the incidents with the greatest impact on the business enables your management of that impact.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Prioritize operational events based on business impact: Ensure that when multiple events require intervention, those that are most significant to the business are addressed first. Impacts can include loss of life or injury, financial loss, regulatory violations, or damage to reputation or trust.

OPS10-BP04 Define escalation paths

Define escalation paths in your runbooks and playbooks, including what triggers escalation, and procedures for escalation. Specifically identify owners for each action to ensure effective and prompt responses to operations events.

Identify when a human decision is required before an action is taken. Work with decision makers to have that decision made in advance, and the action preapproved, so that MTTR is not extended waiting for a response.

Common anti-patterns:

- Your retail site is down. You don't understand the runbook for recovering the site. You start calling colleagues hoping that someone will be able to help you.
- You receive a support case for an unreachable application. You don't have permissions to administer the system. You don't know who does. You attempt to contact the system owner that opened the case and there is no response. You have no contacts for the system and your colleagues are not familiar with it.

Benefits of establishing this best practice: By defining escalations, triggers for escalation, and procedures for escalation you enable the systematic addition of resources to an incident at an appropriate rate for the impact.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Define escalation paths: Define escalation paths in your runbooks and playbooks, including what triggers escalation, and procedures for escalation. For example, escalation of an issue from support engineers to senior support engineers when runbooks cannot resolve the issue, or when a predefined period of time has elapsed. Another example of an appropriate escalation path is from senior support engineers to the development team for a workload when the playbooks are unable to identify a path to remediation, or when a predefined period of time has elapsed. Specifically identify owners for each action to ensure effective and prompt responses to operations events. Escalations can include third parties. For example, a network connectivity provider or a software vendor. Escalations can include identified authorized decision makers for impacted systems.

OPS10-BP05 Enable push notifications

Communicate directly with your users (for example, with email or SMS) when the services they use are impacted, and again when the services return to normal operating conditions, to enable users to take appropriate action.

Common anti-patterns:

- Your application is experiencing a distributed denial of service incident and has been unresponsive for days. There is no error message. You have not sent a notification email. You have not sent text notifications. You have not shared information on social media. You customers are frustrated and looking for other vendors who can support them.
- On Monday, your application had issues following a patch and was down for a couple of hours. On Tuesday, your application had issues following a code deployment and was unreliable for a couple of hours. On Wednesday, your application had issues following a code deployment to mitigate a security vulnerability associated to the failed patch and was unavailable for a couple of hours. On Thursday, your frustrated customers started looking for another vendor who could support them.
- Your application is going to be down for maintenance this weekend. You don't inform your customers. Some of your customers had scheduled activities involving the use of your application. They are very frustrated upon discovery that your application is not available.

Benefits of establishing this best practice: By defining notifications, triggers for notifications, and procedures for notifications you enable your customer to be informed and respond when issues with your workload impact them.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Enable push notifications: Communicate directly with your users (for example, with email or SMS) when the services they use are impacted, and when the services return to normal operating conditions, to enable users to take appropriate action.
  - Amazon SES features
  - What is Amazon SES?
  - Set up Amazon SNS notifications

Resources

Related documents:

- Amazon SES features
- Set up Amazon SNS notifications
- What is Amazon SES?

OPS10-BP06 Communicate status through dashboards

Provide dashboards tailored to their target audiences (for example, internal technical teams, leadership, and customers) to communicate the current operating status of the business and provide metrics of interest.

You can create dashboards using Amazon CloudWatch Dashboards on customizable home pages in the CloudWatch console. Using business intelligence services such as Amazon QuickSight you can create and publish interactive dashboards of your workload and operational health (for example, order rates, connected users, and transaction times). Create Dashboards that present system and business-level views of your metrics.

Common anti-patterns:

- Upon request, you run a report on the current utilization of your application for management.
- During an incident, you are contacted every twenty minutes by a concerned system owner wanting to know if it is fixed yet.

Benefits of establishing this best practice: By creating dashboards, you enable self-service access to information enabling your customers to informed themselves and determine if they need to take action.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Communicate status through dashboards: Provide dashboards tailored to their target audiences (for example, internal technical teams, leadership, and customers) to communicate the current operating status of the business and provide metrics of interest. Providing a self-service option for status information reduces the disruption of fielding requests for status by the operations team. Examples include Amazon CloudWatch dashboards, and AWS Health Dashboard.
  - CloudWatch dashboards create and use customized metrics views

Resources

Related documents:

- Amazon QuickSight
- CloudWatch dashboards create and use customized metrics views

OPS10-BP07 Automate responses to events

Automate responses to events to reduce errors caused by manual processes, and to ensure prompt and consistent responses.

There are multiple ways to automate runbook and playbook actions on AWS. To respond to an event from a state change in your AWS resources, or from your own custom events, you should create CloudWatch Events rules to trigger responses through CloudWatch targets (for example, Lambda functions, Amazon Simple Notification Service (Amazon SNS) topics, Amazon ECS tasks, and AWS Systems Manager Automation).

To respond to a metric that crosses a threshold for a resource (for example, wait time), you should create CloudWatch alarms to perform one or more actions using Amazon EC2 actions, Auto Scaling actions, or to send a notification to an Amazon SNS topic. If you need to perform custom actions in response to an alarm, invoke Lambda through an Amazon SNS notification. Use Amazon SNS to publish event notifications and escalation messages to keep people informed.

AWS also supports third-party systems through the AWS service APIs and SDKs. There are a number of monitoring tools provided by AWS Partners and third parties that allow for monitoring, notifications, and responses. Some of these tools include New Relic, Splunk, Loggly, SumoLogic, and Datadog.

You should keep critical manual procedures available for use when automated procedures fail

Common anti-patterns:

- A developer checks in their code. This event could have been used to start a build and then perform testing but instead nothing happens.
- You application logs a specific error before it stops working. The procedure to restart the application is well understood and could be scripted. You could use the log event to invoke a script and restart the application. Instead, when the error happens at 3am Sunday morning, you are woken up as the on-call resource responsible to fix the system.

Benefits of establishing this best practice: By using automated responses to events, you reduce the time to respond and limit the introduction of errors from manual activities.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Automate responses to events: Automate responses to events to reduce errors caused by manual processes, and to ensure prompt and consistent responses.
  - What is Amazon CloudWatch Events?
  - Creating a CloudWatch Events rule that triggers on an event
  - Creating a CloudWatch Events rule that triggers on an AWS API call using AWS CloudTrail
  - CloudWatch Events event examples from supported services

Resources

Related documents:

- Amazon CloudWatch Features
- CloudWatch Events event examples from supported services
- Creating a CloudWatch Events rule that triggers on an AWS API call using AWS CloudTrail
- Creating a CloudWatch Events rule that triggers on an event
- What is Amazon CloudWatch Events?

Related videos:

- Build a Monitoring Plan

Related examples:

Evolve

Evolution is the continuous cycle of improvement over time. Implement frequent small incremental changes based on the lessons learned from your operations activities and evaluate their success at bringing about improvement.

To evolve your operations over time, you must be able to:

Topics

Learn, share, and improve

It's essential that you regularly provide time for analysis of operations activities, analysis of failures, experimentation, and making improvements. When things fail, you will want to ensure that your team, as well as your larger engineering community, learns from those failures. You should analyze failures to identify lessons learned and plan improvements. You will want to regularly review your lessons learned with other teams to validate your insights.

Best practices

OPS11-BP01 Have a process for continuous improvement

Regularly evaluate and prioritize opportunities for improvement to focus efforts where they can provide the greatest benefits.

Common anti-patterns:

- You have documented the procedures necessary to create a development or testing environment. You could use CloudFormation to automate the process, but instead you do it manually from the console.
- Your testing shows that the vast majority of CPU utilization inside your application is in a small set of inefficient functions. You could focus on improving them and reduce your costs but you have been tasked to create a new usability feature.

Benefits of establishing this best practice: Continual improvement provides a mechanism to regularly evaluate opportunities for improvement, prioritize opportunities, and focus efforts where they can provide the greatest benefits.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Define processes for continuous improvement: Regularly evaluate and prioritize opportunities for improvement to focus efforts where they provide the greatest benefits. Implement changes to improve and evaluate the outcomes to determine success. If the outcomes do not satisfy the goals, and the improvement is still a priority, iterate using alternative courses of action. Your operations processes should include dedicated time and resources to make continuous incremental improvements possible.

OPS11-BP02 Perform post-incident analysis

Review customer-impacting events, and identify the contributing factors and preventative actions. Use this information to develop mitigations to limit or prevent recurrence. Develop procedures for prompt and effective responses. Communicate contributing factors and corrective actions as appropriate, tailored to target audiences.

Common anti-patterns:

- You administer an application server. Approximately every 23 hours and 55 minutes all your active sessions are terminated. You have tried to identify what is going wrong on your application server. You suspect it could instead be a network issue but are unable to get cooperation from the network team as they are too busy to support you. You lack a predefined process to follow to get support and collect the information necessary to determine what is going on.
- You have had data loss within your workload. This is the first time it has happened and the cause is not obvious. You decide it is not important because you can recreate the data. Data loss starts occurring with greater frequency impacting your customers. This also places addition operational burden on you as you restore the missing data.

Benefits of establishing this best practice: Having a predefined processes to determine the components, conditions, actions, and events that contributed to an incident enables you to identify opportunities for improvement.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Use a process to determine contributing factors: Review all customer impacting incidents. Have a process to identify and document the contributing factors of an incident so that you can develop mitigations to limit or prevent recurrence and you can develop procedures for prompt and effective responses. Communicate root cause as appropriate, tailored to target audiences.

OPS11-BP03 Implement feedback loops

Include feedback loops in your procedures and workloads to help you identify issues and areas that need improvement.

Common anti-patterns:

- A single frustrated customer opens a support case for a new product feature request to address a perceived issue. It is added to the list of priority improvements.
- You do not accept customer feedback.
- There are a significant number of requests for a change to an existing feature. Its originator maintains that it works as intended and does not need to change, so no changes are made.

- When a specific team member performs a deploy it always fails. No action is taken.

Benefits of establishing this best practice: By using the data from feedback loops to identify where to target improvement, you can minimize the impact of event based motivations or emotional investment.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Feedback loops: Have procedures embedded in your operations activities to capture feedback from their execution and to identify areas for improvement.
  - Immediate feedback: Immediate feedback comes from the execution of operations activities where, through review of the execution and outcomes, it is recognized that the process could be improved. Feedback can come from customers, team members, or automated output of an activity. When the improvement has a low level of effort, or significant benefit, consider implementing it immediately. Track opportunities for improvement in your backlog or issue system as appropriate. For example, a process where data is staged on an intermediate device could be optimized by instead placing the data directly into the target environment. This would eliminate a step in the process and the requirement for the intermediate resources.
  - Retrospective analysis: Perform retrospective analysis regularly to capture feedback from the review of operational outcomes and metrics over time. Use trends to identify areas that need improvement. For example, review the rate of deployment failures to identify when potential issues with development and deployment activities have emerged.
    - Serverless big data analytics - Amazon Athena and Amazon QuickSight - 2017 AWS Online Tech Talks
    - View AWS CodeDeploy logs in Amazon CloudWatch console
    - Analyzing VPC flow logs with Amazon Kinesis Firehose, Amazon Athena, and Amazon QuickSight

Resources

Related documents:

- Analyzing VPC flow logs with Amazon Kinesis Firehose, Amazon Athena, and Amazon QuickSight
- View AWS CodeDeploy logs in Amazon CloudWatch console

Related videos:

- Serverless big data analytics - Amazon Athena and Amazon QuickSight - 2017 AWS Online Tech Talks

OPS11-BP04 Perform knowledge management

Mechanisms exist for your team members to discover the information that they are looking for in a timely manner, access it, and identify that it's current and complete. Mechanisms are present to identify needed content, content in need of refresh, and content that should be archived so that it's no longer referenced.

Common anti-patterns:

- A single frustrated customer opens a support case for a new product feature request to address a perceived issue. It is added to the list of priority improvements.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Knowledge management: Ensure mechanisms exist for your team members to discover the information that they are looking for in a timely manner, access it, and identify that it's current and complete. Maintain mechanisms to identify needed content, content in need of refresh, and content that should be archived so that it's no longer referenced.

OPS11-BP05 Define drivers for improvement

Identify drivers for improvement to help you evaluate and prioritize opportunities.

On AWS, you can aggregate the logs of all your operations activities, workloads, and infrastructure to create a detailed activity history. You can then use AWS tools to analyze your operations and workload health over time (for example, identify trends, correlate events and activities to outcomes, and compare and contrast between environments and across systems) to reveal opportunities for improvement based on your drivers.

You should use CloudTrail to track API activity (through the AWS Management Console, CLI, SDKs, and APIs) to know what is happening across your accounts. Track your AWS developer Tools deployment activities with CloudTrail and CloudWatch. This will add a detailed activity history of your deployments and their outcomes to your CloudWatch Logs log data.

Export your log data to Amazon S3 for long-term storage. Using AWS Glue, you discover and prepare your log data in Amazon S3 for analytics. Use Amazon Athena, through its native integration with AWS Glue, to analyze your log data. Use a business intelligence tool like Amazon QuickSight to visualize, explore, and analyze your data

Common anti-patterns:

- You have a script that works but is not elegant. You invest time in rewriting it. It is now a work of art.
- Your start-up is trying to get another set of funding from a venture capitalist. They want you to demonstrate compliance with PCI DSS. You want to make them happy so you document your compliance and miss a delivery date for a customer, losing that customer. It wasn't a wrong thing to do but now you wonder if it was the right thing to do.

Benefits of establishing this best practice: By determining the criteria you want to use for improvement, you can minimize the impact of event based motivations or emotional investment.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Understand drivers for improvement: You should only make changes to a system when a desired outcome is supported.
  - Desired capabilities: Evaluate desired features and capabilities when evaluating opportunities for improvement.
    - What's New with AWS
  - Unacceptable issues: Evaluate unacceptable issues, bugs, and vulnerabilities when evaluating opportunities for improvement.
    - AWS Latest Security Bulletins
    - AWS Trusted Advisor
  - Compliance requirements: Evaluate updates and changes required to maintain compliance with regulation, policy, or to remain under support from a third party, when reviewing opportunities for improvement.

- AWS Compliance
- AWS Compliance Programs
- AWS Compliance Latest News

Resources

Related documents:

- Amazon Athena
- Amazon QuickSight
- AWS Compliance
- AWS Compliance Latest News
- AWS Compliance Programs
- AWS Glue
- AWS Latest Security Bulletins
- AWS Trusted Advisor
- Export your log data to Amazon S3
- What's New with AWS

OPS11-BP06 Validate insights

Review your analysis results and responses with cross-functional teams and business owners. Use these reviews to establish common understanding, identify additional impacts, and determine courses of action. Adjust responses as appropriate.

Common anti-patterns:

- You see that CPU utilization is at 95% on a system and make it a priority to find a way to reduce load on the system. You determine the best course of action is to scale up. The system is a transcoder and the system is scaled to run at 95% CPU utilization all the time. The system owner could have explained the situation to you had you contacted them. Your time has been wasted.
- A system owner maintains that their system is mission critical. The system was not placed in a high security environment. To improve security, you implement the additional detective and preventative controls that are required for mission critical systems. You notify the system owner that the work is complete and that he will be charged for the additional resources. In the discussion following this notification, the system owner learns there is a formal definition for mission critical systems that this system does not meet.

Benefits of establishing this best practice: By validating insights with business owners and subject matter experts, you can establish common understanding and more effectively guide improvement.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Validate insights: Engage with business owners and subject matter experts to ensure there is common understanding and agreement of the meaning of the data you have collected. Identify additional concerns, potential impacts, and determine a courses of action.

OPS11-BP07 Perform operations metrics reviews

Regularly perform retrospective analysis of operations metrics with cross-team participants from different areas of the business. Use these reviews to identify opportunities for improvement, potential courses of action, and to share lessons learned.

Look for opportunities to improve in all of your environments (for example, development, test, and production).

Common anti-patterns:

- There was a significant retail promotion that was interrupted by your maintenance window. The business remains unaware that there is a standard maintenance window that could be delayed if there are other business impacting events.
- You suffered an extended outage because of your use of a buggy library commonly used in your organization. You have since migrated to a reliable library. The other teams in your organization do not know that they are at risk. If you met regularly and reviewed this incident, they would be aware of the risk.
- Performance of your transcoder has been falling off steadily and impacting the media team. It isn't terrible yet. You will not have an opportunity to find out until it is bad enough to cause an incident. Were you to review your operations metrics with the media team, there would be an opportunity for the change in metrics and their experience to be recognized and the issue addressed.
- You are not reviewing your satisfaction of customer SLAs. You are trending to not meet your customer SLAs. There are financial penalties related to not meeting your customer SLAs. If you meet regularly to review the metrics for these SLAs, you would have the opportunity to recognize and address the issue.

Benefits of establishing this best practice: By meeting regularly to review operations metrics, events, and incidents, you maintain common understanding across teams, share lessons learned, and can prioritize and target improvements.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Operations metrics reviews: Regularly perform retrospective analysis of operations metrics with cross-team participants from different areas of the business. Engage stakeholders, including the business, development, and operations teams, to validate your findings from immediate feedback and retrospective analysis, and to share lessons learned. Use their insights to identify opportunities for improvement and potential courses of action.
  - Amazon CloudWatch
  - Using Amazon CloudWatch metrics
  - Publish custom metrics
  - Amazon CloudWatch metrics and dimensions reference

Resources

Related documents:

- Amazon CloudWatch
- Amazon CloudWatch metrics and dimensions reference
- Publish custom metrics
- Using Amazon CloudWatch metrics

OPS11-BP08 Document and share lessons learned

Document and share lessons learned from the operations activities so that you can use them internally and across teams.

You should share what your teams learn to increase the benefit across your organization. You will want to share information and resources to prevent avoidable errors and ease development efforts. This will allow you to focus on delivering desired features.

Use AWS Identity and Access Management (IAM) to define permissions enabling controlled access to the resources you wish to share within and across accounts. You should then use version-controlled AWS CodeCommit repositories to share application libraries, scripted procedures, procedure documentation, and other system documentation. Share your compute standards by sharing access to your AMIs and by authorizing the use of your Lambda functions across accounts. You should also share your infrastructure standards as AWS CloudFormation templates.

Through the AWS APIs and SDKs, you can integrate external and third-party tools and repositories (for example, GitHub, BitBucket, and SourceForge). When sharing what you have learned and developed, be careful to structure permissions to ensure the integrity of shared repositories.

Common anti-patterns:

- You suffered an extended outage because of your use of a buggy library commonly used in your organization. You have since migrated to a reliable library. The other teams in your organization do not know they are at risk. Were you to document and share your experience with this library, they would be aware of the risk.
- You have identified an edge case in an internally shared microservice that causes sessions to drop. You have updated your calls to the service to avoid this edge case. The other teams in your organization do not know that they are at risk. Were you to document and share your experience with this library, they would be aware of the risk.
- You have found a way to significantly reduce the CPU utilization requirements for one of your microservices. You do not know if any other teams could take advantage of this technique. Were you to document and share your experience with this library, they would have the opportunity to do so.

Benefits of establishing this best practice: Share lessons learned to support improvement and to maximize the benefits of experience.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Document and share lessons learned: Have procedures to document the lessons learned from the execution of operations activities and retrospective analysis so that they can be used by other teams.
  - Share learnings: Have procedures to share lessons learned and associated artifacts across teams. For example, share updated procedures, guidance, governance, and best practices through an accessible wiki. Share scripts, code, and libraries through a common repository.
    - Delegating access to your AWS environment
    - Share an AWS CodeCommit repository
    - Easy authorization of AWS Lambda functions
    - Sharing an AMI with specific AWS Accounts
    - Speed template sharing with an AWS CloudFormation designer URL
    - Using AWS Lambda with Amazon SNS

Resources

Related documents:

- Easy authorization of AWS Lambda functions
- Share an AWS CodeCommit repository
- Sharing an AMI with specific AWS Accounts
- Speed template sharing with an AWS CloudFormation designer URL
- Using AWS Lambda with Amazon SNS

Related videos:

- Delegating access to your AWS environment

OPS11-BP09 Allocate time to make improvements

Dedicate time and resources within your processes to make continuous incremental improvements possible.

On AWS, you can create temporary duplicates of environments, lowering the risk, effort, and cost of experimentation and testing. These duplicated environments can be used to test the conclusions from your analysis, experiment, and develop and test planned improvements.

Common anti-patterns:

- There is a known performance issue in your application server. It is added to the backlog behind every planned feature implementation. If the rate of planned features being added remains constant, the performance issue will never be addressed.
- To support continual improvement you approve administrators and developers using all their extra time to select and implement improvements. No improvements are ever completed.

Benefits of establishing this best practice: By dedicating time and resources within your processes you make continuous incremental improvements possible.

Level of risk exposed if this best practice is not established: Low

Implementation guidance

- Allocate time to make improvements: Dedicate time and resources within your processes to make continuous incremental improvements possible. Implement changes to improve and evaluate the results to determine success. If the results do not satisfy the goals, and the improvement is still a priority, pursue alternative courses of action.

Conclusion

Operational excellence is an ongoing and iterative effort.

Set up your organization for success by having shared goals. Ensure that everyone understands their part in achieving business outcomes and how they impact the ability of others to succeed. Provide support for your team members so that they can support your business outcomes.

Every operational event and failure should be treated as an opportunity to improve the operations of your architecture. By understanding the needs of your workloads, predefining runbooks for routine activities, and playbooks to guide issue resolution, using the operations as code features in AWS, and maintaining situational awareness, your operations will be better prepared and able to respond more effectively when incidents occur.

Through focusing on incremental improvement based on priorities as they change, and lessons learned from event response and retrospective analysis, you will enable the success of your business by increasing the efficiency and effectiveness of your activities.

AWS strives to help you build and operate architectures that maximize efficiency while you build highly responsive and adaptive deployments. To increase the operational excellence of your workloads, you should use the best practices discussed in this paper.

Contributors

- Rich Boyd, Operational Excellence Pillar Lead, Well-Architected, Amazon Web Services
- Jon Steele, Solutions Architect Well-Architected, Amazon Web Services
- Ryan King, Sr. Technical Program Manager, Amazon Web Services
- Chris Kunselman, Advisory Consultant, Amazon Web Services
- Peter Mullen, Advisory Consultant, Amazon Web Services
- Brian Quinn, Sr. Advisory Consultant, Amazon Web Services
- David Stanley, Cloud Operating Model Lead, Amazon Web Services

Further reading

For additional guidance, consult the following sources:

- AWS Well-Architected Framework
- AWS Architecture Center

Document revisions

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

Change	Description	Date
Whitepaper updated (p. 90)	Best practices expanded and improvement plans added.	October 20, 2022
Minor update (p. 90)	Small editorial update.	August 8, 2022
Whitepaper updated (p. 90)	Updates to reflect new AWS services and features, and latest best practices.	February 2, 2022
Minor update (p. 1)	Added Sustainability Pillar to introduction.	December 2, 2021
Updates for new Framework (p. 90)	Updates to reflect new AWS services and features, and latest best practices.	July 8, 2020
Whitepaper updated (p. 90)	Updates to reflect new AWS services and features, and updated references.	July 1, 2018
Initial publication (p. 90)	Operational Excellence Pillar - AWS Well-Architected Framework published.	November 1, 2017